

RĪGAS TEHNISKĀ UNIVERSITĀTE
Datorzinātnes un informācijas tehnoloģijas fakultāte
Informācijas tehnoloģijas institūta

Maksims Matulenko
bakalaura līmeņa studiju programmas
students, stud. apl. nr. 201RDB012

**Salīdzinošs pētījums par
mašīnmācīšanās algoritmiem "Bitcoin"
cenu prognozēšanai
BAKALAURA DARBS**

Zinātniskais vadītājs: Asoc.prof. Modelēšanas un imitācijas katedra, DITF, RTU.
Dr. Sergejs Paršutins;

RĪGA 2023

RĪGAS TEHNISKĀ UNIVERSITĀTE
DATORZINĀTNES UN INFORMĀCIJAS TEHNOLOĢIJAS FAKULTĀTE
Informācijas tehnoloģijas institūta

bakalaura darba izpildes lapa

Noslēguma darba autors:

students(-e) Maksims Matulenko

(paraksts, datums)

Noslēguma darbs ieteikts aizstāvēšanai:

Zinātniskais vadītājs:

Asoc.prof Dr. Sergejs Paršutins

(paraksts, datums)

ANOTĀCIJA

Atslēgas: Bitcoin, Dziļā mašīnmācīšanās, Bajesa optimizācija, Akciju tirdzniecība

“Salīdzinošs pētījums par mašīnmācīšanās algoritmiem "Bitcoin" cenu prognozēšanai” šī pētījuma mērķis ir noteikt efektīvāko veidu, kā prognozēt Bitcoin cenu, un iegūt dziļāku izpratni par faktoriem, kas ietekmē prognozēšanas modeļu veikspēju šajā kontekstā. Darbs sākas ar literatūras analīzi, izpētot dažādus Bitcoin aspektus, tostarp tā pamatprincipus, ieguves procesu un tirgus dinamiku. Tie tiek pārbaudīti mašīnmācīšanās principi, koncentrējoties uz uzraudzītu mācīšanos un dziļo mācīšanos. Šim konkrētajam prognozēšanas uzdevumam tika apmācīti un pārbaudīti dažādi modeļi. Hiperparametru regulēšanai pētījumā tika implementēts Bajesa optimizācijas algoritms. Darbs noslēdzas ar atziņām par to, kā prognozēt Bitcoin cenu, izmantojot dažādus mašīnmācīšanās algoritmus, un faktoriem, kas ietekmē modeļu veikspēju.

Dati par darba apjomu – 56 lappušu skaits, 26 attēlu skaits, 6 tabulu skaits, un 15 izmantoto informācijas avotu skaits.

ABSTRACT

Keys: Bitcoin, Deep Learning, Bayesian optimization, stock trading

“A Comparative Study of Machine Learning Algorithms for Bitcoin Price Prediction” aims to determine the most effective way to predict the price of Bitcoin and to gain a deeper understanding of the factors that influence the performance of forecasting models in this context. The thesis starts with a literature analysis, exploring various facets of Bitcoin, including its foundational principles, mining process, and market dynamics. Machine learning principles are examined, with a focus on supervised learning and deep learning. Different models were trained and tested for this specific forecasting task. For hyperparameter tuning, the study implemented a Bayesian optimization algorithm. The paper culminates with insights into how to predict the price of Bitcoin using various machine learning algorithms and the factors that affect the performance of the models.

Data on the volume of work – 56 pages, 26 images, 6 tables, and 15 sources of information.

SATURA RĀDĪTĀJS

IEVADS.....	6
1. LITERATŪRAS ANALIZĒ.....	7
1.1. Bitcoin papīrs	7
1.2. Bitcoin ieguve	8
1.3. Bitcoin tirgus.....	9
1.4. Tirdzniecības analīze	10
1.4.1. Tehniskā analīze	10
1.4.2. Fundamentālā analīze	12
1.4.3. Blockchain fundamentālie dati	13
1.5. Laika rindas.....	14
1.6. Pētījuma mainīgais.....	15
1.7. Mašīnmācīšanās	16
1.7.1. Uzraudzītā mācīšanās	17
1.7.2. Arima	18
1.8. Dziļā mācīšanās	18
1.8.1. Neironu tīkla mācīšanās.....	21
1.8.2. Konvolūcijas Neironu Tīkls.....	22
1.8.3. Atkārtotie neironu tīkli	23
1.8.4. LSTM.....	24
1.8.5. Uzmanība	25
1.9. Bajesa optimizācija	26
1.10. Saistītie darbi	26
2. PRAKTISKAIS DARBS	29
2.1. Datu apkopošana.....	29
2.2. Train / Test sadalījums.....	31
2.3. Pamata līnija un metrikas.....	31
2.4. Naīva prognoze	32
2.5. Datu izpratne.....	33
2.6. GPU	36
2.7. Logu izveide	37
2.8. Modeļa apmācība.....	38
2.9. Dažādas modeļu arhitektūras	39

2.10. Stacionaritātes analīze.....	40
2.11. Daudzdimesionāla laika sēriju prognoze	41
2.11.1. Tehniskā analīze	42
2.11.2. Fundamentālā analīze	42
2.12. Apzīmējumi: apmācība pirmie modeļi ar daudzmainīgu laika intervālu, izmantojot dažādas aktivācijas funkcijas	43
2.12.1. Relu.....	44
2.12.2. GELU.....	45
2.12.3. ELU.....	45
2.12.4. Tanh	46
2.13. CNN modelis	46
2.14. Bajesa optimizācija	47
2.15. Vairākām izvades prognozēšana.....	50
2.16. Nākotnes darbs.....	52
REZULTĀTI UN SECINĀJUMI.....	54

IEVADS

Laikrindu prognozēšana ir labi izpētīta joma, īpaši nobriedušos finanšu tirgos, piemēram, akciju tirgū. Tomēr Bitcoin rada unikālu izaicinājumu kā laikrindu prognozēšanas problēmu nestabilā, pārejošā tirgū bez skaidrām sezonālām tendencēm. Tradicionālās metodes, piemēram, Holt-Winters eksponenciālās izlīdzināšanas modeļi, nav ļoti efektīvas Bitcoin cenu prognozēšanai. Turklāt Bitcoin decentralizētais, caurspīdīgais raksturs rada paradigmas maiņu no tradicionālajiem finanšu tirgiem.

Akciju tirgum ir liela loma pasaules ekonomikā, tas var mainīt tirdzniecības un rūpniecības izaugsmi abos virzienos. Akciju cenu kustību prognozēšana ir sarežģīts uzdevums un daudz faktoru mijiedarbības dēļ šāda rīka izveides ir neticami grūts. Mašīnmācība varētu sniegt priekšrocības, ātri nosakot tirgus modeļus un cenas virzienu varbūtību, tomēr, viena no galvenajām problēmām laikrindu analīzē ir jēdzienu novirze, kad modeļi laika gaitā mainās, pircējiem un pārdevējiem pielāgojot savas stratēģijas.

Viena no mašīnmācīšanas nodalām ir - padziļināta mācīšanās. Dziļās mācīšanās modeļi ir parādījuši savas iespējas analizēt milzīgus datu apjomus un identificēt sarežģītās un nelineārās regularitātes, padarot tos par ideālu kandidātu BTC prognozēšanai, izmantojot valūtas vēsturiskos datus.

Bakalaura darba mērķis ir izveidot modeli, kas var prognozēt nākamo BTC cenu, pamatojoties uz iepriekšējiem vēsturiskajiem datiem, lai palīdzētu tirgotājiem pieņemt lēmumus kuri veda peļņas palielināšanai, analizēt prognozēšanas precizitātes uzlabošanas veidus, izmantot dažādas dziļās mācīšanās arhitektūras un pārbaudīt Bayese optimizācijas algoritmu hiperparametru regulēšanai.

Šeit ir veicamo uzdevumu saraksts:

- 1) **Izpildīt literatūras analīzi** par cenu tirgu, par Bitcoin, par esošiem mašīnmācīšanās algoritmu pielietojumu laikrindas prognozēšanai.
- 2) **Apkopot un apstrādāt Bitcoin cenas vēsturiskos datus**
- 3) **Izveidot bāzes līniju**
- 4) **Izveidot prediktorus**
- 5) **Hiperparametru regulēšanai izmantot Bayese optimizācijas algoritmu.**
- 6) **Izveidot dziļās mācīšanās modeļu ansamblu.**
- 7) **Pārspēt bāzes līniju.**
- 8) **Izveidot prognozi nākotnē.**

1. LITERATŪRAS ANALIZĒ

1.1. Bitcoin papīrs

Bitcoin ir tīri Peer-to-Peer (P2P) elektroniskās skaidras naudas sistēma, kas ļauj tiešsaistes maksājumus nosūtīt tieši no vienas puses uz otru, neizmantojot finanšu institūciju. Ierosinātais dubulto tēriņu problēmas risinājums ietver P2P tīkla izmantošanu, kas uzliek transakciju laikspiedolu, iejaucot tos nepārtrauktā, uz jaukšanu balstītā darba pierādījuma ķēdē, veidojot ierakstu, ko nevar mainīt bez atkārtotas pārbaudes. no darba. Garākā ķēde kalpo ne tikai kā pierādījums notikušo notikumu secībai, bet arī pierādījums tam, ka tā nāk no lielākā centrālā procesora jaudas. Pašam tīklam ir nepieciešama minimāla struktūra, un mezgli var atstāt tīklu un atkal pievienoties tam pēc vēlēšanās, pieņemot garāko darba pierādījumu ķēdi kā pierādījumu tam, kas notika, kamēr tie nebija. Satoši Nakamoto rakstā "Bitcoin: A Peer-to-Peer Electronic Cash System" ir uzsvērtā nepieciešamība pēc elektroniskas maksājumu sistēmas, kuras pamatā ir kriptogrāfisks pierādījums, nevis uzticēšanās, ļaujot divām ieinteresētajām pusēm veikt tiešus darījumus savā starpā bez nepieciešamības uzticama trešā puse. Ierosinātā sistēma ir droša, kamēr godīgi mezgli kopīgi kontrolē vairāk CPU jaudas nekā jebkura uzbrucēju mezglu grupa, kas sadarbojas.

Satoshi Nakamoto apspriež darba pārbaudes sistēmas izmantošanu, lai ieviestu izplatītu laikspiedolu serveri P2P režīmā. Darba pārbaude ietver tādas vērtības skenēšanu, kuras jaušana, piemēram, ar SHA-256, sākas ar vairākiem nulles bitiem. Vidējais nepieciešamais darbs ir eksponenciāls nepieciešamo nulles bitu skaitā, un to var pārbaudīt, izpildot vienu jaucēju. Darba pierādījums tiek īstenots, palielinot nonce (nejaušs skaitlis, kas ir iekļauts bloka galvenē, lai mēģinātu ģenerēt derīgu hash blokam) blokā, līdz tiek atrasta vērtība, kas piešķir bloka jaušanai nepieciešamos nulles bitus. Viņš arī skaidro, ka vairākuma lēmumu pārstāv garākā ķēde, kurā ir ieguldīts vislielākais darba pierādījums. Stimuls palikt godīgam tiek radīts, nepārtraukti pievienojot nemainīgu daudzumu jaunu monētu, kas ir līdzīgi zelta ieguvējiem, kas tērē resursus, lai pievienotu zeltu apgrozībā. Stimuls var palīdzēt mudināt mezglus palikt godīgiem un novērst uzbrukumus no tiem, kuriem var būt vairāk CPU jaudas nekā godīgiem mezgliem.

Satoshi Nakamoto uzskata, ka iespēja pārbaudīt maksājumus, nepalaižot pilnu tīkla mezglu, saglabājot garākās darba apliecinājuma ķēdes bloku galvenes kopijas un iegūstot Merkle filiāli, kas savieno darījumu ar bloku, kurā tas ir laikspiedols. Vienkāršotā metode ir uzticama tik ilgi, kamēr godīgi mezgli kontrolē tīklu, taču tā ir neaizsargātāka, ja tīklu pārspēj uzbrucējs. Lai aizsargātos pret to, uzņēmumi, kas saņem biežus maksājumus, iespējams, joprojām vēlēšies palaist savus mezglus, lai nodrošinātu neatkarīgāku drošību un ātrāku verifikāciju. Darījumos ir ietvertas vairākas ievades un izvades, lai vērtību varētu sadalīt un apvienot. Tradicionālais banku modelis nodrošina privātumu, ierobežojot piekļuvi informācijai iesaistītajām pusēm un uzticamai trešajai pusei, bet Bitcoin privātumu var saglabāt, pārtraucot informācijas plūsmu citā vietā, saglabājot publiskās atslēgas anonīmas un izmantojot jaunu atslēgu pāri. katram darījumam, lai novērstu to saistību ar kopīgu īpašnieku.

Satoshi Nakamoto savā dokumentā ierosina sistēmu elektroniskiem darījumiem, nepaļaujoties uz uzticēšanos. Sistēma ir balstīta uz P2P tīklu, kas izmanto darba pierādījumu, lai reģistrētu publisku darījumu vēsturi, ko uzbrucējam ātri kļūst nepraktiski mainīt, ja godīgi mezgli kontrolē lielāko daļu CPU jaudas. Tīkls ir stabils savā nestrukturētajā vienkāršībā, un mezgli darbojas vienlaikus ar nelielu koordināciju. Mezgli var atstāt tīklu un atkal pievienoties tīklam pēc vēlēšanās, pieņemot darba pierādījumu ķēdi kā pierādījumu tam, kas notika, kamēr tie nebija. Viņi balso ar savu CPU jaudu, paužot savu akceptu derīgiem blokiem, strādājot pie to paplašināšanas un noraidot nederīgos blokus, atsakoties strādāt pie tiem. Ar šo vienprātības mehānismu var īstenot visus nepieciešamos noteikumus un stimulus. Ierosinātā sistēma nodrošina stingru īpašumtiesību kontroli un novērš dubultu tēriņu izmantošanu, nepaļaujoties uz uzticamu trešo pusi. (Satoshi Nakamoto “Bitcoin: A Peer-to-Peer Electronic Cash System”)

1.2. Bitcoin ieguve

Bitcoin ieguve ir process, kurā tiek verificētas blockchain transakcijās, radot kriptogrāfisku risinājumu, kas atbilst noteiktiem kritērijiem. Ieguvēji saņem atlīdzību noteiktā token daudzumā un maksa par viņu darbu, bet atlīdzība laika gaitā samazinās, līdz tā galu galā izbeidzas, kad aprītē ir 21 miljons Bitcoin un tad ieguvēji saņems atlīdzību tikai no transakcijās maksām. Bitcoin ieguves sirds ir hash - 64 zīmju heksadecimālais skaitlis, ko rada SHA256 hashing algoritms. Mērķa hash, ko izmanto,

lai noteiktu ieguves grūtības, ir skaitlis, ko ieguvēji mēģina atrisināt. Ieguves programmu ģenerē nejaušu hash un pievieno skaitli, ko sauc par nonce, kas mainās par vienu katrā mēģinājumā, līdz tiek radīta hash un nonce kombinācija, kas ir mazāka vai vienāda ar mērķa hash. Kad pareizs risinājums tiek atrasts, ieguvējs, kas to atrada, saņem atlīdzību un maksu, un tiek atvērts jauns bloks. Ieguves process tiek saukts par darba pierādīšanu, un tā grūtību līmenis mainās katru 2 016 blokus, jeb aptuveni ik pēc divām nedēļām. Ieguves process ir radījis pretrunas sakarā ar tā ietekmi uz vidi, bet tas ir nepieciešams, lai uzturētu blockchain integritāti.

Šis ir biznesa projekts, kas gūst peļņu no sava produkta, Bitcoin. Tomēr BTC ieguves izmaksas ir augstas un ietver elektroenerģiju, dzesēšanu, ieguves sistēmas un tīkla infrastruktūru. Elektroenerģijas izmaksas var būt ievērojamas, jo process patērē tik daudz elektroenerģijas, cik to dara dažas valstis. Ieguves sistēmas, kas nepieciešamas efektīvai ieguvei, ir dārgas, un baseina pievienošana var izmaksāt no 4 000 līdz 12 000 dolāriem par ierīci. Turklāt nepieciešama tīkla infrastruktūra, bet latencija ir svarīgāka nekā tīkla ātrums. Lai gūtu peļņu, šo ieejas datu kopējām izmaksām jābūt mazākām par izejvielu, Bitcoin cenu. Neskatoties uz Bitcoin cenas svārstībām, ekonomiskās grūtības Bitcoin ieguvējiem var novest pie zemākas peļņas un ilgākiem laikiem, lai atgūtu ieguldījumu. (Jake Frankenfield. "Bitcoin Ieguve").

Bitcoin ieguvei un tīklam ir savi parametri, un šie parametri var ietekmēt akciju cenu, tie ir jāuzskata par datu pazīmēm un jāiekļauj galīgajā datu kopā.

1.3. Bitcoin tirgus

Bitcoin var iegādāties un pārdot dažādās kriptovalūtu biržās, piemēram, Binance, Coinmetrics, Kraken, HitBTC, Bittrex utt., un to var glabāt digitālajos maciņos. Šis kryptoaktīvs kļuva ļoti populārs ar milzīgu globālo tirgus kapitalizāciju. BTC ir tāds pats kā akcija, to var tirgot biržās vai vienkārši turēt aukstajā maciņā. Ir vēl viens veids, kā iegūt BTC - tas ir P2P apmaiņa (Peer-to-Peer vai Person-to-Person), šādā veidā tā ir vienkārša tirdzniecība starp divām personām. Ir dažāda veida cilvēki, kas tur vai tirgo kriptovalūtu, un katrs no viņiem rada unikālu ekosistēmu, lai krypto cenas palielinātos vai samazinātos. Ieguvēji ir visas algoritma un sistēmas pamats. Ieguve patērē lielu daudzumu elektrības, tāpēc enerģijas cena, tās pieprasījums un piedāvājums var tikt iekļauti arī funkciju datu kopā. BTC aktīvu cena tiek noteikta pēc kopējā tirgus noskaņojuma, kā arī dažādu faktoru, piemēram, tirgus sentimenta,

regulatīvo izmaiņu un tehnoloģisko steku. Ziņas var ietekmēt BTC cenu un izraisīt pozitīvu vai negatīvu sentimentu starp investoriem un tirgotājiem, kas pēc tam novedīs pie cenu kustības. Sentimenta analīze ir viena no galvenajām akciju prognozēšanas stratēģijām, tomēr šajā rakstā autors to neanalizē un to nepievieno kā prognozes argumentu gala modelim, taču to var izdarīt turpmākajā darbā, lai vēl vairāk palielinātu precizitāti. Viena pieeja ir izveidot Twitter sentimenta analīzi, izmantojot kādu lielās valodas modeli (LLM), un iegūt kopējo cilvēku noskaņojumu konkrētā datumā ar #btc vai #bitcoin hashtagu.

1.4. Tirdzniecības analīze

Ir četras galvenās akciju prognozēšanas metodes: fundamentālā analīze, tehniskā analīze, sentimenta analīze un kvantitatīvā tehniskā analīze. Fundamentālā analīze koncentrējas uz ekonomiskiem faktoriem, piemēram, uzņēmuma bilanci un peļņas un zaudējumu pārskatu, lai prognozētu ilgtermiņa cenu kustību. Tehniskā analīze balstās uz akciju cenu un apjoma vēsturisko datu analīzi, lai prognozētu nākotnes cenas, un praktiski mēģina izsecināt modeļus no šiem datiem. Kvantitatīvā tehniskā analīze, no otras puses, balstās uz mašīnmācīšanās algoritmiem, lai, izmantojot kvantitatīvas metodes, prognozētu nākotnes akciju cenas. Kamēr fundamentālā un tehniskā analīze lielā mērā ir kvalitatīva, kvantitatīvā tehniskā analīze ir vairāk vērsta uz kvantitatīvām prognozēšanas metodēm. Kā ir skaidrs, šis darbs koncentrējas uz labas kvantitatīvās tehniskās analīzes rīka izveidi. Autors izvēlējās izmantot tehnisko analīzi un fundamentālo analīzi kā pamatu BTC prognozēšanai (Lisa Hladik "Akciju tirgus prognozēšanas metodes").

1.4.1. Tehniskā analīze

Tehniskajā analīzē (TA) tirgus darbības izpēti galvenokārt tiek veikta, izmantojot diagrammas, lai prognozētu nākotnes cenu tendences. Termins "tirgus darbība" ietver cenu, apjomu un atvērto interesi. Tehniskās analīzes filozofija (saskaņā ar John J Murphy – "Technical Analysis Of The Financial Markets") balstās uz trim pieņēmumiem: **tirgus darbība atspoguļo visu, cenas kustas tendencēs, un vēsture atkārtojas**. Ir svarīgi saprast šos pieņēmumus, lai pilnībā novērtētu tehnisko pieeju un tās apgalvojumus par nākotnes cenu tendenču prognozēšanu. TA atšķiras no fundamentālās analīzes un dažreiz tiek izvirzīti apgalvojumi pret tehnisko pieeju.

Princips "tirgus darbība atspoguļo visu" ir viens no TA pamatiem. Tas nozīmē, ka jebkurš faktors, kas var ietekmēt tirgus cenu, piemēram, politiski vai psiholoģiski faktori, jau ir atspoguļots tirgus cenā. Tāpēc tehniskie analītiķi uzskata, ka pietiek ar cenu kustību izpēti, lai prognozētu tirgus tendences. Tehniskie rīki un indikatori tiek izmantoti, lai atbalstītu tirgus darbības analīzi ar mērķi noteikt nākotnes cenu kustības virzienu. Viens no galvenajiem tirgus cenu kustības principiem ir **atbalsta un pretestības koncepcija** (1.1. attēls), zilā apakšējā līnija ir atbalsts un zilā augšējā līnija ir pretestība. Šīs koncepcijas īstenošana un tās datu sniegšana modelim var būtiski uzlabot prognozēšanas precizitāti.



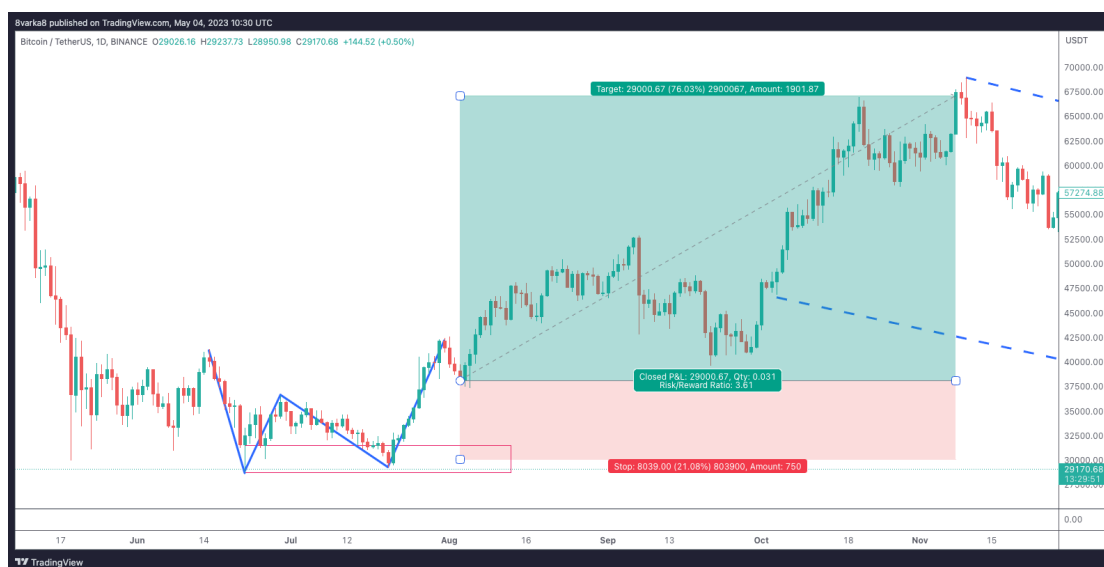
1.1 att. BTC diagramma ar atbalsta un pretestības līnijām (tradingview.com)

Tendences koncepcija ir ļoti svarīga, lai saprastu cenu kustības. Cenu kustības vizualizācijas mērķis ir identificēt tendences to attīstības sākuma stadijās, lai tirgoties tā virzienā. 1.1. attēlā ir parādīts BTC “downtrend”, kas sākās 09.11.2021 un beidzās 09.12.2022. Šie dati tiks iekļauti datu kopā. Lielākā daļa šīs pieejas tehniku ir tendenču sekotājas pēc dabas. Turklāt šī pieņēmuma korolārijs ir tāds, ka kustībā esoša tendence ir vairāk sagaidāma turpinājumu nekā atgriešanos, kas balstās uz Ņūtona pirmo kustības likumu. Tātad tendenču sekotājdarbība ietver esošās tendences jāšanu, līdz tā parāda zīmes par atgriešanos.

TA un tirgus darbības izpēte ietver cilvēka psiholoģijas izpēti tādā veidā, ka daži cenu diagrammu modeļi ir identificēti un analizēti. Tie paši diagrammu modeļi, kas ir būtībā noteiktas attēlu izpausmes cenu diagrammās, var atklāt tirgus “bullish” vai

“bearish” psiholoģiju, kas nozīmē, ka tā var parādīt un identificēt nākotnes cenu tendenci.

Tā kā šie modeļi ir parādījuši tādu pašu cenu uzvedību pēc to parādīšanās diagrammā pagātnē, tiek pieņemts, ka tie turpmāk darbosies labi. Vēsture atkārtojas un tas ir arī svarīgs elements TA, kas nozīmē, ka nākotne ir tikai pagātnes atkārtojums un tas ir līdzīgs tam, ko var izdarīt neironu tīkli. Trenēties uz vēsturisko datu un tad rādīt dažus rezultātus nākotnes prognozēšanā. (John J Murphy – “Technical Analysis Of The Financial Markets”)



1.2. att. BTC diagramma ar Double bottom modeli. (tradingview.com)

1.2. Attēlā parādīts, kā izskatās parastais double bottom diagrammas modelis. Modelis ir uzzīmēts zilā krāsā un pēc tā parādīšanās diagrammā, BTC cena palielinājās par 76%.

1.4.2. Fundamentālā analīze

Fundamentālā analīze (FA) ir vēl viens rīks, ko drīkst izmantot tirgus analīzē, kas koncentrējas uz ekonomiskiem faktoriem un tehnoloģijas vai uzņēmuma pamatā esošiem mehānismiem, kas var ietekmēt investoru lēmumus un cenu tendences. Izpētot visus relevantos faktorus, kas ietekmē tirgus cenas, fundamentālisti mēģina noteikt tirgus iekšējo vērtību. Ja iekšējā vērtība ir augstāka par pašreizējo tirgus cenu, tirgus tiek uzskatīts par novērtētu un to vajadzētu pirkt, savukārt, ja iekšējā vērtība ir zemāka, tirgus tiek uzskatīts par pārvērtētu un to vajadzētu pārdot. Atšķirībā no tehniskās analīzes, kas koncentrējas uz tirgus darbības un diagrammu modeļu izpēti, FA mēģina

izprast iemeslus tirgus kustībām. Kamēr tehniskie analītiķi apgalvo, ka tirgus cenas izpēte ir pietiekama, jo fundamentālie faktori atspoguļojas tirgus cenā, fundamentālisti uzstāj uz pamatīgu cenu kustību iemeslu izpratni. Lai gan šīs divas pieejas bieži vien ir pretrunā, tās abas ir vērstas uz nākotnes cenu kustību prognozēšanu un tās abas tiek izmantotas, lai tirgotāji pieņemtu informētus lēmumus. (John J Murphy – “Technical Analysis Of The Financial Markets”)

1.4.3. Blockchain fundamentālie dati

“Blockchain” ir decentralizēta un nemainīga digitāla grāmatvedība, kas droši un caurskatāmi reģistrē darījumus. Kad blockchain tiek inicializēts, lielākā daļa no šiem parametriem nevar tikt mainīti, tomēr dažus parametrus var atjaunināt, izmantojot blockchain atjaunināšanas mehānismu. Parametru hash tiek pievienots ķēdei, kad tā sāk darboties, lai novērstu nejaušu modificēšanu.

Blockchain parametri ir grupēti sadaļās un ietver pamata ķēdes parametrus, globālās atļaujas, konsensa prasības, noklusējuma vērtības konsensa saistītiem izpildlaika parametriem, iedzimto blockchain valūtu, uzlabotus “mining” parametrus un darījumu ierobežojumus. Pamata ķēdes parametri nosaka blockchain protokolu, tā tekstuālo aprakstu un sākņu straumes nosaukumu vispārīgai datu glabāšanai, starp citiem. Globālās atļaujas regulē tīkla ierobežojumus saistībā ar savienojumu, sūtīšanu un darījumu saņemšanu, kā arī jaunu straume izveidi, starp citiem. Konsensa prasības ietver sākotnējās uzstādīšanas fāzes garumu blokos, minimālo atļauto “mining” daļu, kas nepieciešama, lai piedalītos “round-robin” shēmā, un atļauto administratoru proporciju, kuriem jāpiekrīt blockchain protokola atjaunināšanai, starp citiem. Noklusējuma vērtības konsensa saistītajiem izpildlaika parametriem ietver atsacīšanos no kļūdu kontroles, kas pārveido admin vai “mining” atļauju izmaiņas pēc noteikta kārtu skaita, un bloku skaitu līdz bloka atalgojumam var tikt iztērētam, starp citiem. Uzlabotie “mining” parametri ietver pārbaudes izlaišanu, vai bloku hash parāda pietiekamu darbaspēka pierādījumu, minimālo/sākotnējo darbaspēka grūtības pakāpi un darbaspēka grūtības pakāpes pārrēķināšanas biežumu. Darījumu ierobežojumi regulē standarta darījumu kritēriju atbilstības un nodošanas akceptēšanu, kā arī maksimālo standarta darījumu izmēru, starp citiem. (Multichain – “Customizing blockchain parameters”)

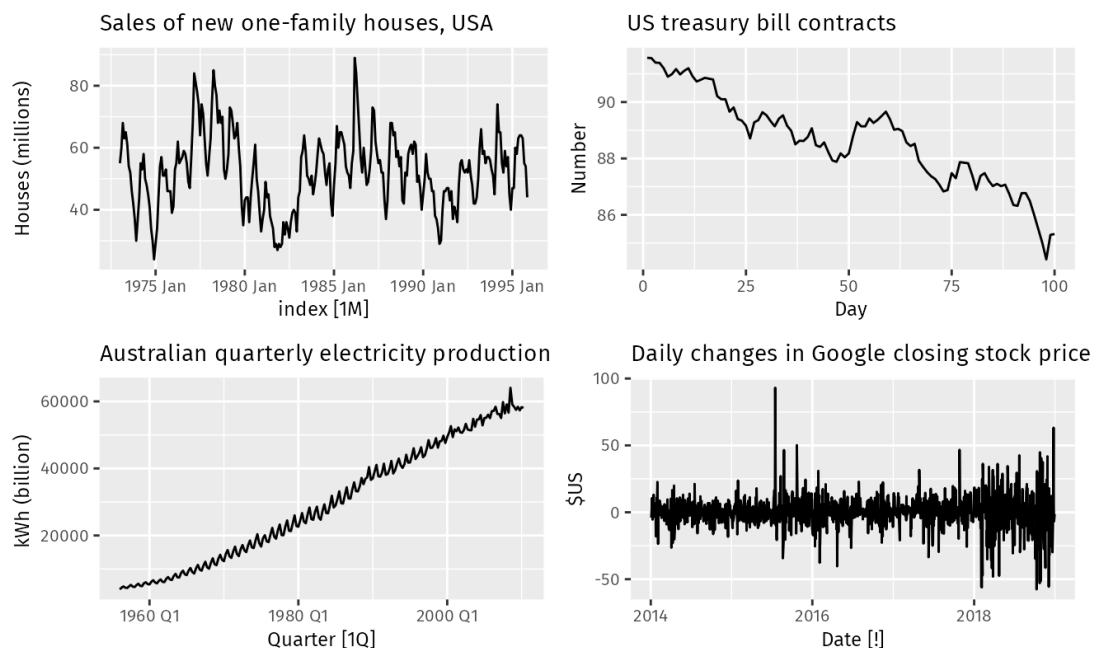
Runājot par BTC blockchain, autors var izcelt dažus vērtīgus parametrus, piemēram, tirgus kapitalizāciju, darījumu skaitu sekundē, kopējo darījumu skaitu konkrētā svečturī, unikālo adrešu skaitu utt. Precīzāks skatījums uz šiem fundamentālajiem rādītājiem ir iekļauts 2. praktiskajā daļā, fundamentālā datu bāzē.

1.5. Laika rindas

Laika rindu dati attiecas uz novērojumu kopumu, ko iegūst, veicot atkārtotus mērījumus laika gaitā, viena no asīm vienmēr ir laiks. Laika rindu dati ir visur, un tiem ir daudz pielietojumu dažādās nozarēs. Laika rindu grafiki vienkārši attēlo laika rindu datus vienā asī pret laiku otrā asī, ilustrējot tendences vai modeļus daudz pieejamākā un intuitīvākā veidā. Laika rindu datus var klasificēt divos veidos: mērījumi, kas veikti regulāros laika intervālos (metrikas), un mērījumi, kas veikti neregulāros laika intervālos (notikumi). Laika rindu dati ir unikāls datu tips, tiem ir dabiska laika kārtība, un bieži tie parāda seriālo atkarību. Lineāros laika rindu datus var uzskatīt par lineāru kombināciju no pagātnes vai nākotnes vērtībām, kamēr nelineāras laika rindas rodas no nelineārām dinamiskām vienādojumu sistēmām. Laika rindu datus izmanto laika rindu analīzē un laika rindu prognozēšanā, lai noteiktu un prognozētu modeļus, un tos bieži glabā mērķtiecīgi izveidotā datubāzē, kas paredzēta, lai tikt galā ar tā mērogu.

Laika rindu analīze palīdz ieguldītājiem saprast tendences un modeļus laika gaitā, identificējot un analizējot datus konsekventos intervālos. Laika rindu datus var klasificēt par krājuma un plūsmas datiem, un variācijas var notikt sporādiski visā datu kopumā. Laika rindu analīzē tiek izmantoti dažādi modeļi un tehniskās metodes, piemēram, Box-Jenkins ARIMA modeļi, Box-Jenkins multivariāti modeļi un Holt-Winters metode. Visbeidzot, Python programmēšanas valoda tiek bieži izmantota laika rindu analīzē, lai atrastu tendences un prognozētu nākotnes notikumus. (tableau.com “Time Series Analysis: Definition, Types, Techniques, and When It's Used”)

Laika rindu datiem ir savas raksturīgās iezīmes: *tendence, sezonālums un cikliski modeļi*, katram ir savi unikāli raksturlielumi. **Tendence** vai (“**trend**”) attiecas uz ilgtermiņa palielināšanos vai samazināšanos datu kopumā, kamēr **sezonālumu** ietekmē sezonālie faktori, piemēram, gada laiks vai nedēļas diena. **Cikliskā uzvedība** parāda celienus un kritumus, kuriem nav noteiktas frekvences un kas bieži ir saistīti ar ekonomiskiem apstākļiem. Daudzās laika rindās ir iekļauti visi trīs modeļi, un izvēloties prognozēšanas metodi, ir būtiski precīzi identificēt laika rindu modeļus datu kopumā.



1.3. att. Dažādas laika rindas (<https://otexts.com/fpp3/tspatterns.html>)

Ikmēneša mājokļu pārdošanas apjomi uzrāda spēcīgu sezonālītāti un ciklisku darbību bez redzamām tendencēm, savukārt ASV valsts parādzīmju līgumi īsā laika periodā uzrāda lejupejošu tendenci. Austrālijas ceturkšņa elektroenerģijas ražošanai ir spēcīga pieauguma tendence ar sezonālītāti, un ikdienas Google slēgšanas akciju cenas izmaiņām nav manāmas tendences. (1.3. attēls) (Rob J Hyndman, George Athanasopoulos “Forecasting: Principles and Practice”)

1.6. Pētījuma mainīgais

Šī pētījuma galvenais neatkarīgais mainīgais ir BTC slēgšanas cena ASV dolāros, kas ņemta no Coinbase btc/usd tirdzniecības tirgus katru dienu kopš 2016. gada 2. janvāra, un pēdējais datu kopas datums ir 2023. gada 21. janvāris. Cenas datu punkti un FA (fundamentālas analīzes) funkcijas ņemtas no coinmetrics, izmantojot coinmetrics.api_client Google Colab piezīmju grāmatā, kas izmantota Python programmēšanas valodā. Dažas funkcijas ir izveidotas no galvenajiem mainīgajiem: Price_Close, Price_Open, Price_High, Price_Low un Volume. Pētījumā ir testētas dažādas prognozēšanas metodes, un vienā no tām, lai prognozētu X dienas nākotnē, katras X_i dienas funkcijas vektors ir jāizveido no galvenajām mērķa vērtībām. Precīzāki

tehniski dati par datu kopas veidošanu un mainīgu inženierijas procesu ir iekļauti šī darba praktiskajā daļā.

1.7 Mašīnmācīšanās

Mašīnmācīšanās (ML) ir Mākslīgās intelektuālās spējas apakškatēgorija. ML palīdz mašīnām saprast tā, kā to dara cilvēki. Mašīnmācīšanās ir AI pielietojums, kas ļauj sistēmām mācīties un uzlaboties no pieredzes bez skaidras programmēšanas. Tas nozīmē, ka datori var piekļūt datiem un tos izmantot, lai mācītos pats par sevi, koncentrējoties uz datorprogrammu izstrādi, kas var mācīties autonomi bez cilvēka iejaukšanās vai palīdzības un pielāgot darbības atbilstoši. Mašīnmācīšanās process sākas ar datu novērošanu, tiešām pieredzēm vai instrukcijām. Tā meklē datu iekšējas struktūras, lai vēlāk varētu izdarīt secinājumus, pamatojoties uz dotajiem piemēriem. ML algoritmi veido matemātisku modeli, pamatojoties uz parauga datiem, kas pazīstami kā “apmācības dati”, lai izdarītu prognozes vai lēmumus, pamatojoties uz iemācītajiem modeļiem.

Pieejot datiem, ML programmas var mācīties pašas un laika gaitā uzlabot savas spējas. ML jau ir pastāvējusi kādu laiku un ir pierādījusi savu vērtību, risinot problēmas ar ātrumu un mērogā. Spējot identificēt datus un automatizēt rutīnas procesus, mašīnas var uzlabot lēmumu pieņemšanu un optimizēt efektivitāti. Tas ir padarījis ML plaši pieņemtu visos biznesa sektoros, ieskaitot datu drošību, finanses, krāpšanas noteikšanu utt. Vadītā mācīšanās izmanto marķētus piemērus, lai prognozētu nākotnes notikumus, kamēr nevadītā mācīšanās pēta, kā sistēmas var secināt kādu funkciju, lai aprakstītu slēpto struktūru no nemarķētiem datiem. Pastiprināšanas mācīšanās mijiedarbojas ar savu vidi, radot darbības un atklājot kļūdas vai atalgojumus.

ML algoritmi var būt ļoti noderīgi, bet mācību un testēšanas laikā var rasties dažas problēmas. Modeļa precizitāte ir atkarīga no datu kvalitātes un datu apjoma, kas nozīmē, ka, ja datu kvalitāte ir zema, tad modelis generalizēs mēslu modeļus datos un reālā produkcijā nebūs labu rezultātu. Daži modeļi prasa daudz datora jaudu, lai trenētu noteiktu datu apjomu, jaunajiem Large Language Modeļiem (LLM) var būt vairāk nekā 100 miljardi parametri, un to trenēšana uz milzīgo teksta datu apjomu, tas var izmaksāt miljonus dolāru. Arī precīzi treniņu soļi un hiperparametru regulēšana ir ļoti sarežģīts process, pat pieredzējušiem datu zinātniekiem, tas ir grūts uzdevums. Vēl viena

problēma ir saistīta ar to, ka ML bieži ir aizspriedumaina, jo datu sadale un kopējais datu līdzsvars, kas nozīmē, ja datus kāda funkcija ir galvenokārt saistīta ar vienu klasi, tad modelis aizspriedumosies, ka šī funkcija ir saistīta ar šo konkrēto klasi. Runājot par LLM, vizizplatītākais piemērs ir vārds “ārsts”, kas galvenokārt ir saistīts ar sievieti, jo mācību tekstos ir vairāk sieviešu ārstu nekā vīriešu. Tas pats var notikt ar jebkuru datu kopu. (expert.ai “What Is Machine Learning? A Definition.”)

1.7.1 Uzraudzītā mācīšanās

Vadītā mācīšanās ir mašīnmācīšanās apakš kategorija, kas izmanto marķētus datu kopas, lai apmācītu algoritmus precīzi klasificēt datus vai veikt prognozes. Šis mācību kopums sastāv no ievaddatiem un pareiziem izvaddatiem, kas ļauj modelim mācīties laika gaitā. Izpildot zaudējumu funkciju, algoritms mēra savu precizitāti, uzlabojas, līdz kļūda ir minimizēta. Vadīto mācīšanos var iedalīt divās problēmu tipā: **klasifikācijā un regresijā.**

Klasifikācija izmanto algoritmus, lai iedalītu testa datus noteiktās kategorijās, savukārt regresija palīdz saprast atkarību starp atkarīgajiem un neatkarīgajiem mainīgajiem. Ierastie klasifikācijas algoritmi ietver loģistikas regresiju, atbalsta vektora mašīnas, lēmumu kokus, tuvāko kaimiņu metodi un nejaušo mežu. Tikmēr populāri regresijas algoritmi ietver lineāro regresiju un polinomiālo regresiju.

Vadītās mācīšanās algoritmi izmanto dažādas tehnikas, piemēram, neironu tīklus, Naive Bayes, lineāro regresiju, loģistikas regresiju, atbalsta vektora mašīnas, tuvāko kaimiņu metodi un nejaušo mežu. Neironu tīkli darbojas, apstrādājot mācību datus caur mezglu slāņiem, savukārt Naive Bayes ir klasifikācijas pieeja, kas pieņem klases nosacītās neatkarības principu. Lineārā regresija identificē attiecību starp atkarīgajiem un neatkarīgajiem mainīgajiem, savukārt loģistikas regresija tiek izmantota, kad atkarīgais mainīgais ir kategorisks. Atbalsta vektora mašīnas veido hiperplakni, kas atdala datu punktu klases, savukārt tuvāko kaimiņu metode klasificē datu punktus, pamatojoties uz to tuvumu citiem pieejamiem datiem. (IBM “What is supervised learning?”)

Šie algoritmi izmanto matemātiskus modeļus, kas nav tik noderīgi akciju cenu prognozēšanā, tāpēc to analīze un ieviešana nav iekļauta šajā pētījumā. Tā vietā autors koncentrējās uz dažādām dziļās mācīšanās pieejām BTC cenas prognozēšanas regresijas uzdevumam.

1.7.2 Arima

Autoregresīvais integrētais slīdošā vidējā (ARIMA) ir statistisks modelis, kas izmanto laika rindu datus, lai prognozētu nākotnes tendences vai labāk izprastu datu kopu. Modelis ir autoregresīvs, tas nozīmē, ka tas prognozē nākotnes vērtības, balstoties uz pagātnes vērtībām, piemēram, akciju cenām vai peļņu. ARIMA izmanto nobīdītos slīdošos vidējos, lai izlīdzinātu laika rindu datus, un to plaši izmanto tehniskajā analizē, lai prognozētu nākotnes drošības cenas. ARIMA modelis ir regresijas analīzes forma, kas novērtē vienas atkarīgās mainīgās stiprumu attiecībā pret citiem mainīgajiem mainīgajiem.

Modeļa mērķis ir prognozēt nākotnes vērtspapīru vai finanšu tirgus kustības, izpētot vērtību atšķirības sērijā, nevis caur faktiskām vērtībām. Katrs komponents ARIMA darbojas kā parametrs ar standarta notāciju, piemēram, p , d un q , norādot izmantotā ARIMA modeļa tipu. ARIMA modeļi ir sarežģīti un vislabāk strādā uz ļoti lielām datu kopām, prasa datoru algoritmus un mašīnmācīšanās tehnikas, lai tos aprēķinātu.

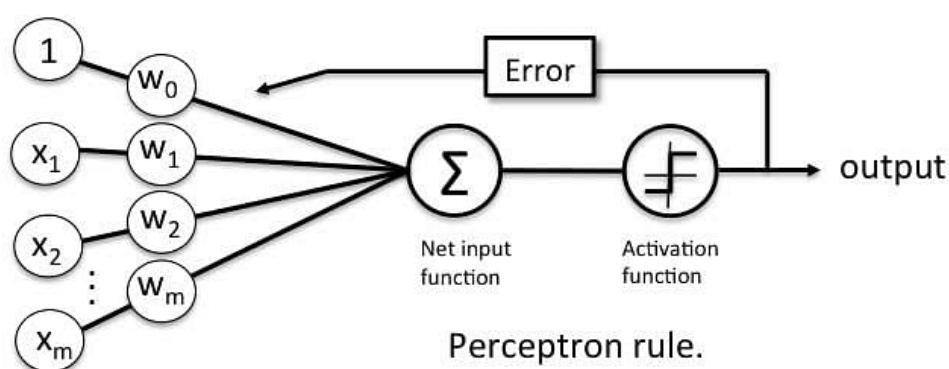
Autoregresīvajā integrētajā slīdošā vidējā modelī dati tiek atšķirīgi, lai tos padarītu stacionārus. Tomēr, ja stacionaritāte nav acīmredzama, daudzus aprēķinus procesa laikā nevar veikt un radīt paredzētos rezultātus. Lai gan ARIMA modeļiem ir stipras puses un tie ir labi īstermiņa prognozēšanā, tiem ir trūkumi, ieskaitot to, ka tie slikti prognozē pagrieziena punktus un ir datoriski dārgi. ARIMA tiek izmantots kā prognozēšanas rīks, lai prognozētu, kā kaut kas darbosies nākotnē, pamatojoties uz iepriekšējo veikspēju, bieži vien tehniskajā analizē, lai prognozētu aktīva nākotnes veikspēju. ARIMA modelēšana parasti ir nepietiekama ilgtermiņa prognozēšanai, piemēram, vairāk kā sešus mēnešus uz priekšu, jo tā izmanto pagātnes datus un parametrus, kas ir ietekmēti ar cilvēka domāšanu. Tāpēc to vislabāk izmantot kopā ar citiem tehniskās analīzes rīkiem, lai iegūtu skaidrāku priekšstatu par aktīva veikspēju. (Adam Hayes “Autoregressive Integrated Moving Average (ARIMA) Prediction Model”)

Šajā darbā ARIMA tiek izmantots kā konkurents autora prognozēšanas modelim, lai analizētu tā prognozēšanas precizitāti.

1.8. Dziļā mācīšanās

Deep Learning (DL) ir mašīnmācīšanās apakš kategorija. Lielākoties DL tiek asociēts ar neironu tīkliem (NN), jo šī matemātikas tehnoloģija ir tā, kas atrodas aiz DL ainas. Parasti DL ir ieejas slānis, slēptie slāņi un izvades slānis, atkarībā no problēmas specifikācijas datu tipi var atšķirties visos iekšējos slāņos.

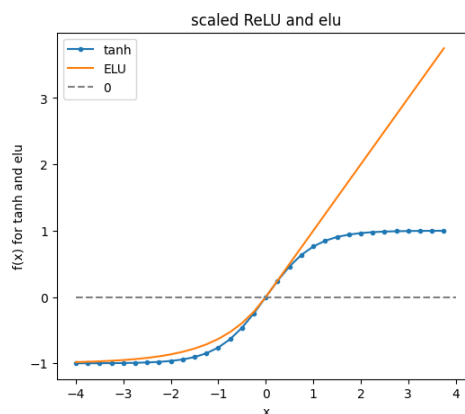
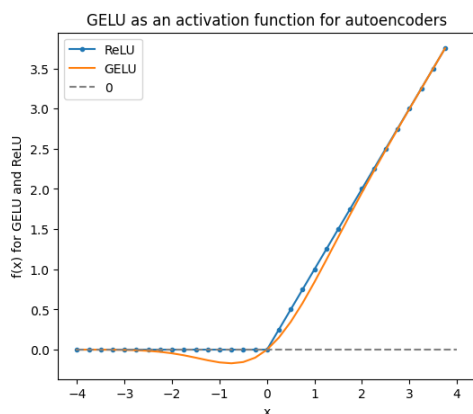
Galvenais datu tips, kas tiek pārveidots un pārvietots no iepriekšējā slāņa uz nākamo, ir tenzori. Tenzors ir datu tips, kas tiek saglabāts datora atmiņā. Tenzors ar 0 asīm ir parasts skaitlis, piemēram, "10", 1. līmeņa tenzors ir vektors "[0,1,2,3,4]". Otrā līmeņa tenzors ir matrica, utt.



1.4. att. Perceptrona struktūra. (Mayank Banoula <https://www.simplilearn.com>)

Katras NN pamatbloks ir perceptrons, kas ir ļoti līdzīgs smadzeņu neironam un ir visām tā īpašībām. Mākslīgais neirons ir matemātiskā funkcija, kas simulē smadzeņu darbības darbu. Tam ir ieejas slānis X_1, X_2, \dots, X_m , kas parasti reprezentē datu paraugu. Ieeja var būt iezīmju vektors vai attēla pikseļi, kas ir izlīdzināti uz vienas dimensijas vektoru, bet īpaši attēliem ir īpaša NN veida, ko sauc par konvolūcijas neironu tīklu (CNN). Katram neironam papildus tiek pievienots nobīde, kas ir parasts skaitlis, lai uzlabotu modeļa precizitāti. Kas ir redzams 1.4. attēlī. Ar skaitli "1". Tad katru ieejas vērtību X_i reizina ar svaru W_1, W_2, \dots, W_m attiecīgi. No katras ieejas līdz katram neironam ir savienojuma līnija, tā saucamā sinapse smadzenēs, lai pārsūtītu signālu no katras kārtas uz nākamo un beigās uz izvadi.

Tad katru pāri no ieejas tiek reizināts ar pieslēgto svaru un apstrādāts neirona (perceptrona) kodolā. Pirmkārt, visas ieejas*svērtu pāri tiek sasummēti, un tad šī summa iet cauri aktivācijas funkcijai. Ir dažādi aktivizācijas funkciju tipi, bet visplašāk izmantotais ir Rectified Linear Unit (ReLU), kas darbojas kā vārte. ReLU attēls ir redzams 1.5. attēlā kreisajā pusē zilā krāsā.



1.5. att. ReLU, GELU, Tanh, ELU aktivācijas funkcijas (pašizgatavots Colab)

```

import math
scale = 1.05070098
alpha = 1.67326324
def gelu(x):
    return 0.5 * x * (1 + math.tanh(math.sqrt(2 / math.pi) * (x + 0.044715 * math.pow(x, 3))))
def relu(x):
    return max(x, 0)
def tanh(x):
    return tf.keras.activations.tanh(x)
def elu(x):
    if x >= 0:
        return x
    else:
        return math.exp(x) - 1

```

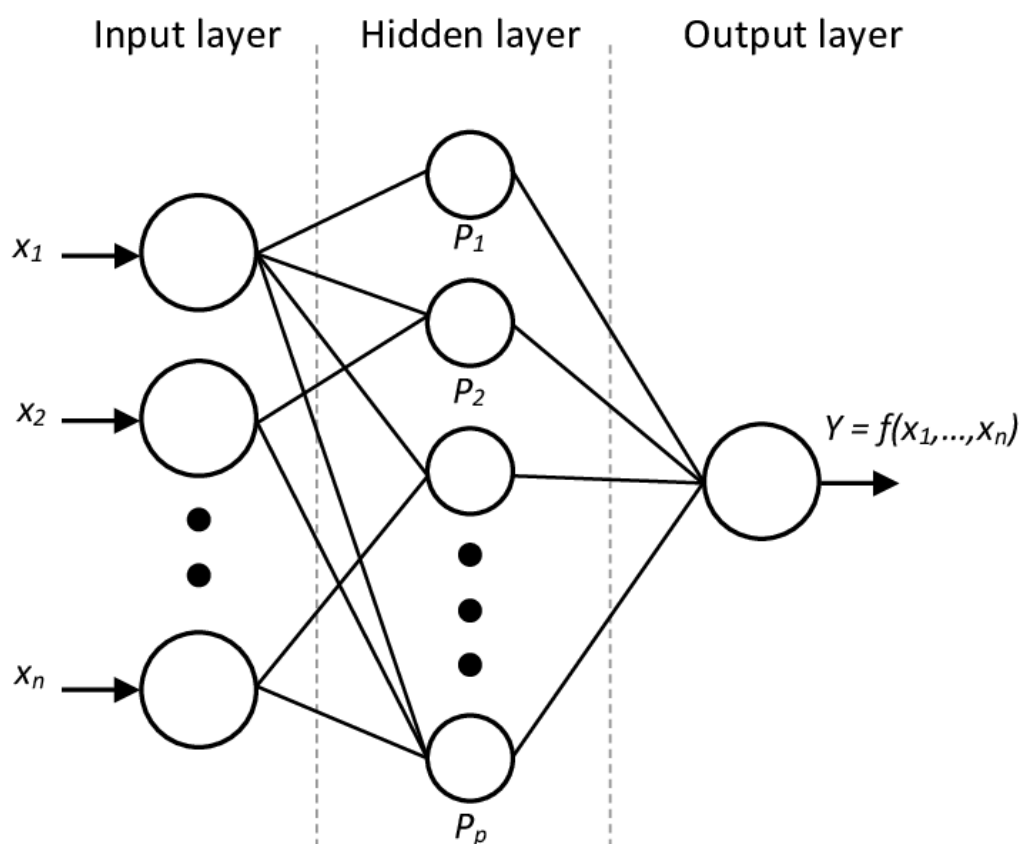
ReLU ir izplatīta aktivācijas funkcija, jo pirmkārt, tā ir ļoti vienkārša un izmaksu efektīva, un otrkārt, to ir ļoti viegli aprēķināt no tās atvasinājuma. Aktivācijas funkcijas ir nepieciešamas, jo tās ievieš nelinearitāti modelī, lai sasniegtu labāko veiktspēju datu regularizācijā.

Tātad vispārējā formula katram neironam izskatās šādi:

$$y = g(W_0 + X^T W) \quad (1.1)$$

Kur y - izvide, g - aktivācijas funkcija, W_0 - nobīde, X^T - ieejas transponēts vektors, lai ļautu matricas dot produktu ar svariem, W - svaru vektors.

Blīvi savienotie neironi ar kādu slēpto slāņu skaitu tiek saukti par “Blīvu neironu tīklu” un tādu pašu tīkla arhitektūru parāda 1.6. attēls.



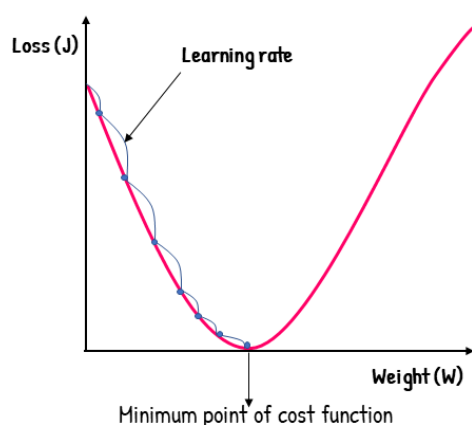
**1.6. att. Blīvs neironu tīkls ar vienu slēpto slāni.. (Yugesh Verma
<https://analyticsindiamag.com>)**

1.8.1. Neironu tīkla mācīšanās

Neironu tīkli parasti veic uzraudzītas mācīšanās uzdevumus, izveidojot zināšanas no datu kopām, kurās pareizā atbilde tiek sniegta iepriekš. Tīkli tad mācās, pielāgojot savas nobīdes un svarus, lai paši atrastu pareizo atbildi, uzlabojot savu prognožu precizitāti. Lai to sasniegtu, tīkls sākotnējos izvades datus salīdzina ar sniegto pareizo atbildi jeb mērķi. Izveidojot izmaksu funkciju, sākotnējie izvades dati tiek modificēti, pamatojoties uz to, cik lielā mērā tie atšķiras no mērķa vērtībām. Izmaksu funkcijas rezultāti tiek pārsūtīti atpakaļ visos neironos un savienojumos, lai pielāgotu nobīdes un svarus, izmantojot metodi, ko sauc par atpakaļizplatīšanu un gradienta lejupslīdi. Tās tiek izmantotas kopā, lai uzlabotu NN prognožu precizitāti, tām ir atšķirīgas lomas.

Atpakaļizplatīšana ir apmācības algoritms, kas tiek izmantots, lai uzlabotu taisnā neironu tīkla izvades prognozi, propagējot kļūdu atpakaļ no izvades slāņa līdz ieejas slānim. Tad tiek aprēķināts izmaksu funkcijas gradients katram svaram, ko gradienta lejupslīde izmanto, lai pielāgotu svarus un minimizētu izmaksu funkciju.

Gradienta lejupslīde ir optimizācijas algoritms, kas tiek izmantots, lai izkāptu pa izmaksu funkciju, līdz tiek sasniegts tās minimālais punkts, kas atbilst svaram, kas minimizē neironu tīkla izvades kļūdu. Gradients palīdz noteikt virzienu izmaksu funkcijas pārlūkošanai, savukārt mācību likme palīdz noteikt soli pārlūkošanas laikā. Gan atpakaļizplatīšana, gan gradienta lejupslīde spēlē kritisku lomu neironu tīklu prognožu precizitātes uzlabošanā, minimizējot izvades kļūdu. Kas ir redzams 1.7. attēlī. (Mbali Kalirane “Gradient Descent vs. Backpropagation: What’s the Difference?”, Michael Skirpan “How do Neural Networks Learn?”)



1.7. att. Svaru optimizācija, izmantojot gradienta lejupslīdi. (Mbali Kalirane analyticsvidhya.com)

1.8.2. Konvolūcijas Neironu Tīkls

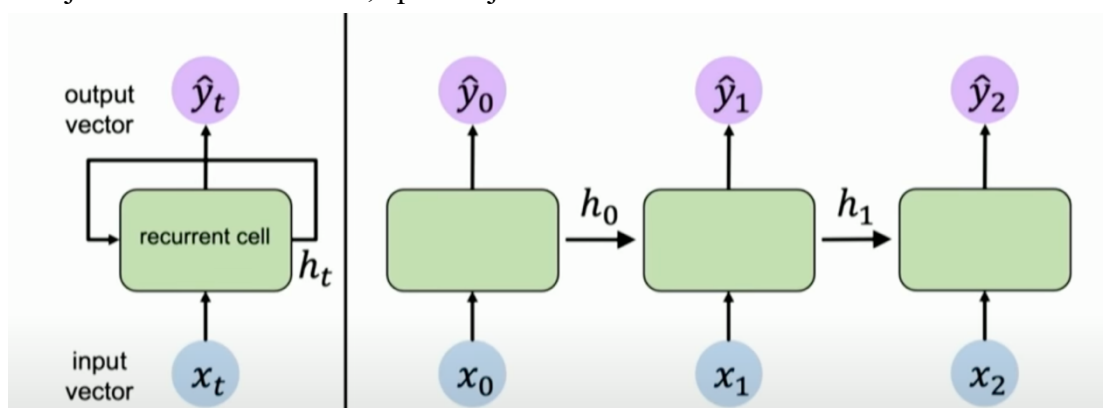
Konvolūcijas neironu tīkli (CNN – *convolution neural network*) ir neironu tīkla veids, kas lielākoties rāda labus rezultātus, apstrādājot attēlu, runu un audio signāla ievades. Tie sastāv no trim galveniem slāņu veidiem: konvolūcijas slānis, izlases slānis un pilnībā savienots slānis. Konvolūcijas slānī notiek lielākā daļa aprēķinu un tiek izmantots filtrs, lai identificētu raksturus attēlā. Izlases slānis samazina parametru skaitu ievadē, savukārt pilnībā savienotais slānis veic klasifikāciju, pamatojoties uz iepriekšējos slāņos izvilktu īpašību. Šādu tīklu autors izvēlējās atstāt nākotnes darbam. Galvenā ideja sastāv no diagrammu svecīšu zīmēšanas un regresijas uzdevuma

veikšanas, pamatojoties uz diagrammu attēlu. Šī metode, apvienojot ar citām NN arhitektūrām vienā lielā ansamblī, var nodrošināt daudz labāku datu generalizāciju un rakstura atpazīšanu diagrammā. CNN lielākais plusiņš ir tas, ka tie izveido īpašības no attēliem, pamatojoties uz gradienta lejupslīdi un kļūdu pēc atpakaļizplatīšanās.

1.8.3. Atkārtotie neironu tīkli

Daudzslāņu perceptroni (MLP – multi layer perceptron) ir uz priekšu vērsti neironu tīkli, kas ir pamatā citiem neironu tīklu modeļiem. Ievades ir piemēri, kas tiek padoti modelim, savukārt izvades ir prognozes. Tomēr MLP ir ierobežojumi. MLP tiek ietekmēta no pazūdošā gradienta problēmas, kas var izraisīt eksploziju vai pazūdjošus gradientus, kas padara tīklam grūti mācīties. Turklāt MLP attiecas pret visu ievadi kā pret objektu kaudzi bez laika secības, un visiem ienākošajiem datiem tiek piemēroti tie paši svāri, kas ir naiva pieeja. Atkārtotie neironu tīkls (RNN – recurrent neural network), arī pazīstams kā dinamisks neironu tīkls, risina dažas no šīm ierobežojumiem, jo tas efektīvi atpazīst laika sērijas uzdevuma laika elementu, ļaujot signāliem plūst gan uz priekšu, gan atpakaļ iteratīvā veidā. Tas arī ļauj tīklam piešķirt īpašus svarus notikumiem, kas notiek sērijās, rezultātā radot dinamisku tīklu. Tomēr RNN joprojām var cīnīties ar pazūdošā gradienta problēmām.

Secības modelēšana, kas ietver neironu tīklus, kas var apstrādāt un mācīties no secīgiem datiem, prasa atšķirīgu neironu tīklu īstenošanas un izveides veidu. Secīgi dati ir klāt daudzos jomās, piemēram, audio, teksts, medicīniskie signāli, finanšu tirgi, bioloģiskās sekvences, klimata modeļi un kustību modeļi. RNN ir izstrādāti, lai apstrādātu secīgas informācijas un uztvertu ciklisku laika atkarību. RNN uztur stāvokli un atjauno to katrā laika solī, apstrādājot virkni. Kas ir redzams 1.8. attēlā.



1.8.att. RNN (Ava Amini

http://introtodeeplearning.com/slides/6S191_MIT_DeepLearning_L2.pdf)

Tātad, vispārējā formula katram prognozējumam katrā laika solī izskatās šādi:

$$y_t = f(x_t + h_{t-1}) \quad (1.2)$$

Kur y_t – izvade laika solī, x_t – ievade konkrētā laika solī, h_{t-1} – iepriekšējais atmiņas stāvoklis.

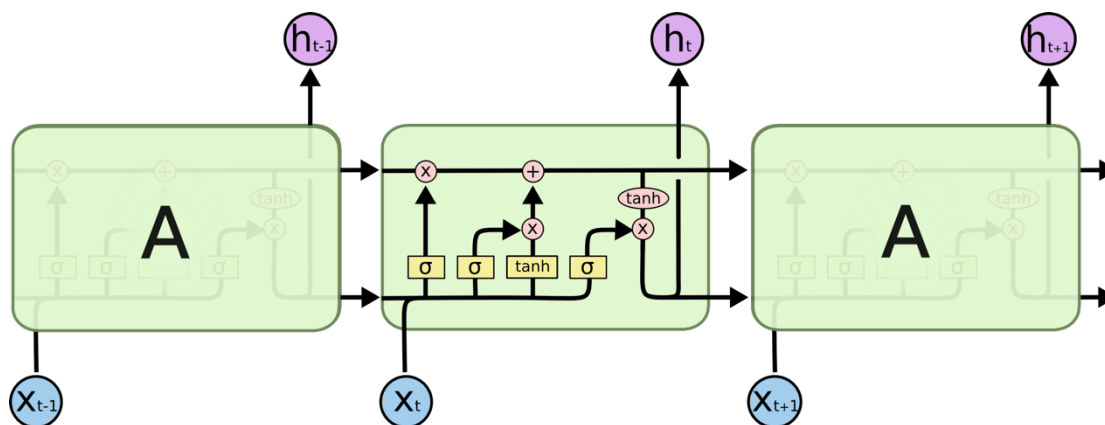
Un formula rekurences attiecībai (slēptajam stāvoklim) laika solī izskatās šādi:

$$h_t = f_W(x_t + h_{t-1}) \quad (1.3)$$

Kur h_t – šūnas stāvoklis, f_W – funkcija ar svariem, h_{t-1} – vecais stāvoklis..

1.8.4. LSTM

Long Short-Term Memory (LSTM) tīkli, kas ir RNN veids, tika ieviesti, lai risinātu problēmu ar ilgtermiņa atkarību. Tos var izmantot, lai mācītos ilgtermiņa atkarības un tie labi darbojas plašā problēmu spektrā. Atšķirībā no standarta RNN, LSTM tīkliem ir četri mijiedarbojoši slāņi atkārtojamajā modulī, nevis viens slānis. Tie ir izstrādāti, lai novērstu ilgtermiņa atkarību problēmu, un šūnas stāvoklis ir to pamatideja. Šūnas stāvoklis, kas ir attēlots ar horizontālu līniju, pāriet caur visu ķēdi, un vārti regulē informācijas pievienošanu vai noņemšanu no tās. LSTM ir trīs vārti, kuri katrs sastāv no sigmoidas neironu tīkla slāņa un punktu reizināšanas darbības. LSTM arhitektūra ir attēlota 1.9. attēlā.



1.9. att. LSTM vienība (Christopher Olah “Understanding LSTM Networks”)

LSTM procesā ir četri soļi. Pirmkārt, aizmiršanas vārtaislemu slānis nosaka, kādu informāciju noņemt no šūnas stāvokļa. Tālāk, ievades vārtaislemu slānis nosaka, kuras vērtības atjaunot, un tanh slānis izveido jaunas kandidāta vērtības. Vecais šūnas stāvoklis tiek atjaunots jaunajā šūnas stāvoklī. Visbeidzot, izvades vārtaislemu slānis

nosaka, kādu izvadi radīt, pamatojoties uz filtrēto šūnas stāvokli. Piemēram, valodas modelī šūnas stāvoklis var ietvert pašreizējā subjekta dzimumu, kas jāaizmirst, ja tiek ieviests jauns subjekts. LSTM tīkls var to efektīvi apstrādāt. (Christopher Olah “Understanding LSTM Networks”)

1.8.5. Uzmanība

Secīgas modelēšanas gadījumā mēs izmantojam pakāpenisku atkārtujošu apstrādi, lai izveidotu spēcīgas arhitektūras, kas apstrādā secīgus datus. Secīgas modelēšanas mērķis ir ņemt ievades datu virkni, izveidot iezīmju kodējumu, izmantojot neironu tīklus, un pārveidot šo iezīmju kodējumu prognozētā izvadā. Atkārtojšie neironu tīkli (RNN) izmanto atkārtotošanos, lai uzturētu secību informāciju un apstrādātu informāciju soli pa solim. Tomēr RNN ir ierobežojumi attiecībā uz kodēšanu, lēnu apstrādi un ierobežotu atmiņu, kas traucē tām sasniegt augstākas spējas modeļu jomā. Lai novērstu šos ierobežojumus, mums ir jāizslēdz nepieciešamība pēc atkārtotās un soli pa solim apstrāde.

Viena naīva pieeja ir visus datus no visiem laika punktiem sasniegt vienā vektora ievadē un padot to uz tiešām propagācijas tīklam (feed-forward), lai aprēķinātu kādu iezīmju vektoru un ģenerētu izvadi. Tomēr šāda pieeja nav mērogojama, jo biežais tiešās propagācijas tīkls būtu jādefinē ar daudzām atšķirīgām savienojumiem. Turklāt, mēs zaudējam secības informāciju, saliekot visu kopā aklā veidā, un mēs tad paliekam ieslodzīti savā spējā izveidot ilgtermiņa atmiņu.

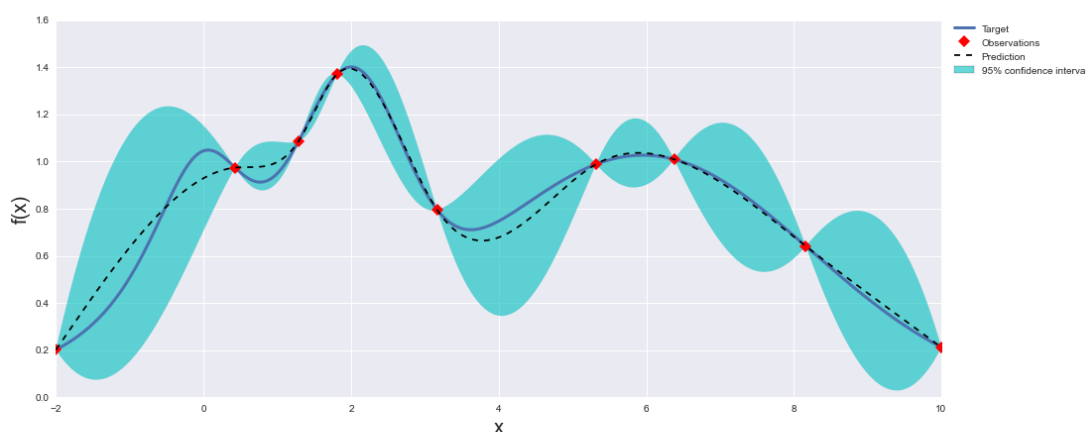
Labāka pieeja ir apvienot laika soļus un izmantot gudrāku veidu, kā mēģināt iegūt informāciju no ievades datiem, izmantojot uzmanību vai pašuzmanību (self-attention). Uzmanība ir spēcīga koncepcija modernā dziļā mācīšanās un AI, un tā ir pamata mehānisms Transformatora arhitektūrā. Uzmanība ļauj mums identificēt un koncentrēties uz svarīgākajiem ievades piemēriem potenciāli secīgās informācijas plūsmā. Būtiska operācija uzmanībā ir koncentrēties uz svarīgākajām ievades daļām. Piemēram, attēlu apstrādē mūsu smadzenes var koncentrēties uz attēla svarīgākajām daļām un izvilkt tās iezīmes, kas pelna vislielāko uzmanību. Pirmā daļa šajā problēmā ir sarežģīta un līdzīga meklēšanas konceptam, kur mēs ņemam lielu informācijas daudzumu un mēģinām identificēt un izvilkt svarīgākās daļas. (Alexander Amini, Ava Amini, “MIT Introduction to Deep Learning”)

Līdzīgi Transformatora arhitektūrai, tehnoloģija, kas izmanto uzmanību, lai izvilkto visvērtīgāko informāciju no ievades datiem un izmantotu šo informāciju, lai uzlabotu prognozēšanas spējas, varētu būt labs turpmākais pētījumu virziens, bet šajā pētījumā tas ir atstāts nākotnes darbam.

1.9. Bajesa optimizācija

Bajesa optimizācijas algoritms tiek izmantots, lai minimizētu skalāru objektīvo funkciju, kas apzīmēta kā $f(x)$, kur x ir robežots domēns, kas sastāv no D komponentēm. x komponentes var būt nepārtrauktas reālskaitļu, veselu skaitļu vai kategoriskas mainīgās vērtības. Objektīvā funkcija var būt noteikta vai stohastiska, kas nozīmē, ka tā var radīt dažādus rezultātus vienādam punktam x . Algoritms izmanto Gausa procesa modeli $f(x)$ un Bajesas atjaunošanas procedūru, lai modificētu Gausa procesa modeli katrā jaunajā $f(x)$ novērtējumā. Algoritms arī izmanto aktivācijas funkciju, kas apzīmēta kā $a(x)$, lai maksimizētu un noteiktu nākamo punktu x novērtēšanai. Gausa procesa modelis tiek iegūts, piemērojot Gausa procesa regresijas modeli objektīvās funkcijas novērojumiem. Aktivācijas funkcija var būt viena no sešām veidu, tostarp cerētā uzlabojuma, uzlabojuma varbūtības un apakšējās uzticamības robežas funkcijām. Algoritms pārtrauc darbību, sasniedzot fiksētu iterāciju skaitu, fiksētu laika ierobežojumu vai pārtraukuma kritēriju. Kopējais process ir attēlots 1.10. attēlā.

Gaussian Process and Utility Function After 9 Steps



1.10. att. Gausa process Bajesas optimizācijā
(<https://github.com/fmfn/BayesianOptimization>)

1.10. Saistītie darbi

Mašīnmācīšanās (ML) ir populāra metode Bitcoin cenas prognozēšanai. Literatūrā ir dažādi ML tehniku piemēri, kas tiek izmantoti, lai prognozētu Bitcoin cenas.

- I. "Predicting the price of Bitcoin using Machine Learning" by Sean McNally.
<https://norma.ncirl.ie/2496/1/seanmcnally.pdf>

Atklājoties McNally et al. (2016) pētījumam, Bitcoin cenas prognozēšana, izmantojot mašīnmācīšanos, ir sarežģīts uzdevums, jo tas ir ļoti mainīgs. Dziļās mācīšanās modeļi, piemēram, RNN un LSTM, ir efektīvi mācītāji, ar LSTM, kas ir spējīgāks atpazīt ilgtermiņa atkarības. Tomēr pārpietiekamība ir galvenais jautājums, kas jālīdzsvaro, lai novērstu pietiekamu mācīšanos no modeļa. Dropout ir noderīga iezīme, kas palīdz šajā procesā, bet tas ne garantē labus validācijas rezultātus. Neskatoties uz labiem snieguma rādītājiem, ARIMA prognoze, pamatojoties uz kļūdu, bija ievērojami sliktāka nekā neironu tīklu modeļi. LSTM modelis nedaudz pārspēja RNN, bet ne nozīmīgi. Tomēr tas prasa ilgāku apmācību laiku. Mašīnmācības algoritmu paralēlizācijas izmantošanas priekšrocības uz GPU ir redzamas, un dati var tikt detalizēti līdz pat minūtēm, lai nodrošinātu vairāk datu punktu. Datu kopu var samazināt, un paralēlizāciju var veikt arī FPGA ierīcēs.

- II. "PRICE PREDICTION OF BITCOIN USING MACHINE LEARNING" by Reshma Sundari Gadey, Nikita Thakur. <https://www.ijeast.com/papers/502-506,Tesma501,IJEAST.pdf>

Mašīnmācības algoritmu, konkrēti dziļo mācīšanās modeļu, piemēram, LSTM, izmantošana ir efektīva Bitcoin cenas prognozēšanā. Tomēr ir svarīgi ņemt vērā pārpildīšanās potenciālu, strādājot ar lieliem datu kopumiem. LSTM arhitektūra ir īpaši efektīva un precīza, bet ir svarīgi ņemt vērā arī citas metodes, piemēram, paralēlizāciju FPGA ierīcēs un datu kopas izmēra samazināšanu, lai uzlabotu prognozes. Kopumā šis pētījums liecina, ka LSTM ir cerīgs pieeja Bitcoin vērtības prognozēšanai nākotnē.

III. Predicting Bitcoin (BTC) Price in the Context of Economic Theories: A Machine Learning Approach by Sahar Erfanian ¹, Yewang Zhou ^{1,*}, Amar Razzaq ^{1,*}, Azhar Abbas ^{2,*}, Asif Ali Safeer ¹ and Teng Li ^{3,4}

Šajā pētījumā ir izmantoti vairāki mašīnmācīšanās modeļi, lai prognozētu Bitcoin cenas, izmantojot dažādus datu kopas. Rezultāti parādīja, ka makroekonomiskie un blokķēdes informācijas rādītāji ir nozīmīgi ilgtermiņa Bitcoin cenas prognozētāji. Pētījums arī atklāja, ka atbalsta vektora regresijas (SVR) modelis izrādījās visefektīvākais starp visiem mašīnmācīšanās modeļiem, un neviens funkciju atlasē tehnika netika pierādīts kā labākais. Šie rezultāti ir svarīgi investoru un aktīvu pārvaldītājiem, lai pieņemtu informētas lēmumus par Bitcoin ieguldījumiem. Tomēr pētījums liecina, ka nākotnes pētījumos var tikt veiktas uzlabojumu modeļu prognozēšanas ziņā, piemēram, izpētot citas funkciju atlasē metodes un salīdzinot citas mašīnmācīšanās modeļus, piemēram, atkārtotu neironu tīklu (RNN) un SVR. Turklāt nākotnes pētījumi var vērsties arī uz citu kriptovalūtu prognozēšanu, tostarp Ethereum un Ripple, un analizēt ziņu ietekmi uz kriptovalūtu tirgu.

Pamatojoties uz šiem pētījumiem, var secināt:

- Dziļās mācīšanās modeļi, piemēram, RNN un LSTM, ir efektīvi mācītāji, ar LSTM, kas spējīgāks atpazīt ilgtermiņa atkarības.
- Bajesas optimizācijas un dropout izmantošana var palīdzēt uzlabot modeļa sniegumu.
- Mašīnmācības algoritmu paralēlizācija uz GPU ir parādījusi snieguma uzlabojumus.
- Literatūras analīze ir parādījusi, ka pamata ekonomikas teorijas, pieprasījuma un piedāvājuma, izmaksu balstītas cenas, makroekonomiskie un blokķēdes informācijas rādītāji ir nozīmīgi Bitcoin cenas prognozētāji.

2. PRAKTISKAIS DARBS

Visas kodu darbības ir veiktas Google Colab piezīmnīcē, kods ir rakstīts Python valodā, izmantojot dažādas bibliotēkas, piemēram, Pandas, Numpy, Matplotlib, Seaborn, Sklearn, Xgboost, Scipy.stats, coinmetric.api_client, Ta, Talib, Tensorflow, Bayes_opt, u.c. Pirmā solis modeļa izveidē ir darba plāna izveide. Šeit ir augsta līmeņa arhitektūra ar soliem, kas jāveic:

- Datu apkopošana
- Datu attīrīšana
- Apmācības/testa datu sadale
- Modeļa apmācība (izmantojot apmācības kopu)
- Mūsu modeļa precizitātes pārbaude (izmantojot testa kopu)
- Modeļa parametru pielāgošana (ieskaitot epohu skaitu, apakšgrupas izmēru, neironu skaitu, optimizācijas funkciju, zaudējumu funkciju utt.), pamatojoties uz veikspēju.
- Modeļa izvietošana, lai prognozētu nākotnes vērtības.
- Modeļa novērtēšana (novērtējam modeļa prognozes rezultātus, aprēķinot atšķirību starp faktiskajām un prognozētajām vērtībām).

2.1. Datu apkopošana

```
df_FA = client.get_market_candles(  
    markets="coinbase-btc-usd-spot",  
    start_time="2016-01-01T04:00:00.000000000Z",  
    end_time="2023-01-21T04:00:00.000000000Z",  
    frequency="1d"  
)
```

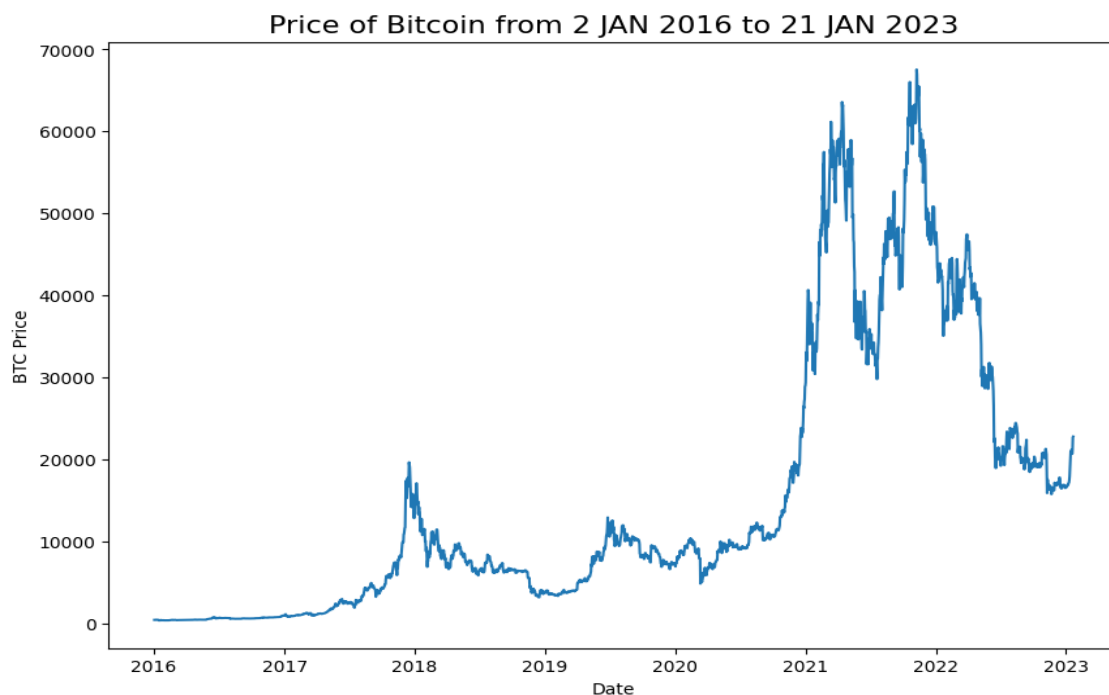
Izmantojot coinmetric.api_client kā klientu un tā metodi client.get_market_candles, autors savāca pirmos Bitcoin datus ar laika zīmēm. Pirmā sākuma datums ir 2016. gada 2. janvāris, un pēdējais dati ir svece ar datuma 2023. gada 21. janvāris.

2.1. tabula

Savāktie Bitcoin datu punkti attiecas uz cenas aizvēšanas vērtību.

Laiks	Price_close
2016-01-02	435.40
2016-01-03	431.91
2016-01-04	433.85
2016-01-05	433.34
2016-01-06	430.87
...	...
2023-01-17	21136.33
2023-01-18	20674.29
2023-01-19	21084.22
2023-01-20	22669.81
2023-01-21	22789.37

BTC price_close vērtība ASV dolāros (\$) ir viens no svarīgākajiem mērķiem. Vispirms mums jāanalizē šī vērtība un jāizveido modelis un pamatlīnija, ņemot vērā tikai to, jo tas ir vienkāršākais uzdevums, ar ko sākt. Cenas dati ir atspoguļoti 2.1. tabulā formātā un 2.1. attēlā vizuālai interpretācijai.

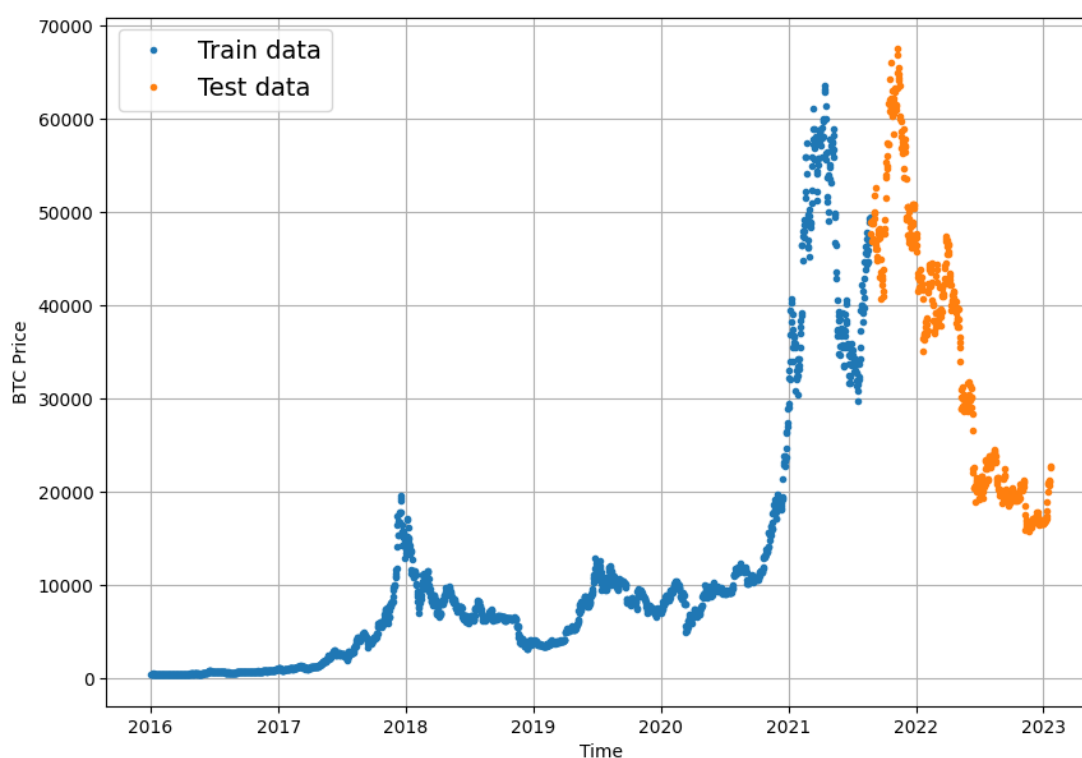


2.1. att. Bitcoin aizvēršanas cena

Nākamais solis būtu datu attīrīšana, bet šeit mums ir tikai viena mērķa vērtība, tāpēc nav nepieciešams attīrīt datus.

2.2. Train / Test sadalījums

Apmācības un testa datu sadalījums tiek veikts pēc 80% / 20% principa, kas nozīmē, ka 20% no visiem datiem tiks izmantoti kā testa dati modelim, un modelis netiks apmācīts ar šiem datiem. Testa datu kopā ir 515 datu elementi, un kopējais datu kopas apjoms ir 2577 datu elementi, sākotnējais datu punkts ir 2021. gada 25. jūlijs. Apmācības un testa datu sadalījuma vizuālais attēlojums ir redzams 2.2. attēlā.



2.2. att. Train / Test sadalījums of BTC cenas

2.3. Pamata līnija un metrikas

Pirms apmācības modeli ir labāk izveidot bāzes līniju, lai redzētu, cik labi mūsu modelis prognozē cenas salīdzinājumā ar bāzes līniju. Ja šis būtu klasifikācijas uzdevums, vienkārša bāzes līnija būtu prognozes precizitāte jābūt augstākai par nejaušu minēšanu, un, starp citu, pareizi prognozēt tikai bināro klasifikācijas uzdevumu (pirkt

vai pārdot konkrētā sveces laikā) ar stabilu precizitāti, kas ir vairāk nekā 51%, ir faktiski lielisks rezultāts.

Autore izvēlējās risināt regresijas problēmu, nevis klasifikāciju, un tāpēc ir specifiskas metrikas, kas var parādīt, cik labi modelis darbojas ar testa datiem. Šeit ir dažādas metrikas, ko mēs izmantosim, lai novērtētu prognozes kvalitāti.

MAE: vidējā absolūtā kļūda. Viegli interpretējama, tāpēc prognoze ir X summas absolūta starpība starp prognozi un patieso cenu konkrētajā laikā, kas ir X summa dolāru (\$). +100 vai -100 ir vienādas 100 dolāru kļūdas.

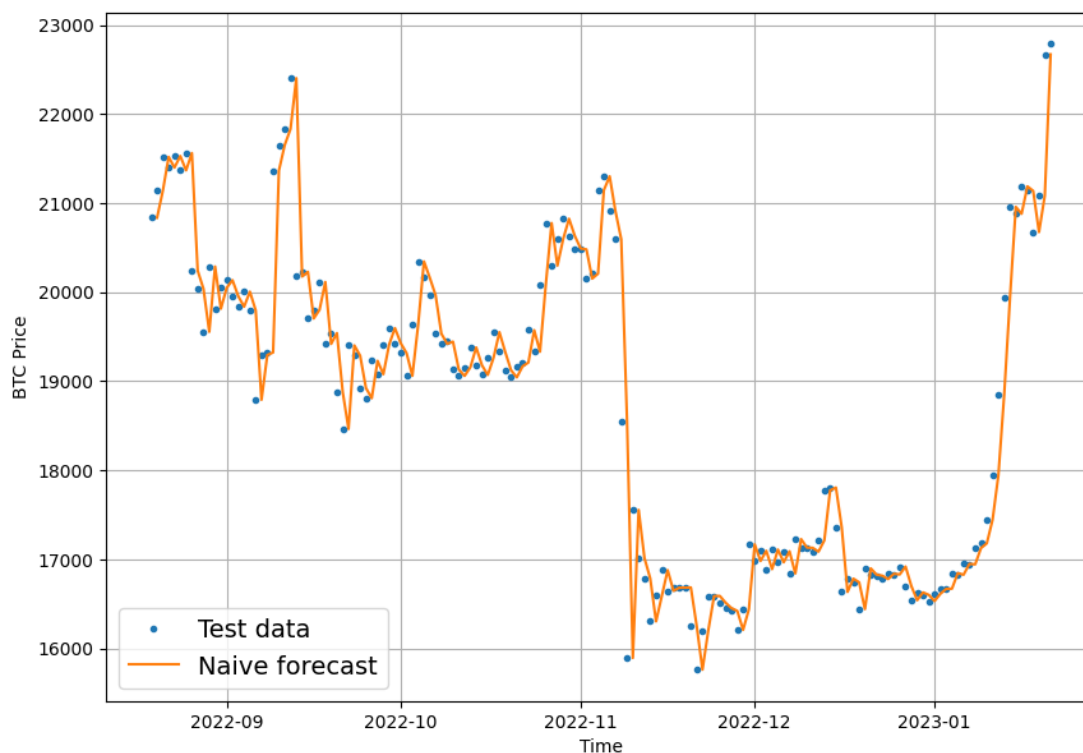
RMSE: vidējā kvadrāta saknes kļūda. Minimizējot RMSE, tiek prognozēta vidējā vērtība.

MAPE: vidējā absolūtā procentuālā kļūda. Visbiežāk lietotā procentuālā kļūda. Tā vietā, lai redzētu tikai starpību starp prognozi un patieso vērtību, šeit redzam atšķirību salīdzinājumā ar visu cenu diapazonu.

sMAPE: šī metrika ir vajadzīga tikai salīdzināšanai ar bāzes līniju.

Kā bāzes līniju tika izvēlēta Naiva prognoze.

2.4. Naiva prognoze



2.3. att. Naiva prognoze

Rakstā "Prognozēšana enerģijas tirgos: cik laba ir naivā prognoze?" no Gavin Bell ir norādīts, ka naivā prognozes metode, kurā nākotnes vērtības tiek uzskatītas par vienādām ar pašreizējo vērtību, bieži tiek kritizēta par tās vienkāršību un ierobežoto precizitāti daudzos pielietojumos. Tomēr dažos tirgos, piemēram, enerģijas tirgus cenās, naivā prognoze sniedz pārsteidzoši labus rezultātus un pat var pārspēt sarežģītākus prognozēšanas metodēs. Naivās prognozes vizuālie rezultāti redzami 2.3. attēlā. Šī panākumu iemesls slēpjas pašos tirgos, kur pašreizējā cena atspoguļo visu pieejamo informāciju tirgum, padarot naivo prognozi par labu īstermiņa prognozētāju. Neskatoties uz šiem panākumiem, rakstā ir arī norādīts, ka daudzas mašīnmācīšanās laika sēriju prognozēšanas modeļi bieži vien beigās mācās būtiski vienkāršu autoregresīvu modeļi, nesniedzot daudz vairāk par naivo prognozi. Tāpēc naivā prognoze nav panaceja un tai ir savas ierobežojums, īpaši ilgtermiņa prognozēs un piemēros, kur var izmantot vairāk informācijas, lai uzlabotu precizitāti. Šeit ir precizitātes metrikas naivai prognozei:

a. 'mae': 833.2916,

b. 'mse': 1602351.6,

c. 'rmse': 1265.8403,

d. 'mape': 2.366376

Vidējā absolūtā kļūda testa datu kopā ir 833 USD, kas ir 2.36% no cenu diapazona. Vidējā BTC cena testa datu kopā ir 34094 USD, un prognozes kļūda 833 USD nav tik slikti.

2.5. Datu izpratne

Lai pārbaudītu, vai autors ir izvēlējis piemērotu pamatliniju, salīdzināsim to ar ARIMA modeli. Attēlā 2.4. ir redzami ARIMA rezultāti testa kopā.

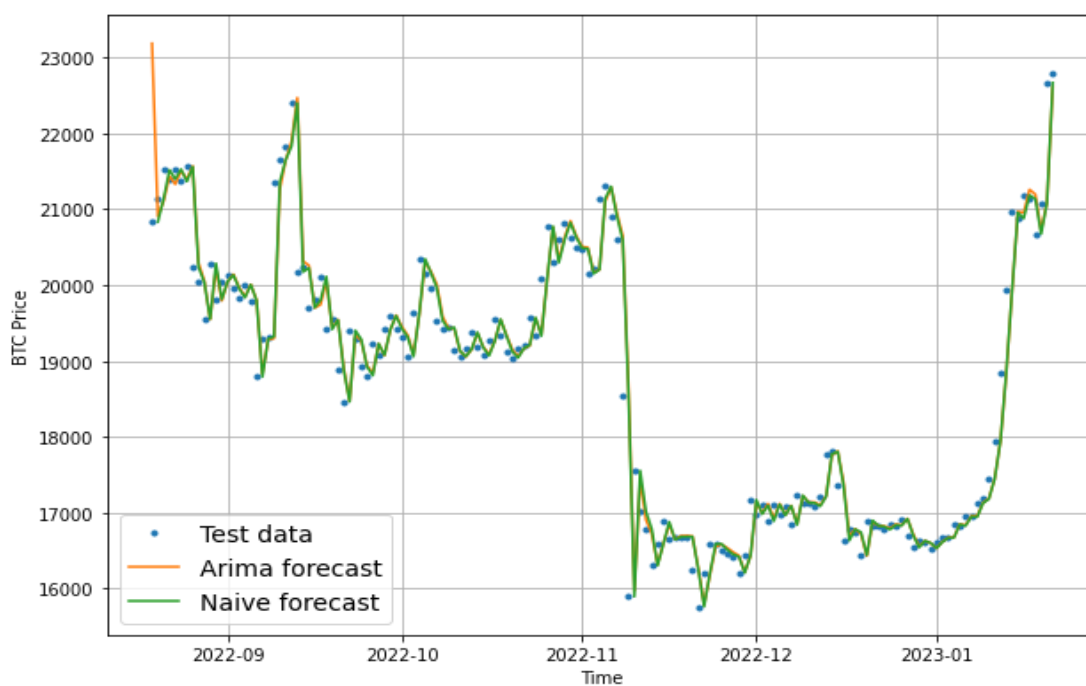


2.4. att. ARIMA forecast on test set of BTC price

Tagad salīdzināsim to ar mūsu pamatliniju Naive Forecast:

- a. 'mae': 842.5529,
- b. 'mse': 1633538.9,
- c. 'rmse': 1278.0997,
- d. 'mape': 2.3919811

Mums ir nedaudz sliktāki rezultāti salīdzinājumā ar pamatliniju, pat ar vienu no labākajām mašīnmācīšanās prognozēšanas metodēm - ARIMA. Attēlā 2.5. redzam, ka starpība ir ļoti maza.



2.5. att. ARIMA un Naive rezultāti pēdējās 360 testa laika posmos.

Autokorelācijas plotā var redzēt savstarpējo saistību laika sērijas un tās atliku starpā. Tas palīdz atrast paraugus un atkarības datus laika sērijas analīzei, prognozēšanai un modeļu izvēlei.

No autokorelācijas plotas var secināt:

Stacionaritāte: Ja autokorelācijas vērtības ātri nokrītas līdz nullei vai zema vērtībai, laika sērija var būt stacionāra. Stacionārai laika sērijai ir konstanta vidējā un dispersija laika gaitā. Pretējā gadījumā nestacionāra laika sērija rāda ievērojamu autokorelāciju, kas saglabājas vairākās atlikās, kas norāda, ka vidējā vai dispersija var mainīties laika gaitā.

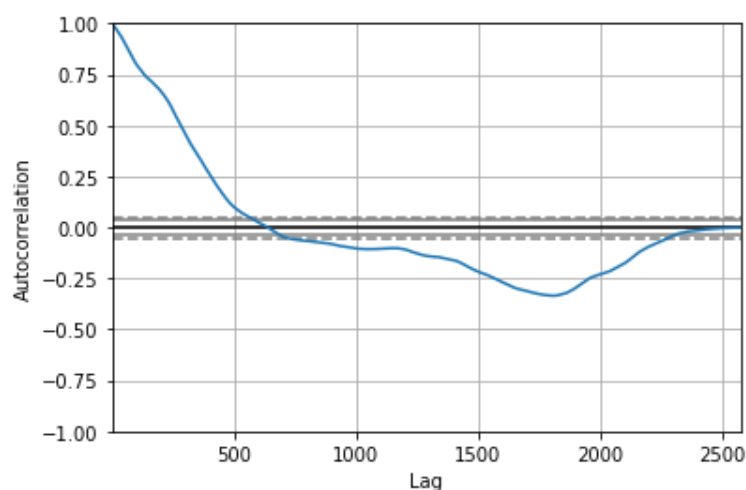
Sezonālums: Regulāri atstatņi un nozīmīgi maksimumi plotā norāda uz sezonālumu datu. Atstatņu attālums atbilst sezonas periodam (piemēram, 12 mēnešiem mēneša datiem ar gadu sezonālumu). Sezonālie raksti var tikt izmantoti prognozēšanai un datu koriģēšanai sezonāluma gadījumā, veidojot modeļus.

Autoregresīvās (AR) vai kustības vidējās (MA) komponentes: Ja plotā redzams pakāpenisks autokorelācijas vērtību samazinājums, tas var norādīt uz autoregresīvu (AR) procesu. Ja plotā pēc dažām atlikām seko strauja kritiena un autokorelācijas vērtības tuvu nullei, tas var norādīt uz kustības vidējo (MA) procesu. Šie atziņas var palīdzēt izvēlēties piemērotus modeļus laika sērijas prognozēšanai, piemēram, ARIMA (autoregresīvs integrēts kustības vidējais).

Daļējā autokorelācija: Papildus autokorelācijas plotam ir vērts izpētīt daļējo autokorelācijas plotu (PACF), kas mēra tiešo attiecību starp diviem novērojumiem, kas atšķirās pēc noteiktas atlikas, izņemot starpnieku novērojumu ietekmi. PACF var sniegt papildu ieskatu AR un MA komponentu kārtībā.

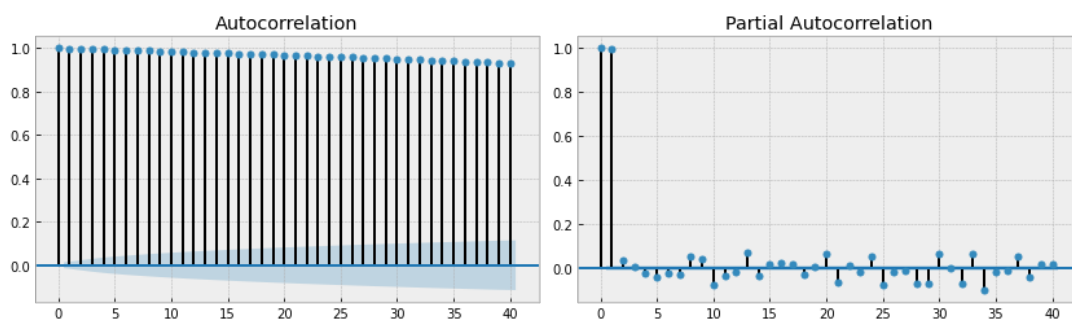
Baltais troksnis: Ja autokorelācijas vērtības nav nozīmīgas (t.i., atrodas ticamības intervālos) visām atlikām, laika sēriju var uzskatīt par baltu troksni. Baltajam troksnim nav paredzama rakstura un to nevar prognozēt, izmantojot tradicionālos laika sēriju modeļus.

Attēlā 2.6. ir redzams BTC cenas autokorelācijas plots.



2.6. att. BTC cenas autokorelācijas plots

No šī plota var secināt, ka BTC cenas nav stacionāras, nav sezonāluma, nav AR vai MA komponentes, dati nav baltā trokšņa visā datu kopā. Tagad attēlā 2.7. ir redzams daļējais autokorelācijas plots.



2.7. BTC cenas daļējais un auto korelācijas plots

Tas ir papildu analīzes rīks, kas palīdz dziļāk izprast BTC cenas mērķa vērtību. Paša paša autokorelācijas daļā plota 2.7. attēlā pirmās divas atlikas rāda nozīmīgi augstākas vērtības salīdzinājumā ar visiem citiem, kas parasti nozīmē, ka visā datu kopā ir spēcīgs augšupejošais trends, kas arī ir apstiprināts ar kreiso daļu plota 2.7. attēlā.

2.6. GPU

Pirmkārt, pamatojoties uz Sean McNally veikto pētījumu, GPU izmantošana modelēšanas apmācībai var ievērojami palielināt apmācības ātrumu, tāpēc autors izmantoja bezmaksas cloud GPU no Google Colab piezīmjdatora. Attēlā 2.8. ir redzami GPU parametri, kas tika izmantoti šajā pētījumā.

```
! nvidia-smi

Thu Apr 27 18:05:24 2023

+-----+
| NVIDIA-SMI 525.85.12      Driver Version: 525.85.12      CUDA Version: 12.0      |
+-----+-----+
| GPU  Name           Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf    Pwr:Usage/Cap |      Memory-Usage | GPU-Util  Compute M. |
|====+=====+====+=====+=====+=====+=====+=====+
|  0   Tesla T4              Off   | 00000000:00:04.0 Off |             0         |
|N/A   56C    P8      10W /  70W |  0MiB / 15360MiB |      0%      Default |
+-----+-----+-----+-----+-----+-----+-----+

+-----+
| Processes: |
| GPU   GI    CI          PID    Type    Process name                  GPU Memory |
|  ID   ID     ID              |                 |           Usage         |
+-----+-----+-----+-----+-----+-----+
| No running processes found |
+-----+
```

2.8. Colab bezmaksas GPU parametri

2.7. Logu izveide

Lai apmācītu dziļo mācīšanās neironu tīklu, mums ir jāpārstrukturē BTC cenas laika sēriju dati par uzraudzības mācības problēmu. To paveic, izveidojot datu logus un horizontus. Laika sēriju analīzes ietvaros logu veidošana ir datu priekšapstrādes tehnika, kas ietver laika sērijas sadalīšanu pārklājošās vai nesalauztās apakšvirknēs, ko sauc par logiem. Katrs logs satur noteiktu laika solu skaitu, un uzdevums ir paredzēt nākamo laika soli (vai vairākus laika soļus) balstoties uz datiem logā. Šāda pieeja pārvērš laika sēriju prognozēšanas problēmu par uzraudzības mācības problēmu, kur katrs logs tiek uzskatīts par atsevišķu apmācības piemēru, un mērķa mainīgais ir nākamais laika solis laika sērijā.

Logu veidošana ir labs risinājums, jo tas ļauj izmantot mašīnmācības algoritmus, piemēram, neironu tīklus, kas prasa apzīmētus apmācības piemērus, lai veiktu prognozes laika sērijas datiem. Izveidojot laika soļu logus, mēs varam izveidot apzīmētus apmācības piemērus no neapzīmētiem laika sērijas datiem, kas ļauj mums izmantot šos algoritmus prognozēšanai.

Logu veidošana arī ļauj mums iegūt laika atkarības, kas ir svarīgas precīzai prognozēšanai. Izmantojot pārklājošus logus, mēs varam nodrošināt, ka katrs laika solis

ir iekļauts vairākos logos, kas ļauj mums iegūt gan īstermiņa, gan ilgtermiņa atkarības laika sērijas datus.

[0, 1, 2, 3, 4, 5, 6] -> [7]

[1, 2, 3, 4, 5, 6, 7] -> [8]

[2, 3, 4, 5, 6, 7, 8] -> [9]

BTC:

[435.4 431.91 433.85 433.34 430.87 459.07 454.44] -> [450.38]

[431.91 433.85 433.34 430.87 459.07 454.44 450.38] -> [449.99]

[433.85 433.34 430.87 459.07 454.44 450.38 449.99] -> [449.19]

2.8. Modela apmācība

Pirmā neironu tīkla arhitektūra ir parasta secīga modelis no TensorFlow, un tā sastāv no 1 slēptā slāņa ar 128 neironiem. Optimizators ir `tf.keras.optimizers.Adam()`, jo daudzi pētījumi to ieteic kā vienu no labākajiem hiperparametriem šim uzdevumam. Zaudējumu funkcija ir 'MAE' (vidējā absolūtā kļūda). Modelis tiek apmācīts 500 epohās, un labākās svaru vērtības tiek atgrieztas pēc 500 epohām. Horizons ir 1, un Logs izmērs ir 7. Apakšgrupas izmērs ir 128.

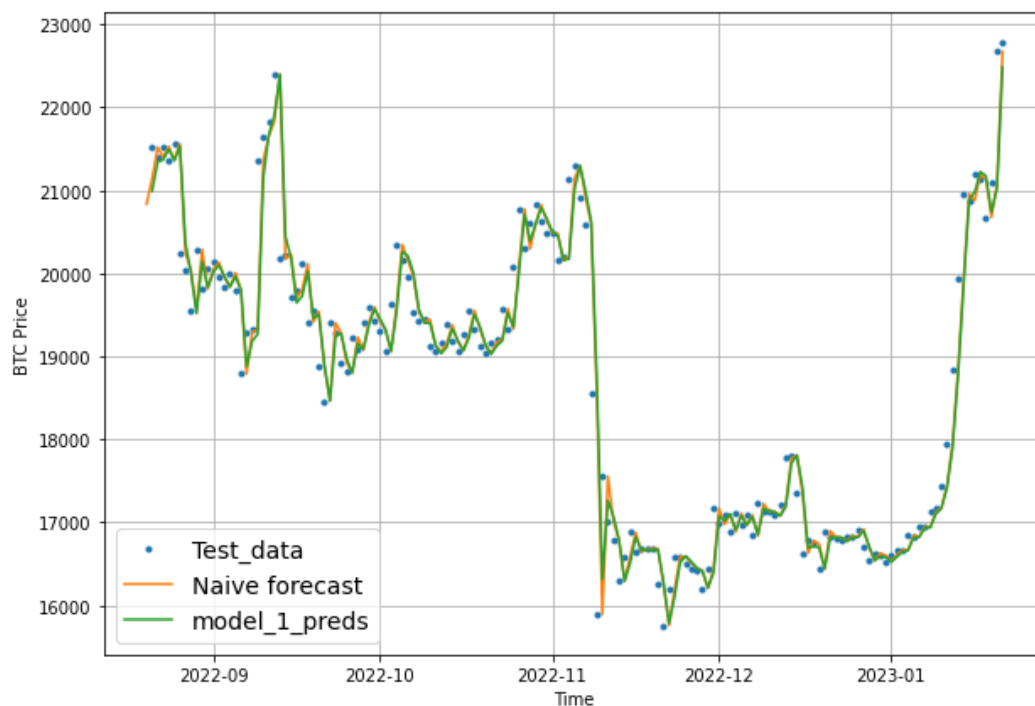
a. 'mae': 838.01746,

b. 'mse': 1633832.2,

c. 'rmse': 1278.2145,

d. 'mape': 2.3890312

Tātad rezultāti ir nedaudz sliktāki nekā pamatlinija.



2.9. att. Modeļa rezultāti salīdzinājumā ar Naive prognozi pēdējās 360 laika soļos testa datu kopā

2.9. Dažādas modeļu arhitektūras

Tagad ir laiks izveidot dažādas modeļu arhitektūras. Šeit ir modeļi ar 2 slēptajiem slāņiem un dažādu neironu izmēru katrā slānī. Modeļi izmanto dažādas aktivizācijas funkcijas, piemēram, ReLU, ELU, SELU, GELU. Modeļi ar izslēgšanas slāņiem, lai novērstu pārmācību datiem, bet tie parāda sliktākus rezultātus. Dati ir attēloti 2.10. attēlā. Modeļi ar 2 slēptajiem slāņiem veic kopumā labāk nekā modeļi ar tikai vienu slēpto slāni, un ReLU aktivizācijas funkcija arī parāda labākos rezultātus.

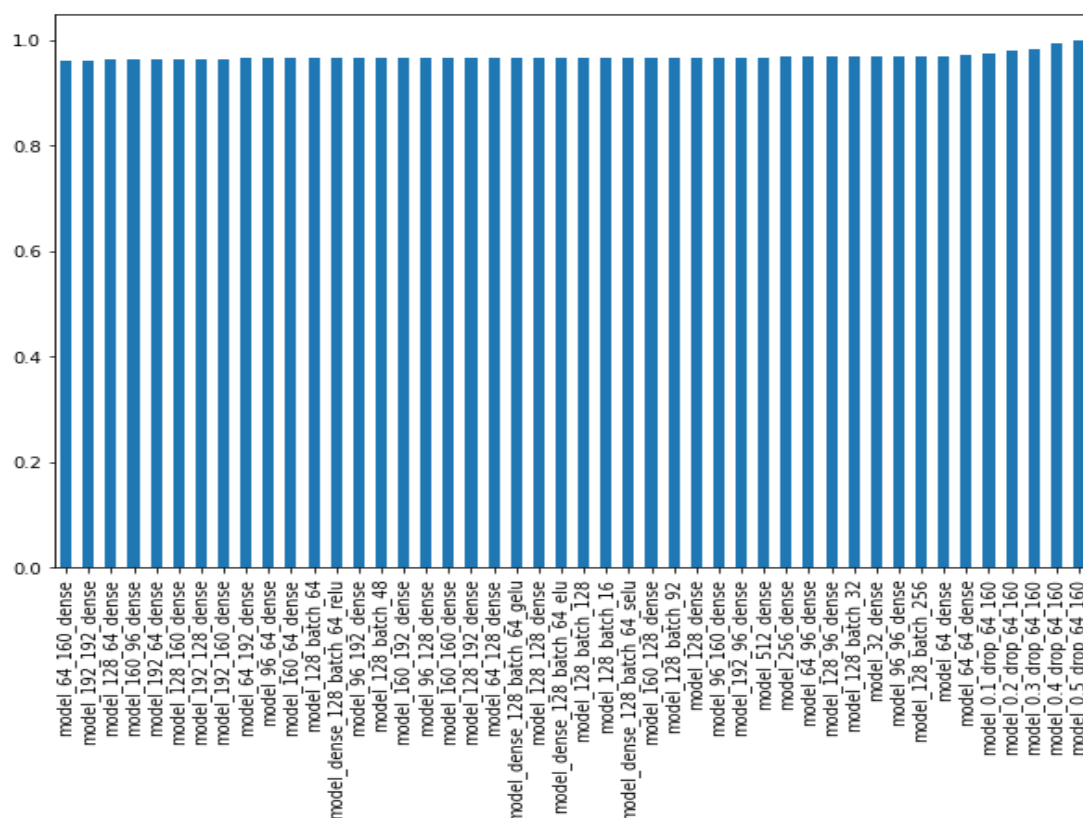
`dense_results['model_64_160_dense']` - tas ir labākais modelis, kas sasniedzusi nedaudz labākus rezultātus nekā pamatlinija, bet ar 831 USD MAE joprojām nav pietiekami labi, lai regulāri gūtu peļņu no tirdzniecības.

a. 'mae': 831.89606,

b. 'mse': 1612133.2,

c. 'rmse': 1269.6981,

d. 'mape': 2.36875



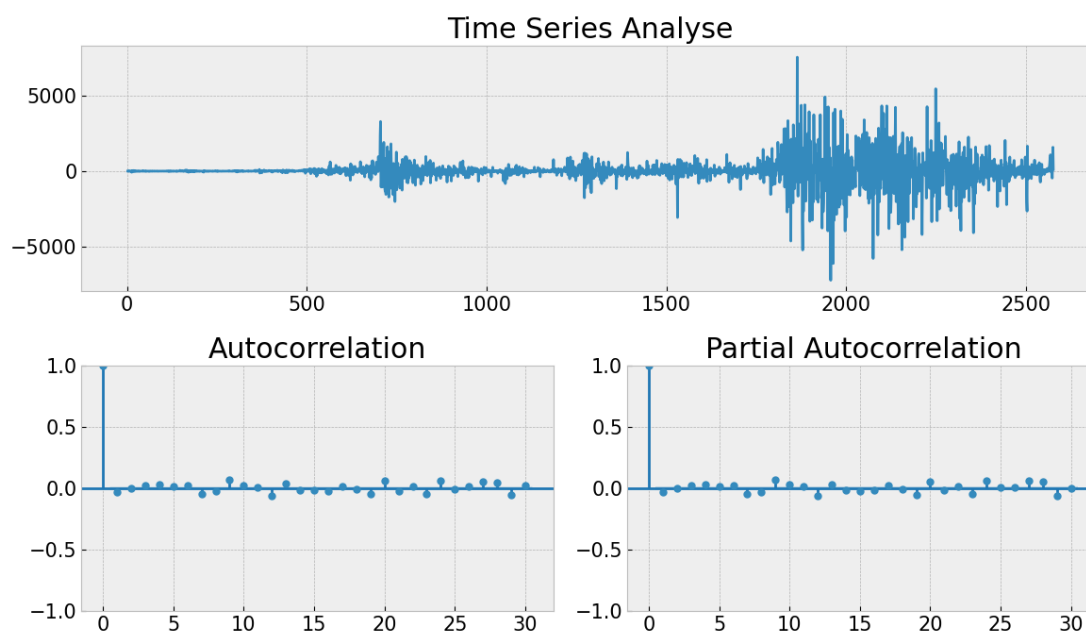
2.10. att. Dažādas modeļu arhitektūras. Mērogojums un sakārtojums, pamatojoties uz MAE.

Konvolūcijas un LSTM modeļi parāda 835 un 836 MAE, tādēļ tie nevar palīdzēt šīs problēmas atrisināšanā.

2.10. Stacionaritātes analīze

Lai pārbaudītu parastas cenas datu stacionaritāti, tiek izmantots **Dickey-Fuller** tests: parāda **p=0,436724**, kas liecina, ka BTC cenas nav stacionāras, kas arī ir sagaidāms.

Vienkāršākais veids, kā pārveidot BTC cenas par stacionāriem datiem, ir izmantojot atšķirību starp cenu un iepriekšējo cenu. 2.11. attēlā redzama atšķirība starp cenu un nobīdi cenu par 1 dienu.



2.11. att. Cena slēgšana - Cena slēgšana nobīdīta par 1 dienu

Dickey-Fuller tests: $p=0,000000$, kas liecina, ka statistiski ar 95% ticamības līmeni šī laika sēriju dati tagad ir stacionāri, kas var redzēt arī attēlā.

Izmantojot ARIMA, lai prognozētu stacionāro cenu, rezultāti ir ļoti slikti:

- a. 'mae': 1539.5793,
- b. 'mse': 4245592.0,
- c. 'rmse': 2060.4834,
- d. 'mape': 4.6764655

Divas reizes sliktāki nekā pamatlinija. Bet LSTM modelis parādīja labākus rezultātus:

- a. 'mae': 836.5677,
- b. 'mse': 1603176.5,
- c. 'rmse': 836.5677,
- d. 'mape': 2.3254757

2.11. Daudzdimensionāla laika sēriju prognoze

Daudzdimensionālās laika sēriju prognozēšanas metode ļauj pievienot vairāk funkciju datu bāzē un analizēt tās kopā, lai veiktu labākas prognozes. Šim uzdevumam tika izvēlētas dažādas datu avotu, piemēram, **Tehniskā analīze, Fundamentālā analīze, Furjē pārveides, Modelēšanas datu atpazīšana, Laika funkcijas.**

2.11.1. Tehniskā analīze

Šeit ir kolonnas, kas tika pievienotas datiem no TA.:

['time', 'price_open', 'price_close', 'price_high', 'price_low', 'vwap', 'volume', 'ma7', 'ma21', 'ma49', 'ma200', 'EMA_26', 'EMA_12', 'ema', 'RSI', 'AD', 'ATR', 'STOCH', 'death_cross', 'life_cross', 'global_trend', 'tsi', 'fii', 'mfi', 'adi', 'cmf', 'macd_diff', 'macd', 'macd_signal', 'bb_bbm', 'bb_bbh', 'bb_bbl', 'bb_bbhi', 'bb_bbli', 'adx', 'adx_neg', 'adx_pos', 'cci', 'ulti_osci', 'roc', 'STC']

2.11.2. Fundamentālā analīze

['time', 'price_open', 'price_close', 'price_high', 'price_low', 'volume', 'candle_usd_volume', 'candle_trades_count', 'mrctCap', 'TxCntSec', 'TxCnt', 'AdrActCnt', 'newMerchantsCnt', 'googleTrends', 'gdp%', 's_p_500', 'gold', 'crude', 'gas', 'block_reward', 'DiffLast', 'HashRate', 'RevHashRateNtv', 'RevHashUSD', 'RevUSD', 'FlowTfrFromExCnt', 'TxTfrValAdjUSD', 'SplyFF']

2.11.3. Veidu atpazīšana

['price_open', 'price_close', 'price_high', 'price_low', 'CDL2CROWS', 'CDL3INSIDE', 'CDL3LINESTRIKE', 'CDL3OUTSIDE', 'CDL3WHITE SOLDIERS', 'CDLADVANCEBLOCK', 'CDLBELTHOLD', 'CDLCLOSINGMARUBOZU', 'CDLDOJI', 'CDLDOJISTAR', 'CDLDRAGONFLYDOJI', 'CDLENGULFING', 'CDLEVENINGDOJISTAR', 'CDLEVENINGSTAR', 'CDLGAPSIDESIDEWHITE', 'CDLGRAVESTONEDOJI', 'CDLHAMMER', 'CDLHANGINGMAN', 'CDLHARAMI', 'CDLHARAMICROSS', 'CDLHIGHWAVE', 'CDLHIKKAKE', 'CDLHIKKAKEMOD', 'CDLINVERTEDHAMMER', 'CDLLONGLEGGEDDOJI', 'CDLLONGLINE', 'CDLMARUBOZU', 'CDLMATCHINGLOW', 'CDLMORNINGDOJISTAR', 'CDLMORNINGSTAR', 'CDLRICKSHAWMAN', 'CDLRISEFALL3METHODS', 'CDLSEPARATINGLINES', 'CDLSHOOTINGSTAR', 'CDLSHORTLINE', 'CDLSPINNINGTOP', 'CDLSTALLEDPATTERN', 'CDLTAKURI', 'CDLXSIDEGAP3METHODS', 'count_1', 'count_-1', 'pr_signal', 'time']

2.11.4. Laika vērtības

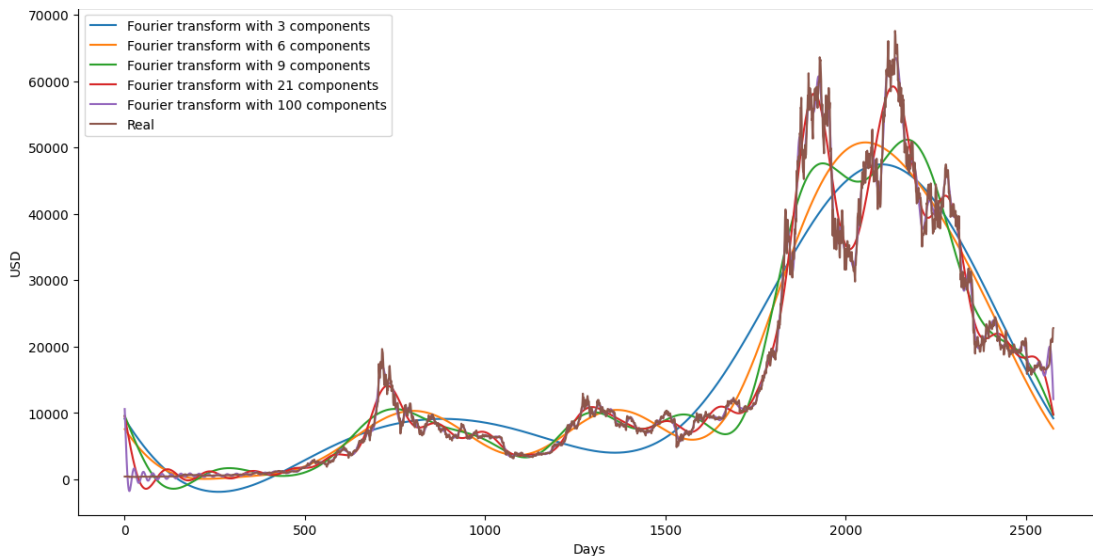
['time', 'price_close', 'price_open', 'price_high', 'price_low', 'seconds', 'day_of_week', 'month', 'year', 'quarter', 'close_lag_7d', 'close_lag_30d',

*'day_of_week_sin', 'day_of_week_cos', 'month_sin','month_cos', 'price_sin',
'price_cos', 'price_7_sin', 'price_7_cos','price_30_sin', 'price_30_cos']*

2.11.5. Fourjē transformācijas

Fourjē transformācija ir matemātiska tehnika, kas pārveido jebkādu laika funkciju par frekvenču funkciju. Fourjē transformācija ir cieši saistīta ar Fourjē sēriju. Tā pieņem jebkādu laika pamatotu raksturu kā ievadi un aprēķina vispārējo cikla nobīdi, rotācijas ātrumu un stiprumu visiem iespējamiem cikliem.

Fourjē transformācija vislabāk piemērota viļņveidīgām funkcijām, jo tās ir atkarīgas no laika un telpas funkcijām. Kad Fourjē transformācija tiek piemērota viļņveidīgai funkcijai, tā tiek izdalīta sinusoidālu komponentu virknē.



2.12. att. Fourjē transformācijas (3, 6, 9, 21, 100) komponentes

Fourjē transformācijas vizualizācija ir redzama 2.12. attēlā. Un datu instances.
*['time', 'price_close', 'price_open', 'price_high', 'price_low','fourier_3_real',
'fourier_3_imag', 'fourier_6_real', 'fourier_6_imag','fourier_9_real', 'fourier_9_imag',
'fourier_21_real','fourier_21_imag', 'fourier_100_real', 'fourier_100_imag']*

2.12. Apzīmējumi: apmācība pirmie modeļi ar daudzmainīgu laika intervālu, izmantojot dažādas aktivācijas funkcijas

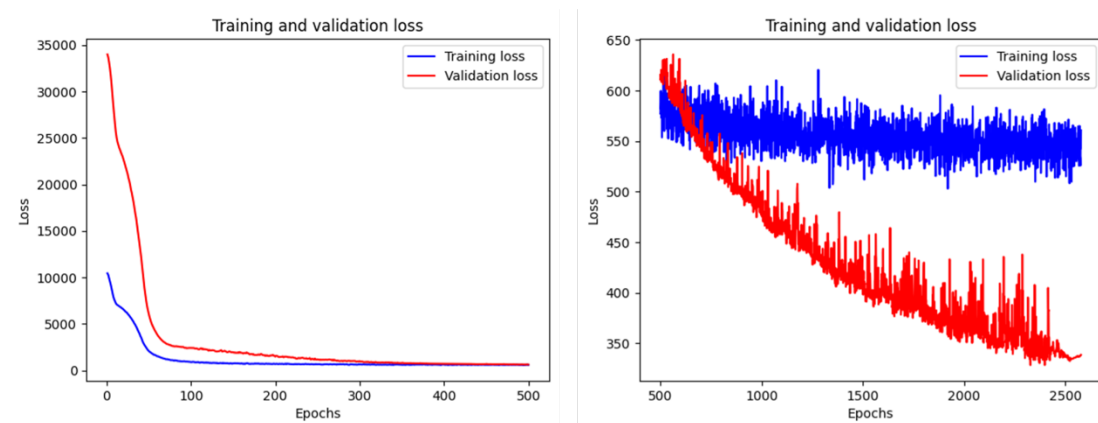
Viena no neironu tīkla hiperparametriem ir aktivācijas funkcija, kas nodrošina ne-līnijas iedarbību. Vispopulārākā ir ReLU aktivācijas funkcija, ko raksturo vienkāršība un labi rezultāti. Konkrētajai analīzei tika izmantota vienāda neironu tīkla ar 4 dažādām aktivācijas funkcijām: ReLU, GELU, ELU, Tanh (1.5. attēlā).

2.12.1. Relu

- 512 neironi 1 slēptajā slānī
- Izslēgšanas iespēja - 0,5
- L2 regularizācija - 0,0001

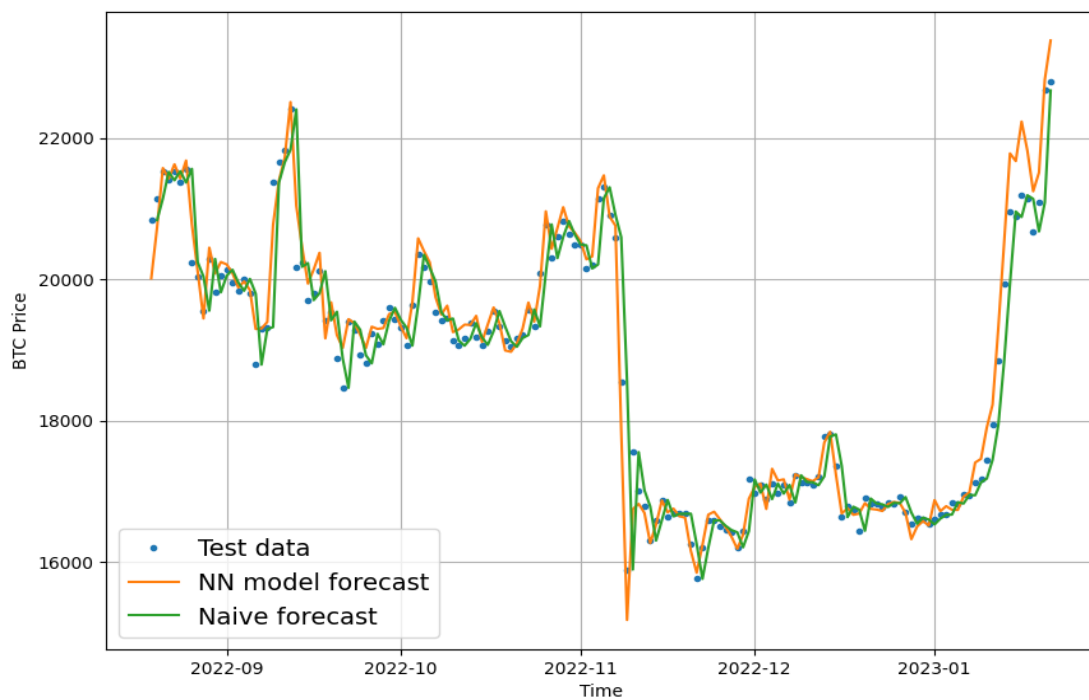
MAE kļūda pēc 2570 epohām ir 317,67783

Pavadītais laiks: 15 minūtes 7 sekundes



2.13. att. Apzīmējumi apmācības un validācijas kļūdām ReLU modelī

Pēdējās epohās validācijas kļūdas svārstības ir ievērojami mazākas nekā iepriekš, jo mācīšanās ātrums tika automātiski pielāgots, izmantojot `tf.keras.callbacks.ReduceLROnPlateau` pēc 180 epohām bez uzlabojumiem validācijas kļūdā. Apmācība tiek pārtraukta, ja pēdējās 250 epohās nav sasniegts nekāds progress validācijas kļūdā.

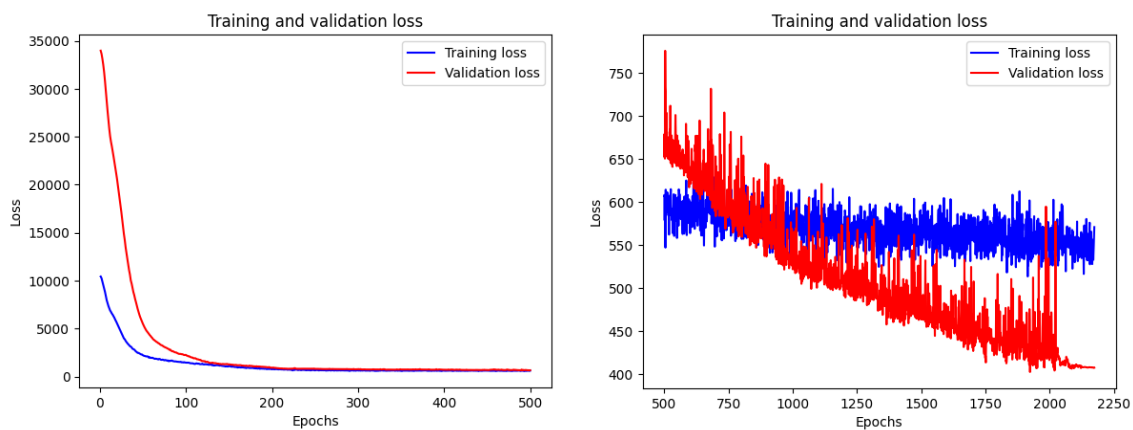


2.14. att. Modeļa rezultāti testa kopā.

2.12.2. GELU

MAE kļūda pēc 2174 epohām ir 390,407

Pavadītais laiks: 14 minūtes 14 sekundes

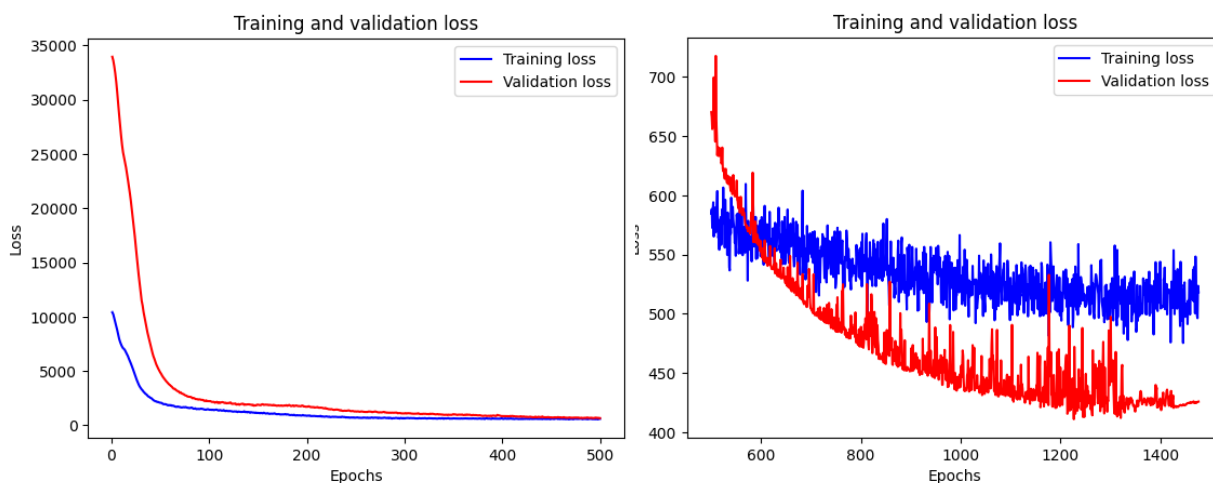


2.15. att. Apzīmējumi apmācības un validācijas kļūdām GELU modeli

2.12.3. ELU

MAE kļūda pēc 1478 epohām ir 398,58

Pavadītais laiks: 8 minūtes 46 sekundes

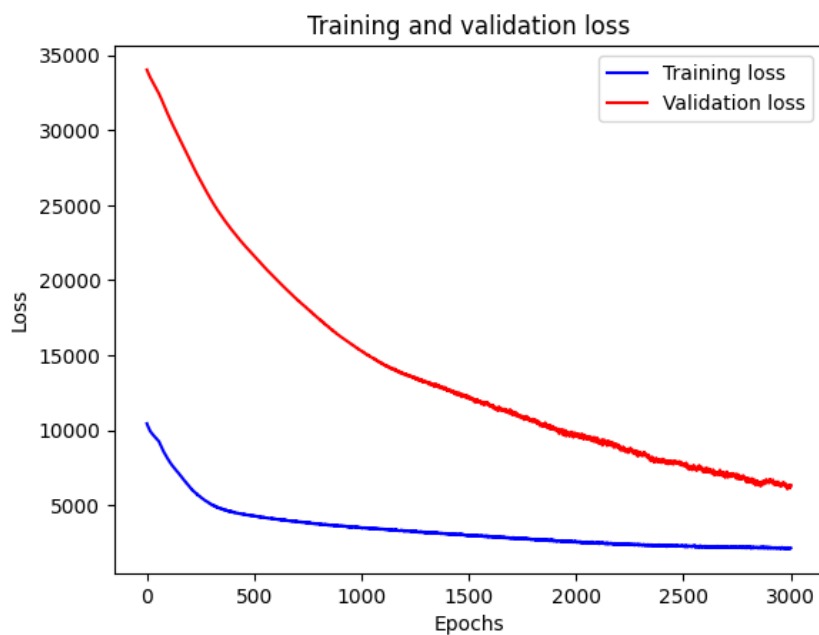


2.16. att. Apzīmējumi apmācības un validācijas kļūdām ELU modelī

2.12.4. Tanh

MAE kļūda pēc 3801 epohām ir 5150,377

Pavadītais laiks: 23 minūtes 38 sekundes



2.15. att. Apzīmējumi apmācības un validācijas kļūdām Tanh modelī

2.13. CNN modelis

Pēc 1331 epohām modelis ar konvolūcijas slāni pēc ieejas slāņa un tādi paši 512 neironi parāda labus rezultātus:

- a. 'mae': 309,73956,
- b. 'mse': 177999,72,
- c. 'rmse': 421,90015,
- d. 'mape': 0,98211205

2.14. Bajesa optimizācija

2.2. tabula

Bajesa optimizācija tehnikas analīzei, izmantojot vienu mērķi "Price Close" ar vienu slēpto slāni.

Iteration	MAE	Reg	Neironu sk.	Dropout
1	425.58072	0.2622	977.4	0.732
2	717.3306	0.4191	161.2	0.156
3	293.99414	0.04066	890.6	0.6011
4	3333.601	0.4957	22.14	0.9699
5	1047.1382	0.5827	219.1	0.1818
6	388.72565	0.1998	876.1	0.5354
7	282.3957	0.02454	888.9	0.3133
8	303.7767	0.7	1.028e+03	0.0
9	307.05038	0.3189	537.1	2.935e-05
10	315.8522	0.0	593.5	0.0
11	1458.766	0.7	475.5	1.0
12	395.85886	0.7	651.4	0.0
13	307.1958	1.0	711.7	1.0
14	301.7545	0.0	772.7	0.0
15	315.97437	0.0	338.4	0.0
16	1172.1052	1.0	385.4	1.0

2.3. tabula

Bajesa optimizācija tehnikas analīzei, izmantojot vienu mērķi "Price Close" ar diviem slēptajiem slāņiem.

Iterācija	MAE	Reg	Neurons 1. slānī	Neurons 2. slānī	Dropout
1	528.8	0.2622	977.4	752.8	0.5987
2	705.2	0.1092	161.2	60.65	0.8662
3	2.498e+0	0.4208	728.2	22.14	0.9699
4	650.7	0.5827	219.1	187.7	0.1834
5	486.2	0.213	539.9	444.6	0.2912
6	306.8	0.2084	530.4	815.3	0.0
7	313.5	0.1609	567.1	812.7	0.007685
8	445.3	0.292	529.7	820.9	0.3404
9	320.2	0.0	534.7	752.0	0.0
10	503.6	0.1784	471.9	778.4	0.9547
11	318.8	0.0	627.3	748.9	0.0
12	308.1	0.0	684.5	812.1	0.0
13	321.3	0.0	715.4	734.3	0.0
14	331.2	0.7	771.7	802.5	0.0
15	323.9	0.0	738.0	886.2	0.0
16	310.4	0.0	826.9	878.0	7.819e-05
17	546.2	0.3631	795.0	955.7	0.6067
18	320.2	0.0	861.8	801.5	0.0
19	320.2	0.0	811.5	723.1	0.0

2.4. tabula

Bajesa optimizācija fundamentālajai analīzei, izmantojot vienu mērķi "Price Close" ar vienu slēpto slāni.

Iterācija	MAE	Batch	Dropout	Neuroni	Reg
1	1.642e+0	77.34	0.4957	22.14	0.9699
2	4.682e+0	106.7	0.1486	187.7	0.1834

3	9.765e+0	39.64	0.3673	444.6	0.2912
4	2.374e+0	102.9	0.2364	199.4	0.151
5	2.984e+0	128.0	0.0	261.1	1.0
6	1.26e+04	104.9	0.5684	188.3	0.5656
7	5.14e+03	98.56	0.4661	92.12	0.3344
8	2.826e+0	15.33	0.4174	140.2	0.4251
9	8.834e+0	31.63	0.532	524.5	0.1088
10	6.322e+0	103.3	0.6096	488.3	0.0545
11	2.167e+0	100.2	0.3928	569.5	0.1771
12	1.852e+0	29.58	0.7	610.4	0.4106
13	6.594e+0	81.92	0.4915	673.1	0.3731

2.5. tabula

Bajesa optimizācija rakstu atpazīšanai ar vienu paslēpto slāni, izmantojot vienu mērķi - cenas aizvēšanu

Iterācija	MAE	Batch	Dropout	Neurons	Reg
1	487.5	48.57	0.6655	752.8	0.5987
2	633.1	20.81	0.1092	60.65	0.8662
3	782.3	77.34	0.4957	22.14	0.9699
4	412.2	106.7	0.1486	187.7	0.1834
5	474.4	39.64	0.3673	444.6	0.2912
6	1.291e+0	97.6	0.1417	254.5	0.09996
7	436.9	105.3	0.3774	188.0	0.5757
8	505.0	125.5	0.4991	166.5	0.3642
9	738.6	127.9	0.4273	199.4	0.4989
10	413.6	101.9	0.03436	166.3	0.09246
11	321.8	101.5	0.0	134.5	0.0
12	337.7	76.7	0.0	137.6	0.0
13	319.0	86.56	0.0	109.7	0.0
14	474.3	120.9	0.1795	103.4	0.8114
15	441.8	53.67	0.0	113.0	0.0

2.6. tabula

Bajesa optimizācija visiem datu kopumiem, izmantojot vienu paslēpto slāni, ar vienu mērķi - cenas aizvēršanu.

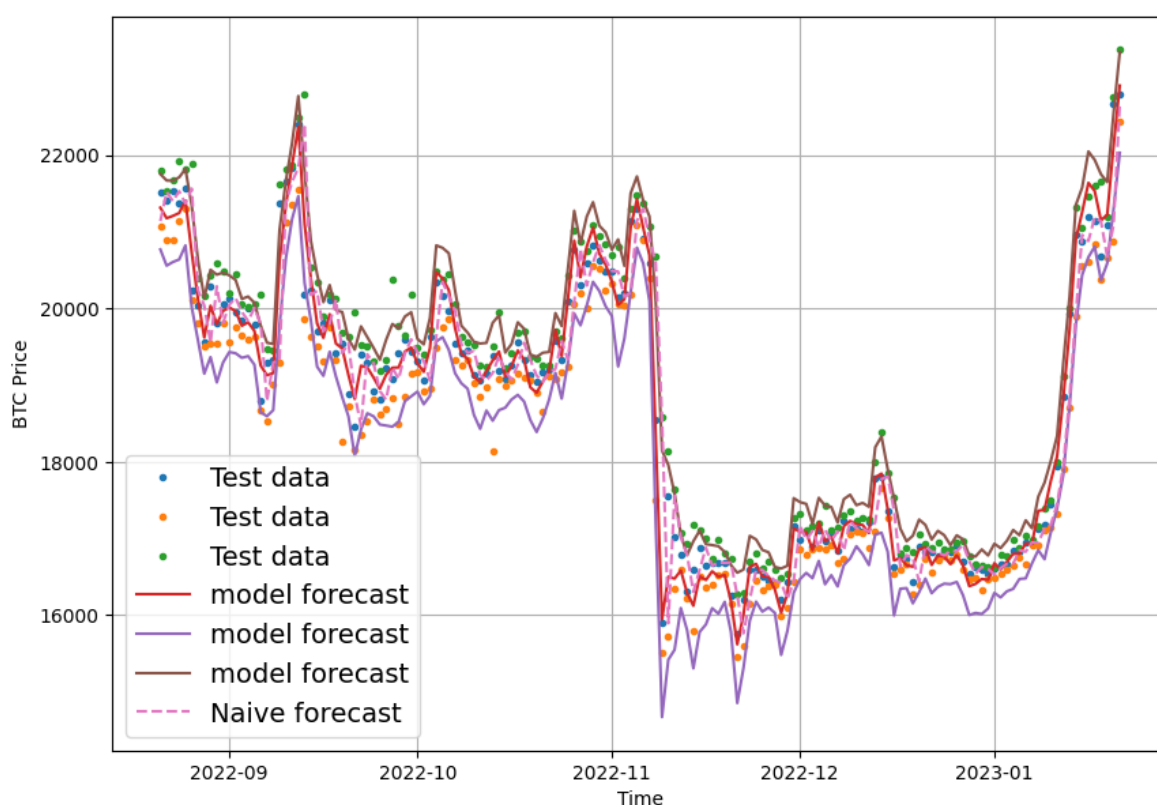
Iterācija	MAE	Batch	Dropout	Neurons	Reg
1	1.099e+0	78.71	0.09765	301.0	0.3664
2	1.177e+0	58.92	0.5496	206.1	0.5142
3	3.364e+0	76.24	0.03252	624.9	0.1705
4	2.033e+0	9.262	0.6642	992.7	0.8084
5	3.053e+0	39.69	0.06837	703.7	0.4402
6	4.392e+0	94.2	0.2042	325.6	0.309
7	1.316e+0	50.99	0.007995	309.2	0.2862
8	2.57e+04	57.39	0.7	279.2	0.0
9	2.549e+0	68.13	0.0	320.2	0.9653
10	7.824e+0	44.01	0.5099	293.7	0.1304
11	1.948e+0	35.01	0.2934	309.8	0.775
12	1.037e+0	76.56	0.3454	284.1	0.8754
13	2.661e+0	95.05	0.03382	294.4	0.7641
14	4.063e+0	108.1	0.0514	309.2	0.4147
15	1.714e+0	110.6	0.5576	330.7	0.05546

2.15. Vairākām izvades prognozēšana

Beigu datu kopai tika izvēlēta Tehniskā analīze un Modelu atpazīšana. Lielākā daļa no šīs datu kopas īpašībām tiek veidotas no slēgšanas, atvēršanas cenas, augstākās un zemākās dienas cenas. Ir divi veidi, kā veikt prognozēšanu nākotnē N dienām. Pirmajā metodei tiek apmācīts modelis ar logiem ar dažādiem horizonta skaitļiem, bet otrais veids ir prognozēt tikai nākamo dienu, pievienot šo jauno prognozi datu kopai un prognozēt N dienas, pakāpeniski prognozējot vienu dienu pa reizei. Lai piemērotu otro metodi, visām īpašībām jābūt izveidotām no esošās sveču īpašības (price_close, price_low, price_open, price_high). Atvēršanas cena ir iepriekšējā aizvēršanas cena, bet citas vērtības nav pieejamas. Tāpēc vairāku izvades regresija ir laba izvēle šai konkrētajai uzdevumam.

Šeit ir rezultāti viena slēptā slāņa Dense NN arhitektūrai ar 0.5 izsvītrosānu un 0.0001 L2 regulēšanu:

- Price_close**: 'mae': **387.11404**, 'mse': 349726.2, 'rmse': 591.3765, 'mape': 1.0884491
- Price_low**: 'mae': **1172.6205**, 'mse': 2045495.9, 'rmse': 1430.2084, 'mape': 3.382294
- Price_high**: 'mae': **869.72144**, 'mse': 1341614.0, 'rmse': 1158.2806, 'mape': 2.5557945
- Kopā**: 'mae': **410.91943**, 'mse': 360925.7, 'rmse': 479.06863, 'mape': 1.2052187



2.16. att. Vairāku izvades regresija par BTC cenu.

2.16. attēlā zilie punkti ir price_close, oranžie punkti ir price_low un zaļie punkti ir price_high. Sarkanā līnija ir modeļa prognoze par price_close, purpursarkana līnija ir modeļa prognoze par price_low, brūnā līnija ir modeļa prognoze par price_high. Rozā punktētā līnija ir Naīva prognoze.

2.16. Plānots darbs

Pirmkārt, galvenais iemesls šāda algoritma izveidei ir prognozēt nākotnes nepazīstamus datus, tāpēc ir būtiski - **pārbaudīt modeli ar validācijas datiem** (līdz pat šai dienai): Lai precīzi novērtētu BTC cenas prognozēšanas modeļa veikspēju, ir svarīgi to novērtēt ar validācijas datiem, kas paplašinās līdz pat pašreizējai dienai. Testējot modeli ar nesenajiem datiem, mēs varam iegūt ieskatu tā spējā prognozēt BTC cenas reāllaika scenārijos. Šī novērtēšana palīdzēs mums saprast modeļa izturību un uzticamību dinamiskos tirgus apstākļos.

Otrais solis ir pievienot skaitļošanas jaudu ar atkārtotā neironu tīkla kā **Long Short Term Memory (LSTM)** arhitektūru: Kā nākotnes darba daļa varētu būt Long Short-Term Memory (LSTM) arhitektūras izpēte, kas var būtiski uzlabot BTC cenas prognozēšanu. LSTM modeļi ir labi piemēroti, lai fiksētu ilgtermiņa atkarības laika sērijās, tāpēc tie ir ideāls izvēle kriptovalūtu cenu prognozēšanai. Ieviešot LSTM arhitektūru, mēs potenciāli varam uzlabot modeļa spēju fiksēt sarežģītas tendences un dinamiku BTC cenas kustībās, palielinot prognozēšanas precizitāti.

RL algoritma ieviešana: Pastiprinātā mācīšanās (Reinforcement Learning, RL) algoritms sniedz cerības pilnveidot BTC cenas prognozēšanas modeļus. RL tehnikas, piemēram, Q-mācīšanās vai Deep Q-Networks, ļauj modeļim mācīties optimālās darbības, balstoties uz atalgojumu, ko saņem no vides. Iekļaujot RL prognozēšanas procesā, modeļis var pielāgoties un pieņemt labākus lēmumus, ņemot vērā tirgus dinamiku un mainīgās tendences. RL algoritma ieviešana var palīdzēt optimizēt modeļa lēmumu pieņemšanas procesu un potenciāli uzlabot tā prognozēšanas spējas.

Kā ar tirdzniecību cilvēku? **Izveidot datorredzi (CNN 2D) attēla rakstīšanas atpazīšanai:** Iekļaujot datorredzes tehnikas, konkrēti Konvolūciju Neironu Tīklus (CNN) ar 2D arhitektūru, var pievienot papildu dimensiju BTC cenas prognozēšanai. Analizējot vizuālos datus, piemēram, kriptovalūtu biržu grafikus, modelis var atpazīt un interpretēt rakstīšanas tendences, kas var ietekmēt BTC cenas kustību. CNN 2D integrācija var sniegt vērtīgu ieskatu tirgus tendencēs un uzlabot kopējo prognozēšanas precizitāti.

Apvienot visus modeļus vienā kopā: Apvienojot vairākus dziļās mācīšanās arhitektūras, RL algoritmus un datorredzes modeļus, kas izstrādāti iepriekšējos soļos, vienā kopā, var panākt precīzākas un noturīgākas BTC cenas prognozes. Izmantojot

individuālo modeļu stiprās puses, šī kopā var nodrošināt visaptverošu prognozi, kas ņem vērā dažādas perspektīvas un palielina prognozēšanas veikspēju.

Nepieciešamības prognozēšana, izmantojot **pierādījumu dziļo mācīšanos**: Papildus BTC cenas prognozēšanai ir svarīgi novērtēt prognozēšanas nenoteikto dabu. Pierādījumu dziļās mācīšanās tehnikas, piemēram, Monte Karlo metode, Dropout, Bajesa neironu tīkli vai MIT Capsa, var sniegt nenoteiktus novērtējumus, iegūstot modeļa pārliecību par tā prognozēm. Iekļaujot pierādījumu dziļo mācīšanos, mēs varam kvantificēt nenoteiktību BTC cenas prognozēs, kas var būtiski noderīga risku pārvaldībai un lēmumu pieņemšanai.

Testēt esošu algoritmu ar atpakaļējās pārbaudes stratēģiju ieguvumiem: Lai novērtētu BTC cenas prognozēšanas modeļa praktisko izmantojamību, ir svarīgi to testēt ar esošu algoritmu, izmantojot atpakaļējās pārbaudes stratēģiju. Atpakaļējā pārbaude ietver modeļa pielietošanu vēsturiskiem datiem, lai novērtētu tā veikspēju simulētā tirdzniecības vidē. Analizējot atpakaļējās pārbaudes iegūtos ienesīguma un riska rādītājus, mēs varam noteikt modeļa efektivitāti rentablu tirdzniecības stratēģiju ģenerēšanā.

Visbeidzot, pēc iepriekšējo soļu ieviešanas, būs laiks izveidot **serveri automātiskai pirkšanai/pārdošanai, balstoties uz prognozēm**: Lai vēl vairāk izmantotu BTC cenas prognozēšanas modeli, var izstrādāt automatizētu tirdzniecības sistēmu. Izveidojot serveri, kas integrē modeļa prognozes ar izpildes platformu, piemēram, kriptovalūtu biržas API, mēs varam automatizēt BTC pirkšanas vai pārdošanas procesu, balstoties uz modeļa signāliem. Šāda servera sistēma var uzlabot efektivitāti un ļaut pieņemt reāllaika tirdzniecības lēmumus, izmantojot modeļa prognozēšanas spējas finanšu ieguvumiem.

REZULTĀTI UN SECINĀJUMI

Kvantitatīva analīze, izmantojot autoru izveidotu dziļās mācīšanās modeli, var novērst cilvēka emocijas un būt ļoti noderīga peļņas veidojošo darījumu veikšanā. Tos var arī izmantot kā tehniskus brīdinājumus un lietot kopā ar fundamentālo analīzi, izmantojot sociālo tīklu sentimentu kopsavilkumu, ar valodas modeļu palīdzību. Neironu tīkli nodrošina kvantitatīvāku ietvaru, lai identificētu un izsekotu sarežģītās attiecības, kas pastāv finanšu tirgos.

Turpmākajā darbā ieteicams izpētīt Long Short-Term Memory (LSTM) arhitektūras un Reinforcement Learning (RL) algoritmu integrāciju, lai uzlabotu BTC cenas prognozēšanas modeli. Turklāt datorredzes tehniku izmantošanas izpēte, piemēram, konvolūcijas neironu tīkli (CNNs), attēlu rakstura atpazīšanai, varētu tālāk uzlabot modeļa prognozēšanas spējas.

Izmantojot atvērto API, ir viegli iegūt visus nepieciešamos datus, lai izveidotu prognozēšanas modeļus, šajā darbā "coinmetrics" api palīdzēja iegūt ikdienas cenas datus, un python bibliotēkas tika izmantotas, lai izveidotu citas funkcijas no šiem datiem. Šis process ir viegli automatizējams un to var izmantot galīgajā produktā (automatizēta tirdzniecības bota).

Naīva prognoze ir vienkāršs rīks, kas tika izmantots kā pamatvērtība vienai cenas mainīgajai, lai to pārspētu, tika testētas dažādas DL arhitektūras un tikai dažas no tām varēja to nedaudz pārsniegt.

Neskatoties uz teoriju par stacionaritāti, BTC cenas mainīgā pārveidošana stacionārā, izmantojot box-cox, tad veicot prognozi par testa datiem un atjaunojot to normālā, rāda sliktāku veikspēju ar blīvu neironu tīklu nekā bez šīs transformācijas.

Aktivācijas funkcijas analīze parāda, ka šim konkrētajam uzdevumam visbiežāk lietotā ReLU ir labākā no visām pārbaudītajām. $\text{ReLU} > \text{GELU} > \text{ELU} > \text{Tanh}$. Blīvs NN ar Tanh aktivizācijas funkciju nevar apgūt iekšējos datu raksturus un veikt pareizas prognozes.

Visu pieejamo datu pievienošana galīgajam datu kopām nav pareizais risinājums, kā tas pierādījās praksē, tā vietā tika izmantota tikai Tehniskā Analīze (TA) un Rakstura Atzīšana (PR), lai sasniegtu labāko veikspēju.

Izmantojot Dropout un regularizāciju tika iegūts vislabākais modeļa rezultāts:

Vislabākā kopējā viena mērķa (cenas slēgšana) prognoze tika sasniegta, izmantojot Bajesa Optimizāciju TA datu kopā, kas ir **282.3 MAE**, gandrīz **3** reizes labāk nekā naivas prognozes pamatvērtība (**833 MAE**).

Daudz izvades prognoze rāda labu veikspēju, noķerot cenas punktus tās prognozes intervālā. Kā parādīts 2.16. att. No visiem datu punktiem price_close, price_low, price_high, 80% atrodas starp price_high un price_low. Tirdziniekiem, kuri veic ilgtermiņa darījumus, cenas intervāli var parādīt vairāk nekā vienu cenas prognozi.

Modelis kopumā parāda labu veikspēju, bet tas joprojām ir Alpha versija un, lai izveidotu pilnībā autonomu tirdzniecības produktu, ir nepieciešams turpmākais darbs.

IZMANTOTIE INFORMĀCIJAS AVOTI

- [1] Satoshi Nakamoto *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2009. gadā.
Available from: <https://bitcoin.org/bitcoin.pdf>
- [2] Jake Frankenfield. *Bitcoin Mining*. 26. aprīlī, 2023. gadā
Available from: <https://www.investopedia.com/terms/b/bitcoin-mining.asp>
- [3] Lisa Hladik *Methods of Stock Market Prediction* 2019. gadā.
Available from: <https://muse.union.edu/2019capstone-hladikl/methods-of-stock-market-prediction-2/>
- [4] John J Murphy – *Technical Analysis Of The Financial Markets*
Available from:
<https://cdn.preterhuman.net/texts/unsorted2/Stock%20books%20029/John%20J%20Murphy%20%20Technical%20Analysis%20Of%20The%20Financial%20Markets.pdf>
- [5] Tableau.com *Time Series Analysis: Definition, Types, Techniques, and When It's Used*
Available from: <https://www.tableau.com/learn/articles/time-series-analysis>
- [6] Rob J Hyndman, George Athanasopoulos *Forecasting: Principles and Practice*
Available from: <https://otexts.com/fpp3/index.html>
- [7] Multichain – *Customizing blockchain parameters*
Available from: <https://www.multichain.com/developers/blockchain-parameters/>
- [8] Expert.ai *What Is Machine Learning? A Definition*. 14. martā 2022. gadā
Available from: <https://www.expert.ai/blog/machine-learning-definition/>
- [9] IBM *What is Supervised Machine Learning?*
Available from: <https://www.ibm.com/topics/supervised-learning>
- [10] Adam Hayes *Autoregressive Integrated Moving Average (ARIMA) Prediction Model* 18. decembrī 2022. gadā
Available from: <https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp#:~:text=An%20autoregressive%20integrated%20moving%20average%2C%20or%20ARIMA%2C%20is%20a%20statistical,values%20based%20on%20past%20values.>
- [11] Michael Skirpan *How do Neural Networks Learn?*

Available from: <https://www.kdnuggets.com/2015/12/how-do-neural-networks-learn.html>

[12] Mbali Kalirane - *Gradient Descent vs. Backpropagation: What's the Difference?*

Available from: <https://www.analyticsvidhya.com/blog/2023/01/gradient-descent-vs-backpropagation-whats-the-difference/#:~:text=Gradient%20descent%20is%20used%20to%20find%20a%20weight%20combination%20that,point%20of%20the%20cost%20function.>

[13] Alexander Amini, Ava Amini *MIT Introduction to Deep Learning 2023*. gadā

Available from: <http://introtodeeplearning.com>

[14] Christopher Olah *Understanding LSTM Networks*

Available from: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

[15] Gavin Bell – *Forecasting in power markets: how good is the naive forecast?* 1. septembrī 2021. gadā

Available from: <https://www.optimeering.com/post/forecasting-in-power-markets-how-good-is-the-naive-forecast#:~:text=In%20time%20series%20forecasting%2C%20the,simple%20of%20all%20forecasting%20methods>