

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
INFORMATIKOS KATEDRA

Kursinis darbas

Automatizuotas kriptovaliutų prekybos robotas
(Automated cryptocurrency trading bot)

Atliko: 4 kurso 1 grupės studentas

Matas Kaminskas (parašas)

Darbo vadovas:

J. Asist., Dr. Igor Katin (parašas)

Vilnius
2023

Turinys

Ivadas	2
1. Pagrindinė tiriamaoji dalis	3
1.1. Darbinė aplinka	4
1.1.1. Python	4
1.1.1.1. Python-binance API	4
1.1.2. Binance	4
1.1.2.1. Binance Spot Test Network	4
2. Laiko eilučių analizė	6
2.1. Laiko eilutės	6
2.2. Laiko eilučių stacionarumas	6
2.3. ACF ir PACF	7
2.4. ADF testas	7
3. Autoregresiniai modeliai	9
3.1. AR modelis	9
3.2. MA modelis	9
3.3. ARMA modelis	9
3.4. ARIMA modelis	10
3.5. SARIMA modelis	10
4. Metodika	12
4.1. Modeliai	12
4.2. Duomenys	12
4.3. Modeliai	13
4.4. Prognozavimas	13
4.5. Vertinimas	14
5. Realizacija	15
5.1. Modelių realizacija	15
5.2. Prognozės	16
6. Rezultatai ir išvados	19
Sąvokų apibrėžimai	20

Įvadas

Pasaulis vis labiau modernėja, technologijos tampa vis išmanesnės ir našesnės, nei bet kada anksčiau. Šiais laikais turime kontraversiškai pagarsėjusią kriptovaliutų rinką, kuri yra kaip skaidresnė atsvara standartinei akcijų biržai. Po 2007-2008 metų įvykusios finansų krizės, visai neužilgo - [Nak08] 2008 metais, atsirado pirmoji kriptovaliuta - Bitcoin. Tai yra decentralizuotą ir nepriklausoma elektroninę valiutą, paremta blokų grandinės technologija. Dabartiniais apskaičiavimais šios rinkos bendra vertė viršija 800 milijardų JAV dolerių [Coi], nors kiek daugiau nei prieš metus ši vertė buvo dvigubai didesnė - 1.6 trilijonų JAV dolerių, o tarp 2018 ir 2020 metų vertė daugmaž buvo stabili - apie 100 milijardų JAV dolerių. Šie duomenys dar kartą pabrėžia kriptovaliutų rinkos nepastovumą, bei ženkliai padidėjusi susidomėjimą, populiarumą visuomenėje.

Pagrindinis skirtumas tarp kriptovaliutų rinkos ir tradicinės akcijų biržos yra jog mainai vyksta 24 valandas per parą. Akcijų biržoje yra nustatytos pagrindinės darbo valandos, jos dažniausiai yra tokios pat kaip vietos darbo valandos ir pagrindiniai mainai vyksta šiomis valandomis, todėl stebėti tendencijas ir realizuoti strategijas yra įmanomas darbas asmeniui. Kai rinka yra prieinama visa parą dirbant be sustojimo yra nepraktiška, bei neįmanoma stebėti ir atlikti prasmingus mainus rinkoje. Atsiranda puiki terpė automatizuoti šį procesą - įdarbinti automatizuotą prekybos robotą, kuris gebėtų analizuoti rinkos duomenis ir prognozuoti tolimesnę rinkos eigą.

Šio darbo tikslas yra analizuoti ir ištirti ko reikia automatizuoto kriptovaliutų prekybos roboto kūrimui. Darbe tiriama, kokį autoregresinių modelių galima taikyti: AR, ARMA, ARIMA, SARIMA tiksliausiai nuspėti kriptovaliutos kainas ateityje. Pagal gautus rezultatus ir tikslumą galima spręsti, koks modelis yra tinkamiausias robotui naudoti.

Tikslui įgyvendinti keliama uždaviniai:

- Mokslinės literatūros analizė,
- Atlikti autoregresinių modelių analizę,
- Skirtingų autoregresinius modelių pritaikymas duomenims,
- Ištirti gautus rezultatus atlikus modelių taikymą.

1. Pagrindinė tiriamoji dalis

Pagrindinė ir pirmoji kriptovaliuta - Bitcoin. Nors pirmieji 50 BTC buvo "iškasti" 2009 metų pradžioje, Bitcoin susilaukė didesnio dėmesio tik 2013 metais[Mac⁺14]. Ši kriptovaliuta buvo laikoma ateities valiuta dėl savo naujoviškumo ir decentralizavimo, bet susilaukdavo ir neigiamų atsiliepimų dėl atsiskaitymo nelegaliuose prekybos vietuose internete. Vienas pirmųjų ir itin didelės apyvartos sulaukęs nelegalių prekių tinklalapis "Silkroad" atsiskaitymui už prekes naudojo Bitcoin. Didelė apyvarta skatino bitcoin augimą, susidomėjimą ir kitų kriptovaliutų progresą. Stebint kriptovaliutų rinką matoma koreliacija tarp Bitcoin kainos ir visų kitų kriptovaliutų kainos, taip dar kartą pabrėžiama, jog ši valiuta yra pagrindinė rinkoje. Kyla bitcoin kaina - kyla kitų valiutų kaina, krenta bitcoin kaina - krenta kitų valiutų kaina. Retesniais atvejais kriptovaliutos sparčiau keičia kainą dėl pažangaus vystymosi ar dėl esančių problemų.

Dauguma kriptovaliutų yra riboto kiekio, tad jeigu prekybos robotas uždirba konkrečios kriptovaliutos mažai, jų kainą gali keleriopai išaugti tikrų valiutų atžvilgiu, todėl į kriptovaliutą galima žiūrėti ir kaip į investiciją. 2018 metais, nepaisant to, kad kriptovaliuta buvo galima atsiskaityti už paslaugas ar prekes, populiariausia paskirtis buvo investicija. Viena iš priežasčių - tuo metu kaina buvo pakilus nenusipėjusiai aukštai[Gar⁺18]. Akcijų birža, rinka naudojama investiciniais tikslais, jau kurį laiką yra analizuojama dėl savo finansinės naudos ir sudėtingumo. Šios rinkos turi panašumų, todėl galima tirti akcijų biržos analizę literatūroje.

Šiais laikais vis daugiau žmonių naudojami technologijomis ir nori nepriklausomybės nuo institucijų išleidžiamų ir manipuliuojami valiutų, infliacijos ir kitų dalykų, kurie mažina finansinių institucijų pasitikėjimą. Kriptovaliutos yra paremtos blokų grandinės technologija, todėl daugelis jų yra decentralizuoti tinklai. Visa blokų informacija yra viešai prieinama publikai, puikiai žinoma tarp kokių šalių vyksta sandoriai, taip suteikiamas skaidresnis ir patikimesnis būdas publikai turėti savo nepriklausomą rinką. Ypatingas kriptovaliutų bruožas yra tas, kad jų paprastai neišleidžia jokia centrinė institucija, todėl teoriškai jos nėra apsaugotos nuo vyriausybės kišimosi. Ankščiau minėtos savybės suteikia anonimiškumo, todėl tai sudaro palankias sąlygas klestėti sukčiavimo atvejams, "pump-and-dump" ar "ponzi" schemoms. Dažniausiai sutinkama yra "pump-and-dump" schema, kuri apibūdina procesą, kai valiutos paklausa yra iš anksto apgalvota ir trumpam padidinama, taip pakeliant jos kainą, vėliau, kai kainą yra pasiekusi tinkamą tašką, organizatoriai, staigiai parduoda savo turimą kiekį, taip pasipelnędami[XL19].

Pagrindinis tyrimas šiame darbe yra tinkamiausio auto regresinio modelio parinkimas automatizuotam robotui lošėjui. Lošimo procesas tam tikrą laiką gali būti atliekamas žmogaus, stebėti besikeičiančias kainas ir atitinkamai elgtis, tačiau toks darbas yra monotoniškas ir atsiranda natūrali terpė jį automatizuoti. Automatizavimui reikia taikyti pasirinktą strategiją, kuri spęstų kada yra tinkamas metas įsigyti kriptovaliutą, o kada parduoti, bet pirmiausia reikalingas tinkamas prognozavimo modelis. Auto regresiniai modeliai taikomi laiko eilutėmis yra paplitę statistikoje norint prognozuoti tendenciją ateityje. Modelio prognozės tikslumui bei tinkamumui palyginti bus naudojama santykinė ir absoliuti klaidos.

1.1. Darbinė aplinka

Šiam robotui ir statistiniams modeliams teks naudotis jau sukurtais įrankiais, kurie palengvins darbą. Naudojama Python programavimo kalba, kadangi ši kalba turi patogų API su dauguma kriptovaliutų rinkų ir patogias bibliotekas autoregresyviems modeliams bei kitiems reikalingiems API.

1.1.1. Python

Python yra aukšto lygio programavimo kalba. Joje galima sutikti ne viena programavimo paradigmą: procedurinę, objektinę ir netgi funkcinę. Ši kalba suteikia patogų abstrakcijos lygį šiai užduočiai - automatizuoti robotą lošėją ir panaudoti AR modelius. Šiais laikais python yra viena populiariausių kalbų pasaulyje, dėl savo paprastumo ir paprasto naudojimo naujam vartotojui, bet tikrai yra daugybę savybių kuriomis prireiktų ne vienus metus suprasti ir prasmingai naudoti programuojant. Dėl šios kalbos privalumų dauguma egzistuojančių API bibliotekų palaiko Python programavimo kalbą.

1.1.1.1. Python-binance API

Šiame darbe naudojama "python-binance" API. Šis API yra "Wrapper" oficialiam Binance API - binance-connector-python. Oficialus API yra labai paprastas ir lengvasvoris, todėl didelio patogumo nėra, tik paprasčiausios užklausos, kurias visvien reiktų apdoroti, labai padeda padaryti šis wrapper API - python-binance.

1.1.2. Binance

Binance yra didžiausia kriptovaliutų birža pagal prekybos apimtį, turinti itin didelį valiutų pasiulą ir patogiai prieinamus duomenis. Ši birža taip pat suteikia galimybę "lošti" kriptovaliutomis testiniame tinkle. Tinklo paskirtis yra lengvai nuspėjama iš pavadinimo, jame galima atlikti tuos pačius veiksmus kaip realioje rinkoje, tik naudojama netikra valiuta kuri tikros vertės neturi.

1.1.2.1. Binance Spot Test Network

Testinis tinklas yra beveik identiškas tikrajam "Binance" tinklui, kainos yra vienodos kaip ir pagrindiniame tinkle ar pasaulyje, todėl galima analizuoti seniausius rinkos duomenis ir juos taikyti reikiame modelyje. Testinis tinklas nėra toks populiarus, tad jame yra ženkliai mažiau prekiaujančių žmonių. Testiniame tinkle pašyra sukuriami naudojant "Github" prisijungimą. Kiekvienas naujas vartotojas turi pradinį balansą susidendantį iš kriptovaliutų matomų 1 lentelėje.

1 lentelė. Pradinis likutis

Kripto valiuta	Kiekis
BNB	1,000
BTC	1
BUSD	10,000
ETH	100
LTC	500
TRX	500,000
USDT	10,000
XRP	50,000

Šiuo likučiu galima elgtis kaip norima. Pradinis likutis yra pakankamas atlikti prasmingoms transakcijomis testiniame tinkle. Reikėtų pabrėžti, jog testinis tinklas kas mėnesį laiko yra iš naujo nustatomas ir visas turimas likutis yra konvertuojamas atgal į pradinį likutį, išvalant ankščiau buvusį balansą, tad jeigu nepavyko praturtėti prekiaujant kriptovaliutomis, vėl įgaunama proga pradėti prekybą, na o sėkmingu atveju visas pelnas yra ištrinamas ir reikia vėl pradėti nuo nulio.

2. Laiko eilučių analizė

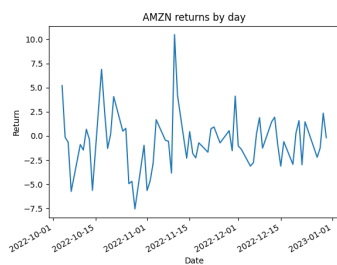
”Visi modeliai yra klaidingi. Kai kurie modeliai yra naudingi” (angl. ”All models are wrong. Some models are useful”) - citata priskiriama George Box, kurios reiktų nepamiršti analizuojant laiko eilutes. Šis analizės būdas bando pastebėti pasikartojančius modelius bėgant laikui, jog būtų galima atlikti kuo tikslesnes prognozes apie ateitį. Dažnu atveju tai yra vienas ar keli izoliuoti kintamieji, kurie yra stebimi nustatytą laiko tarpą ar surenkant jų istorinius duomenis. Laiko eilučių duomenų rinkimo ir analizavimo pavyzdžiai: sekamas paciento širdies pulsas, prekyboje sandėlio prekių kiekio priežiūra, konkrečios akcijos kainos prognozavimas akcijų biržoje. Laiko eilučių analizės vienas iš tikslų yra nuspėti kaip kinta reikšmė bėgant laikui, tai ir yra ko prireiks norint prognozuoti kriptovaliutų kainą šiam robotui. Kriptovaliutų kainų duomenys, taip pat yra laiko eilutė, nes kaina yra sekama bėgant laikui, tačiau dažniausiu atveju tai nėra stacionari laiko eilutė. Reikia pabrėžti, kad toks analizės būdas tik bando nuspėti koks yra labiausiai tikėtinas rezultatas remiantis turimais duomenimis ir naudojamu modeliu.

2.1. Laiko eilutės

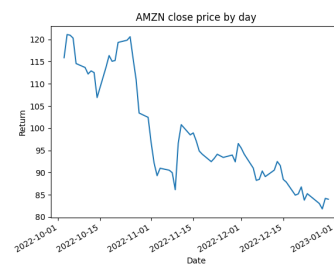
Laiko eilutė yra chronologiškai surikiuotas duomenų rinkinys. Laiko eilučių analize analitškai gali pastebėti įvairias tendencijas ir taip prognozuoti reikšmes ateityje bei geriau pasiruošti įvykiams. Laiko eilutė gali būti stacionari arba ne stacionari. Šiame darbe norint analizuoti turimus duomenis naudojant autoregresyviuosius modelius reikia turėti stacionarią laiko eilutę, jog būtų galima pritaikyti autoregresiniui modeliui. Tačiau kriptovaliutų kainą dažnu atveju yra ne stacionari laiko eilutė, todėl prieš dirbant, reikia atitinkamai šiuos duomenis susitvarkyti arba analizuoti kitas eilutes.

2.2. Laiko eilučių stacionarumas

Stacionari laiko eilutė yra laikoma tokia eilutė, kurios statistinės savybės bėgant laikui nekinta [Nas06]. Tai reiškia, jog laiko eilutės vidurkis, variacija ir kovariacija turi pastovias reikšmes. Stacionarios laiko eilutės yra lengviau analizuojamos, todėl autoregresyvieji modeliai yra pritaikyti dirbti su stacionariais duomenimis. Autoregresinių modelių taikymas nestacionariems duomenims gali suteikti nepatikimas prognozes ir prastesnius rezultatus. Prieš dirbant su duomenimis, juos galima pasiversti stacionariais, panaikinant sezoniskumus ar tendencijas imtyje.



(a) Stacionari laiko eilutė



(b) Nestacionari

1 pav. Stacionari ir nestacionari laiko eilutė

Grafai parodantys skirtumą tarp stacionarios ir nestacionarios laiko eilutės. Grafuose naudojama "Amazon" kompanijos (NASDAQ: AMZN) akcijos duomenys gauti iš "Yahoo Finance" naudojant Python API[Aro21]. Pirmajame (stacionariame) grafе yra pavaizduota kiekvienos dienos grąža investuojant į AMZN akciją trijų mėnesių laikotarpyje. Stacionarumas matomas ir vizualiai - vidurkis imtyje išlieka panašus, centruojasi ties 0, taip pat variacija ir kovariacija yra panašūs visuose taškuose. Antrame grafе yra pavaizduota tos pačios akcijos uždarymo kaina tokiam pat laikotarpyje. Nestacionarumas matomas ir vizualiai, kadangi yra akivaizdi tendencija žemyn. Šiuos teiginius patvirtina ir ADF testas.

2.3. ACF ir PACF

Autokoreliacijos funkcija (ACF) (angl. autocorrelation function) ir dalinė autokoreliacijos funkcija (PACF) (angl. partial autocorrelation function) naudojamos analizuoti laiko eilutes, padedančios suprasti autokoreliaciją duomenų rinkinyje. Pritaikant ACF bei PACF galima lengviau parinkti ARMA(p, q) ir ARIMA(p, d, q) modelių pradines reikšmes [AR17]. Ženkliai didesnės reikšmės, pritaikius ACF laiko eilutei su tam tikru lagu, parodo optimalias ARMA ir ARIMA modelių slankaus vidurkio laipsnio (q) reikšmes. PACF tokiu pat būdu nurodo p - autoregresinio parametro reikšmę.

2.4. ADF testas

ADF (Angl. Augmented Dickey-Fuller) testas nustato ar laiko eilutė yra stacionari. Tai yra dažnai naudojamas įrankis analizuojant laiko eilutes ir jų algoritmais, kurie tikisi stacionarių duomenų[Chi18]. Stacionariai laiko eilutėje privaloma atmesti nulinę hipotezę tam tikrame užtikrintumo intervale. Jeigu ADF testo absoliuti statistinė vertė yra daugiau už kritinę vertę, nulinę hipotezę yra atmetama ir ši laiko eilutė yra stacionari. pritaikius ADF testą[SP10] AMZN akcijų duomenims (1 pav.) gauti rezultatai:

Time Series	ADF Statistic	p-value	1% Critical Value	5% Critical Value	10% Critical Value
Daily Returns	-7.087727	0.000000	-3.542	-2.910	-2.593
Daily Close Price	-1.640623	0.461935	-3.542	-2.910	-2.593

Pagal šiuos rezultatus

- ADF statistika yra skaičius, kuris lyginamas su kritinėmis vertėmis iš lentelės (skirtingais lygiais), kad būtų nuspręsta, ar atmesti ar priimti nulio hipotezę apie stacionarumą.
- Pirmu atveju ADF statistika dienos grąžoms yra -7.087727, kuri mažesnė, nei kritinė vertė -3.542 1% lygyje. Todėl nulinė hipotezė yra atmesta ir laiko seka laikoma stacionaria.
- Antru atveju ADF statistika dienos uždarymo kainoms yra -1.640623, kuris yra nemažesnė, nei kritinė vertė -3.542 1% lygyje. Todėl nulio hipotezė nėra atmesta ir laiko seka laikoma nestacionari.

3. Autoregresiniai modeliai

Vienas iš būdų analizuoti laiko eilutės yra autoregresija ir ją naudojantis modeliai. Egzistuoja ne vienas autoregresija naudojantis modelis. Populiarusi ir dažniausiai sutinkami modeliai yra ARMA(p, q) ir ARIMA(p, d, q). Šie modeliai taip pat naudojami anksčiau nepaminėtu MA modeliu kuris yra slankaus vidurkio modelis (angl. MA - moving average). Taip pat yra SARIMA, SARFIMA ir kitų modelių, turinčių savo specifinius panaudojimo atvejus. Toliau apžvelgsime detaliau autoregresinius modelius ir kaip jie galėtų būti panaudoti išskeltiems užduotims šiame darbe spręsti.

3.1. AR modelis

AR (angl. autoregressive) modelis remiasi tik praeities duomenimis, kad nuspėti kintamojo reikšmę ateityje, ieškoma ar pastebimas pasikartojantis modelis, kuris padėtų tiksliau nuspėti dydį ateityje[Chi18]. Šis modelis dar dažnai vadinamas ARp modeliu, nes naudojamas kintamasis "p", nusakantis kiek praeities reikšmių iš laiko periodo norima naudoti. Laikant, kad kintamasis X yra laiko eilutės kintamasis AR(p) modelio formulė gali atrodyti taip:

$$X_t = \Phi_1 X_{t-1} + \epsilon_t$$

X_t - stacionari laiko eilutė, Φ - AR modelio koeficientas, P - AR modelio laipsnis, ϵ_t - AR modelio paklaida.

3.2. MA modelis

Toliau tyrinėjami autoregresyvieji modeliai susideda iš dar vienos dalies - Slankaus vidurkio - MA (angl. Moving average). Slenkančio vidurkio dabartinė reikšmė tiesiškai priklauso nuo dabartines ir praeities reikšmių. Žymėjimas MA(q) reiškia q laipsnio slenkamąjį vidurkį, kurio formulę atrodo taip:

$$X_t = \epsilon_t - \sum_{i=1}^q \theta_i \epsilon_{t-i}$$

X_t - stacionari laiko eilutė, q - MA modelio laipsnis, ϵ_t - MA modelio paklaida.

3.3. ARMA modelis

ARMA (angl. autorogressive moving average) modelis yra autoregresyviaus ir slankaus vidurkio modelio junginys. Modelis dažnai žymimas kaip ARMA(p, q), kur p yra AR laipsnis, o q yra MA laipsnis. Pirmą kartą sujungtas 1938 mokslininko Herman Wold, jis pastebėjo jog ARMA modelis gal apimti dideles stacionarias laiko eilutes, kai yra tinkamai nurodytas p laipsnis ir q laipsnis[MH97]. Taip pat reikia pabrėžti, jog ARMA(0, q) = MA(q) ir ARMA(p, 0) = AR(p) Reiškia, jog X_t eilutė gali būti modeliuojama kaip kombinacija praeities x_t reikšmių ir/arba praeities e_t

klaidų:

$$X_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \theta_q e_{t-q}$$

3.4. ARIMA modelis

ARIMA (angl. autoregressive integrated moving average) modelis yra paplitęs nuo 1970-ųjų iki šių dienų. Modelio pavadinimo šifravimas pažodžiui: AR - Autoregresinis modelis, I - Integracija (Diferencijavimas), atsižvelgiama į duomenų tendenciją, MA - (Moving average) slankusis vidurkis. AR ir MA yra atskiri modeliai, kurie gali būti naudojami paprastesnei laiko eilučių analizei. Šio modelio privalumas, jog jis sujungia AR ir MA naudojimą kartu su diferencijavimu, kuris suteikia galimybę giliau analizuoti laiko eilutes. Diferencijavimo dalis turimus duomenis padeda paversti į stacionarius, kurie yra reikalingi korektiškam modelio veikimui. Diferencijavimas vyksta baigtinį kartų skaičių, jog būtų pasiekta beveik-validi stacionari būseną. Kitais žodžiais ARIMA yra ekvivalentus ARMA modelis tiems patiems MA ir AR laipsniams [Hua20]. Taigi ARIMA modelis yra paremtas ARMA modeliu. Pagrindinis skirtumas tarp šių modelių yra tas, jog ARIMA konvertuoja nestacionarius duomenis į stacionarius prieš dirbant su jais. Modelio tipas yra klasifikuojamas kaip ARIMA(p,d,q), kur p - autoregresyvoji dalis, d - integravimo (diferencijavimo) dalis, q - slenkančio vidurkio dalis. Visos ARIMA modelio reikšmės yra neneigiami sveikieji skaičiai [MSG14].

Dažniausiai yra 3 žingsniai naudoti ARIMA modelį:

1. Duomenų surinkimas ir verifikavimas - tikriname ar gauti duomenys yra stacionarūs ir ieškome sezoniškumo.
2. Diferencijavimas - vertimas laiko eilutę į stacionarią.
3. Prognozavimas naudojant ARIMA modelį - dažniausias taikomas ARIMA modelis yra ARIMA(1,1,0)

3.5. SARIMA modelis

SARIMA (angl. seasonal autoregressive integrated moving average) modelis yra ARIMA modelio išplėtimas turintis vieną papildomą savybę - sezoniškumą[CE20]. Pastebėjus, jog periodiškai kartojasi rezultatai laiko eilutėse, galima taikyti šį modelį. Geras ir paprastas pavyzdys, prekyba šventiniu laikotarpiu - kalėdos. Prekybos centruose, tuo metu padidėja prekybą, tačiau kitą mėnesį ženkliai sumažėja. Spartus sumažėjimas nereiškia, jog prekybai yra didelės problemos ir reikia pokyčių. Tai dažniausiai yra žmonių poilsis nuo pirkinių po šventinio laikotarpio. Šis modelis pastebi panašius sezoniškai atsitinkančius įvykius, juos atpažįsta ir pritaiko naudojant ankščiau minėtą ARIMA modelį.

SARIMA galima išreikšti kaip $ARIMA(p,d,q)(P,D,Q)[S]$, kur p,d,q standartiniai ARIMA modelio parametrai, o P - sezoniškumo autoregresivumo laipsnis, D - sezoniškumo diferencijavimo laipsnis, Q - sezoniškumo slankiojo vidurkio laipsnis ir S - sezoniškumo ciklio ilgis.

4. Metodika

Skyriuje aprašomi žingsniai, kurių buvo imtasi atliekant tyrimus ir analizę kuriant automatizuotą kriptovaliutų prekybos robotą. Tyrimo metu pagrindinis dėmesys buvo skiriamas tinkamų kriptovaliutų kainų analizės ir prognozavimo įrankiams ir metodams nustatyti bei skirtingų modelių veikimui įvertinti. Duomenys buvo renkami iš Binance testavimo tinklo naudojant python-binance API, o duomenims pritaikyti statistinės ir laiko eilučių analizės metodai. Kainų prognozavimui buvo naudojami autoregresyvūs modeliai, tokie kaip ARIMA ir SARIMA. Šio tyrimo išvados taps pagrindu kuriant prekybos robotą būsimame darbe.

4.1. Modeliai

Prognozavimui bus naudojami ARIMA ir SARIMA modelis. Kadangi kriptovaliutų kainų duomenys nėra stacionarūs, todėl naudojamas ARIMA modelis, kuris gali taikyti diferencijavimą. ARIMA naudoja anksčiau minėtus AR ir MA modelius, bei pritaiko diferencijavimą, taip galėdamas dirbti ir su nestacionariais duomenimis. Jeigu laiko eilutė yra nestacionari ir bandoma naudoti AR, MA ir ARMA modelius, rezultatai galimai bus klaidingi arba netikslūs. Taip pat naudojamas SARIMA modelis, nes jis yra ARIMA modelio pratęsimas tiriantis sezoniškumą laiko eilutėje, todėl dar bus patikrinama ar duomenyse egzistuoja sezoniškumas galintis padėti prognozuoti.

4.2. Duomenys

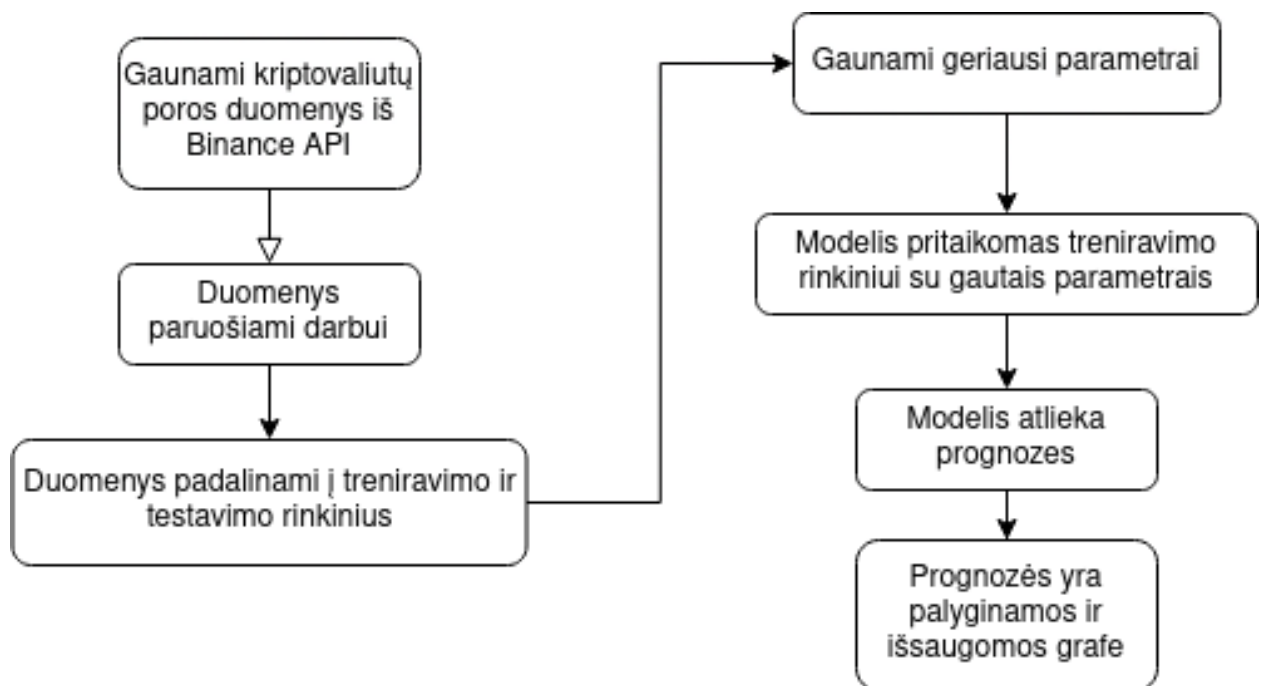
Istoriniai duomenys "Candlesticks" surinkti iš Binance testavimo tinklo naudojant python-binance API. Kadangi testinis tinklas kiekvieną mėnesį automatiškai išvalo visus duomenis, imami 2023m. sausio 4d. duomenys savaitei į priekį, todėl verta paminėti, kad šie duomenys galioja tik tam tikrą laikotarpį. Kadangi tinklas yra testinis, jame prekiaujama nevisomis galimomis kriptovaliutų poromis. Iš viso yra 20 skirtingų porų tinkle. Pasirinkta tirti šias poras: i) BTC/BUSD, ii) LTC/BUSD, iii) XRP/BUSD. Gaunamuose duomenyse yra daugiau informacijos bet bus naudojama tik datos ir 'close' stulpelis, parodantis konkrečios kriptovaliutos uždarymo kainą tam tikru laiku. Tai yra paskutinė kaina, už kurią buvo prekiaujama tam tikru laikotarpiu, pavyzdžiui, valandą ar dieną. Uždarymo kaina yra svarbi, nes ji atspindi galutinę kainą, už kurią pirkėjai ir pardavėjai sutarė prekiauti kriptovaliuta. Duomenys bus naudojami tokie kokie yra ir iškart perduodami į modelius. Nagrinėjamos dvi populiariausios valiutos BTC ir ETH, taigi imama BTCBUSD ir ETHBUSD kriptovaliutų pora. Modeliui apmokyti naudojamas 90 procentų duomenų rinkinio dydis. Kitus 10% bandoma prognozuoti po vieną reikšmę į priekį. Modeliui atlikus prognozę, jis yra iš naujo apmokomas pridėjus tikrąją sekančią vertę prie mokymo rinkinio ir taip vėl yra prognozuojama kitą reikšmę, jau su naujai apmokytu modeliui.

4.3. Modeliai

Prognozuojant naudojamos ARIMA ir SARIMA pagalbinės bibliotekos[SP10] kuri aprašo šiuos modelius, bei euristicinis `auto_arima` algoritmas[Smi⁺–] padedantis nustatyti geriausius ARIMA ir SARIMA parametrus konkrečiam duomenų rinkiniui. SARIMA modeliui sezoniškumo intervalas naudojamas 4 - bandoma pamatyti ar atsiranda sezoniškumas kiekvieną valandą, nes duomenų intervalas yra 15min. Modelių vertinimui bus naudojama MSE ir MAPE paklaidos.

4.4. Prognozavimas

Kriptovaliutų kainų prognozavimo metodas naudoja anksčiau minėtus modelius ir Binance testinio tinklo duomenis. Žemiau matoma struktūrinė schema, parodanti bendrą procesą, kurį sudaro keli pagrindiniai žingsniai. Pirmas žingsnis yra duomenų gavimas iš binance API, bet jų gaunama daugiau negu reikia, todėl atsirenkami tik reikalingi duomenys, šis žingsnis yra svarbus norint įsitikinti, kad duomenys yra paruošti analizei. Po to pasirenkame modelį (ARIMA/SARIMA). Kitame žingsnyje yra gaunami geriausi parametrai konkrečiam duomenų rinkiniui ir modeliui, kad būtų optimalus veikimas. Tada taikome modelį treniruočių duomenims, kad apmokytų modelį tolimesniam prognozavimui. Po to modelis prognozuoja reikšmes ir jos yra saugomos, kad būtų galima palyginti su tikromis reikšmėmis. Paskutinis žingsnis - palyginti gautus rezultatus ir išsaugoti duomenis grafe. Pateikta struktūrinė schema padės geriau suprasti visą procesą.



2 pav. Struktūrinė schema

4.5. Vertinimas

Vidutinė kvadratinė paklaida (MSE) ir vidutinė absoliučioji procentinė paklaida (MAPE) yra dvi dažniausios metrikos, naudojamos prognozavimo modelių našumui įvertinti. MSE yra prognozuotų ir faktinių verčių vidutinio skirtumo kvadrato matas, o MAPE yra vidutinio procentinio skirtumo tarp numatytų ir faktinių verčių matas. Abu šie rodikliai gali būti naudingi lyginant ARIMA ir SARIMA prognozavimo kriptovaliutoms rezultatus. MSE yra gera metrika norint išmatuoti, kiek prognozės nukrypsta nuo faktinių verčių, mažesnė MSE reikšmė rodo tikslesnį modelį. MAPE yra gera metrika prognozavimo paklaidoms procentais matuoti, mažesnė MAPE reikšmė rodo tikslesnį modelį. Abi šios metrikos gali būti naudojamos vertinant modelių veikimą ir lyginant ARIMA ir SARIMA prognozavimo kriptovaliutoms rezultatus.

5. Realizacija

Šio darbo realizavimo ir rezultatų skyriuje pristatomas metodikos įgyvendinimas ir gauti rezultatai. Skyriuje pateikiamos duomenų ir prognozavimo rezultatų vizualizacijos, taip pat kodo fragmentai, demonstruojantys naudojamų modelių įgyvendinimą. Vizualizacijos suteikia aiškų vaizdą apie modelių veikimą ir prognozių tikslumą. Kodo fragmentai suteikia išsamų supratimą apie modelių įgyvendinimą. Šioje dalyje taip pat palyginami rezultatai, gauti naudojant ARIMA ir SARIMA modelius, leidžiantys įvertinti modelių veikimą.

5.1. Modelių realizacija

Iš pradžių prisijungiama prie testinio tinklo naudojant API ir naudotojo raktus

```
with open(CONFIG_FILE_NAME, 'r') as config_file:
    data = json.load(config_file)
    validate(instance=data, schema=config_schema)
    client = Client(data['api_key'], data['api_secret'], testnet=True)
```

Tuomet gaunami istoriniai duomenys iš API. Kaip minėta, jog Sausio 4d. įvykio atnaujinimas ir eksperimentiniais tikslais imama duomenys savaitė nuo pradžios į priekį. tuomet kviečiami funkcija, jog paverstų gautus duomenis į reikalingą struktūrą.

```
klines = client.get_historical_klines(symbol, Client.KLINE_INTERVAL_15MINUTE
                                     , end_str='11 Jan, 2023')

df = binance_klines_to_df(klines)
model_predict_arima(symbol, df)
model_predict_arima(symbol, df, seasonal=True)

def binance_klines_to_df(klines):
    klines = np.array(klines)
    df = pd.DataFrame(klines.reshape(-1, 12), dtype=float, columns=('Open Time', 'Open', 'High', 'Low', 'Close', 'Volume', 'Close time', 'Quote asset volume', 'Number of trades', 'Taker buy base asset volume', 'Taker buy quote asset volume', 'Ignore'))
    df['Open Time'] = pd.to_datetime(df['Open Time'], unit='ms')
    return df
```

Turint duomenis kviečiama model_predict_arima funkcija, kuri pagal seasonal parametą atlieka ARIMA arba SARIMA modelį.

```
df = df[['Open Time', 'Close']]
df.set_index('Open Time', inplace=True)

to_row = int(len(df) * 0.9)
training_data = list(df[0:to_row]['Close'])
```



```

if seasonal:
    order, seasonal_order = best_params_arima_seasonal(training_data)
else:
    order = best_params_arima(training_data)
seasonal_order = None

testing_data = list(df[to_row:]['Close'])
model_predictions = []
for i in range(len(testing_data)):
    model = ARIMA(training_data, order=order, seasonal_order=seasonal_order)
    model_fit = model.fit()
    output = model_fit.forecast(steps=1)
    yhat = output[0]
    model_predictions.append(yhat)
    actual_test_value = testing_data[i]
    training_data.append(actual_test_value)

```

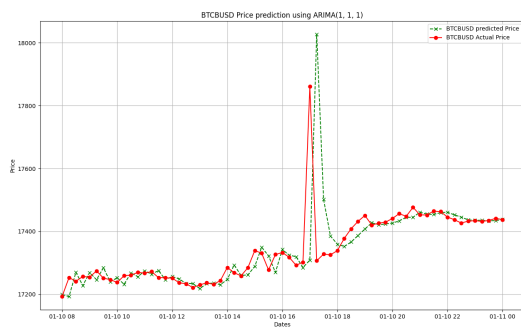
Pagrindinė kodo dalis atliekanti skaičiavimus Pasiemama is duomenų tik data ir uždarymo kainą, tuomet duomenys padalijami 90% apmokymui ir kita dalis prognozavimui. Tuomet modeliui gaunami optimalūs parametrai. Po šito inicijuojamas tuščias sąrašas, vadinamas `model_predictions`. Kiekvienai iteracijai atliekami šie veiksmus:

- sukuria ARIMA modelį turimais duomenimis ir parametrais ir jis yra pritaikomas naudojant `fit()`.
- gaunama prognozuojama reikšmė `yhat`
- prie prognozių sąrašo pridedama reikšmė
- į turimus duomenis pridedama tikroji reikšmė ir toliau cikliška apmokamas modelis

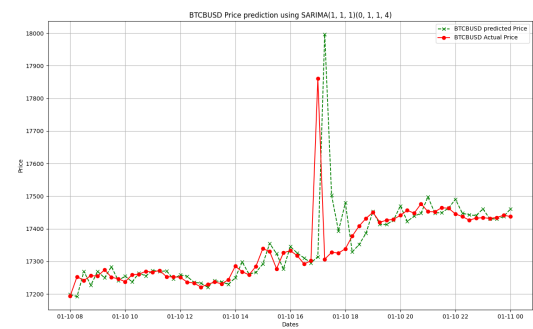
5.2. Prognozės

Šie rezultatai rodo ARIMA ir SARIMA modelių našumą prognozuojant trijų skirtingų simbolių - BTCBUSD, LTCBUSD ir XRPBUSD - kriptovaliutų kainas. Modelių veikimui įvertinti naudojami MSE (Mean Squared Error) ir MAPE (Mean Absolute Percentage Error). MSE matuoja vidutinį skirtumą tarp numatytų ir faktinių verčių kvadratu, o MAPE – vidutinį procentinį skirtumą tarp numatomų ir faktinių verčių. Abu šie rodikliai yra svarbūs vertinant modelių tikslumą.

Taikymo rezultatai trimis skirtingoms kriptovaliutų poroms: BTCBUSD, LTCBUSD ir XRPBUSD. Grafikuose matosi prognozuota ir tikra kaina laikotarpiu. BTCBUSD



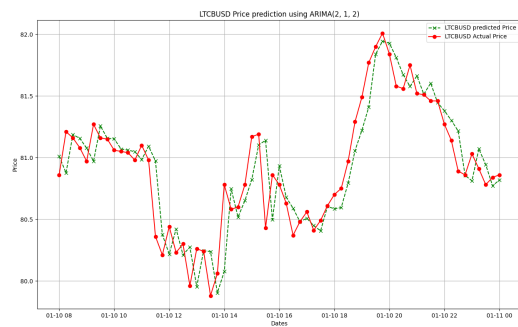
(a) ARIMA modelis



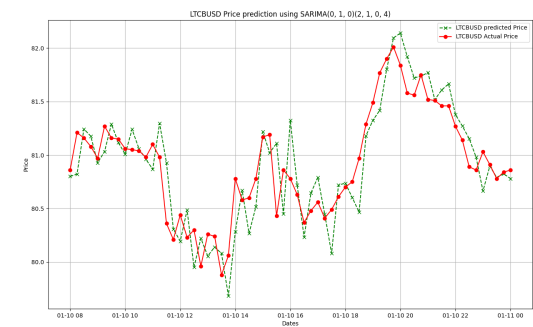
(b) SARIMA modelis

3 pav. BTCUSD prognozavimas

LTCBUSD



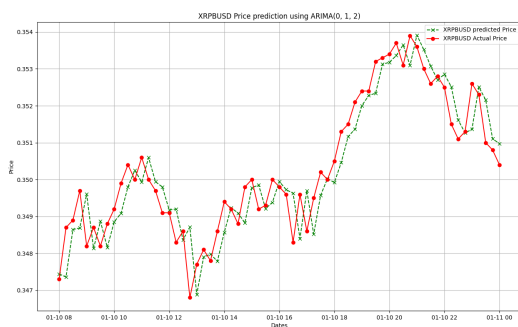
(a) ARIMA modelis



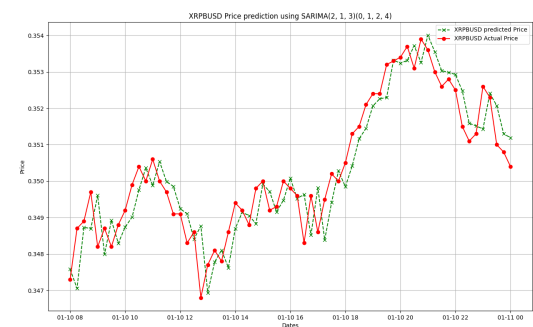
(b) SARIMA modelis

4 pav. LTCBUSD prognozavimas

XRPBUSD



(a) ARIMA modelis



(b) SARIMA modelis

5 pav. LTCBUSD prognozavimas

Symbol	Model	MSE	MAPE
BTCBUSD	ARIMA	13702.25	0.22%
BTCBUSD	SARIMA	13349.98	0.24%
LTCBUSD	ARIMA	0.05	0.2%
LTCBUSD	SARIMA	0.06	0.24%
XRPBUSD	ARIMA	4.92	0.16%
XRPBUSD	SARIMA	5.24	0.17%

2 lentelė. Modelių taikymo rezultatai

Iš lentelės matome, kad BTCBUSD poros MSE yra šiek tiek mažesnė SARIMA modeliui nei ARIMA modeliui, tai rodo, kad SARIMA modelis gali šiek tiek geriau atitikti šį duomenų rinkinį, bet MAPE yra šiek tiek mažesnė ARIMA modeliui.

LTCBUSD ir XRPBUSD poroje abiejų modelių MSE ir MAPE yra panašiai žemi, o tai rodo, kad abiejų modelių šių duomenų rinkinių našumas yra panašus. Visų porų MSE ir MAPE yra palyginti žemos, o tai rodo, kad modeliai gerai tinka duomenims, o prognozės yra gana tikslios.

Verta paminėti, kad šie rezultatai yra naudojami konkrečiams duomenų rinkiniui, taigi modeliai gali duoti skirtingus rezultatus kitiems rinkiniams.

6. Rezultatai ir išvados

Atlikta mokslinė analizė, literatūros apžvalga, kurios dėka pavyko aiškiau suprasti statistinius modelius, kriptovaliutų rinką, bei naudojamus įrankius.

- Susipažinta su darbine aplinka, bei reikalingomis bibliotekomis ir API,
- Palyginti autoregresiniai modeliai,
- Kadangi kainos nėra stacionarus duomenys negalima naudoti su paprastesniais AR, ARMA modeliais,
- Šie modeliai gali būti naudojami prognozuoti kriptovaliutų kainas,
- Kadangi rezultatai tiek ARIMA tiek SARIMA modelio panašūs, dėl paprastumo vertėtų naudoti ARIMA modelį.
- Padėti pamatai roboto lošėjui kūrimui, kuris implementuos konkretų planą ar strategiją,

Neišvengiama, jog tokiam trumpame darbe būtų aptartos visos įmanomos sritys ir atvejai, taigi tolimesnei tyrimo eigai pasiūlymas nagrinėti modelius, kurie naudoja dirbtinį intelektą ar dar kitus modelius (VAR, GARCH), suprasti jų veikimą ir palyginti su standartiniais statistiniais prognozavimo būdais aptartais šiame darbe. Šiame darbe apsistojama ties kainų tendencijų prognozavimu, tolimesnė programos galima eiga būtų realizuoti konkrečias strategijas robotui lošėjui.

Sąvokų apibrėžimai

API - Aplikacijų programavimo sąsaja (angl. Application programming interface), tai sistemos suteikiama sąsaja, kuria galima naudotis norint pasiekti tos sistemos funkcionalumą ar apsikeisti duomenimis.

Autoregresinis modelis - Statistinis modelis yra autoregresinis, jei jis numato būsimas reikšmes pagal praeities reikšmes. Pavyzdžiui, autoregresinis modelis gali siekti numatyti būsimas akcijų kainas, remiantis ankstesniais rezultatais.

Euristinis algoritmas - euristiniai algoritmai yra tokios intelektualios optimizavimo uždavinių sprendimo priemonės, kuriomis siekiama rasti aukštos kokybės (bet nebūtinai optimalius) sprendinius per priimtina laiką. Algoritmas negarantuoja, kad rastas sprendimas bus optimalus. [MBB09]

Kripto valiuta - Kripto valiuta yra skaitmeninė arba virtuali valiuta, kuri yra apsaugota kriptografija, todėl beveik neįmanoma jos padirbti ar išleisti dvigubai.

Kripto valiutų pora - Kripto valiutų pora rinkoje yra naudojama prekiaujant. Kripto valiutos yra susietos poromis, tad norint nusipirkti BTC valiutos pirmiausia reikia surasti galimus keitimo variantus, jei egzistuoja pora BTC/BUSD, galima nusipirkti BTC kripto valiutos už turimas BUSD valiutas. Dažnu atveju rinkoje pasidėjus ("FIAT") valiuta prekybos rinką ją konvertuoją į panašių token valiuta kaip BUSD(us -regulated stablecoin), USDT (usd tether) ar BNB (binance coin)

MSE - Vidutinė kvadratinė paklaida, angl. (Mean squared error)

MAPE - Vidutinė absoliučioji paklaida išreikšta procentais, angl. (Mean absolute percentage error)

Literatūra

- [AR17] Nashirah Abu Bakar ir Sofian Rosbi. Autoregressive integrated moving average (arima) model for forecasting cryptocurrency exchange rate in high volatility environment: a new insight of bitcoin transaction. *International Journal of Advanced Engineering Research and Science*, 4(11):130–137, 2017.
- [Aro21] Ran Aroussi. Yfinance, 2021. URL: <https://github.com/ranaroussi/yfinance> (tikrinta 2023-01-07).
- [CE20] David Carl ir Christian Ewerhart. Ethereum gas price statistics. *University of Zurich, Department of Economics, Working Paper*, (373), 2020.
- [Chi18] Wan Le Chi. Stock price forecasting based on time series analysis. *AIP Conference Proceedings*, tom. 1967 numeris 1, p. 040032. AIP Publishing LLC, 2018.
- [Coi] CoinMarketCap. Global cryptocurrency charts. Total cryptocurrency market cap. the graph shows the total market cap of all cryptoassets, including stablecoins and tokens. URL: <https://coinmarketcap.com/charts/> (tikrinta 2022-12-28).
- [Gar⁺18] Shruti Garg ir k.t. Autoregressive integrated moving average model based prediction of bitcoin close price. *2018 international conference on smart systems and inventive technology (ICSSIT)*, p.p. 473–478. IEEE, 2018.
- [Hua20] Yiqing Hua. Bitcoin price prediction using arima and lstm. *E3S Web of Conferences*, tom. 218, p. 01050. EDP Sciences, 2020.
- [Mac⁺14] Alec MacDonell ir k.t. Popping the bitcoin bubble: an application of log-periodic power law modeling to digital currency. *University of Notre Dame working paper*:1–33, 2014.
- [MBB09] Alfonsas Misevičius, Vytautas Bukšnaitis ir Jonas Blonskis. Euristinių algoritmų klasifikavimas. *Information & Media*, 48:117–126, 2009.
- [MH97] Spyros Makridakis ir Michele Hibon. Arma models and the box–jenkins methodology. *Journal of forecasting*, 16(3):147–163, 1997.
- [MSG14] Prapanna Mondal, Labani Shit ir Saptarsi Goswami. Study of effectiveness of time series modeling (arima) in forecasting stock prices. *International Journal of Computer Science, Engineering and Applications*, 4(2):13, 2014.
- [Nak08] Satoshi Nakamoto. Bitcoin: a peer-to-peer electronic cash system [white paper], 2008. URL: <https://bitcoin.org/bitcoin.pdf> (tikrinta 2022-12-28).
- [Nas06] Guy P Nason. Stationary and non-stationary time series. *Statistics in volcanology*, 60, 2006.
- [Smi⁺–] Taylor G. Smith ir k.t. pmdarima: arima estimators for Python, 2017–. URL: <http://www.alkaline-ml.com/pmdarima>.

- [SP10] Skipper Seabold ir Josef Perktold. Statsmodels: econometric and statistical modeling with python. *9th Python in Science Conference*, 2010. (Tikrinta 2023-01-12).
- [XL19] Jiahua Xu ir Benjamin Livshits. The anatomy of a cryptocurrency pump-and-dump scheme. *28th USENIX Security Symposium (USENIX Security 19)*, p.p. 1609–1625, 2019.