VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ STUDIJŲ PROGRAMA

Gilusis skatinamasis mokymas vertybinių popierių portfeliui optimizuoti

Deep Reinforcement Learning for Stocks Trade Automation

Magistro baigiamasis darbas

Atliko: Visvalas Stonkus (parašas)

Darbo vadovas: prof. dr. Aistis Raudys (parašas)

Recenzentas: asist. dr. Linas Petkevičius (parašas)

Vilnius – 2021

Santrauka

Šiame darbe nagrinėjami skatinamojo mokymo metodai ir jų tinkamumas naudoti vertybinių popierių prekybos automatizavimui. Darbe siekiama automatizuoti vertybinių popierių prekybos procesą siekiant didžiausios ilgalaikės investicinės grąžos su mažiausia rizika. Darbe analizuojami pagrindiniai skatinamojo mokymo principai, gilusis skatinamasis mokymas bei naujausios giliojo skatinamojo mokymo idėjos: DDPG, TD3 ir SAC giliojo skatinamojo mokymo algoritmai. Šios idėjos pritaikomos praktiškai vertybinių popierių prekybos uždaviniui optimizuojant vertybinių popierių portfelį susidedantį iš 30 Dow Jones indekso akcijų. Nustatyta, kad TD3 skatinamojo mokymo modelis geriau nei DDPG ar SAC modeliai tinka vertybinių popierių portfelio optizavimo uždaviniui.

Raktiniai žodžiai: gilusis skatinamasis mokymas, Markovo sprendimų procesai, vertybinių popierių prekybos automatizavimas, vertybinių popierių portfelio optimizavimas, dirbtinis intelektas

Summary

In this paper, we examine reinforment learning methods and their sutability for use in stock trading automation by maximizing long term investment return. This paper consists 5 main chapters. In chapter 1 we take an overview of related works in the field. In chapter 2 stock trading problem we are trying to solve is defined. In chapter 3, the key concepts of reinforment learning is analyzed. In chapter 4 we analyze the problems when applying reinforment learning in complex systems. We analyze the extremum variants of reinforcement learning and examine modern deep reinforcement learning algorithms: Deep Deterministic Policy Gradient (DDPG), Twin Delayed DDPG (TD3) and Soft Actor Critic (SAC). In chapter 5, we propose software program for analysing automated stock trading using deep reinformecent learning techniques. We train a deep reinforcement learning models to optimize a stock portfolio containing 30 Dow Jones stocks. For the performance metrics anualized return measured by the Sharpe ratio is used. Results shows the TD3 algorithm suits the task better than DDPG and SAC.

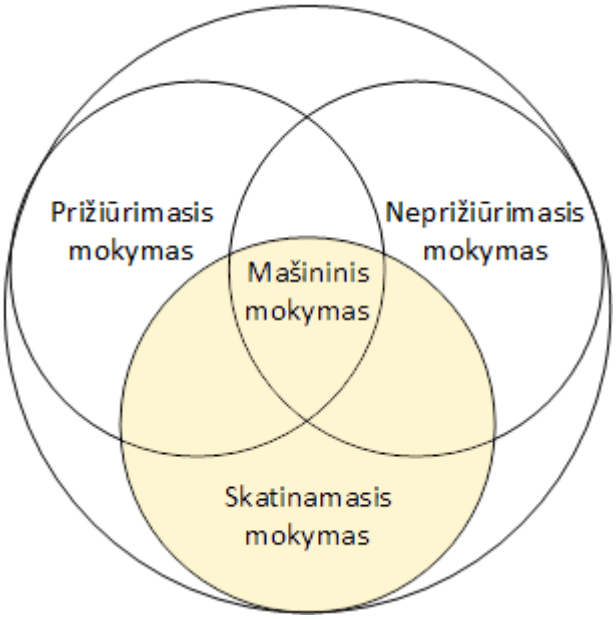
Keywords: deep reinforcement learning, Markov Decision Process, stock trading automation, portfolio allocation, artificial intelligence

TURINYS

Įvadas

Žmogus, kaip ir bet kuris kitas intelektą turintis subjektas geba mokytis iš savo praeities. Mūsų patirtys formuoja tai, kaip mes suvokiame aplinką. Kai naujagimis žaidžia, žvalgosi, klauso, skleidžia garsus, mojuoja rankomis ir kitaip sąveikauja su aplinka jis neturi informacijos kas yra gerai, o ko nereikėtų daryti, tačiau jis turi daugybę jutiminių organų leidžiančių jausti jį supančią aplinką bei motoriką, leidžiančią keisti aplinką. Jais naudojantis yra renkama informacija apie veiksmus ir jų pasekmes. Prilietęs karštą daiktą naujagimis kitą kartą bus atsargesnis. Šunų dresūroje šuo atlieka veiksmą, jeigu veiksmas atitinka žmogaus lūkesčius šuo gauna teigiamą grįžtamąjį ryšį ir taip padidina šio veiksmo pasikartojimą ateityje. Iš visų mašininio mokymo formų skatinamasis mokymas yra labiausiai atitinkantis mokymosi procesas, kaip tai daro žmonės bei gyvūnai ir daug svarbių skatinamojo mokymo algoritmų buvo įkvėpti biologinių mokymosi sistemų [SB18]. Mokymasis sąveikaujant su aplinka yra esminė idėja grindžianti beveik visas mokymosi ir intelekto teorijas [SB18]. Atliktas tyrimas [LCM+04] parodė, kad gyvūnai ieško optimalių sprendimų naudodami skatinamojo mokymo algoritmus.

Siekiu tobulinti dirbtinį intelektą mašininis mokymas užima svarbią vietą kompiuterių moksle. Kaip pateikta 1 paveikslėlyje mašininis mokymas yra vystomas trimis pagrindinėmis šakomis: prižiūrimas mokymas, neprižiūrimas mokymas ir skatinamasis mokymas.



1 pav. Mašininio mokymo šakos

Prižiūrimajame mokyme yra mokomasi iš pateikiamų pavyzdžių, kurie yra iš anksto pažymėti, taip nusakantys šiuos duomenis. Mašina yra apmokoma išanalizuojant pateiktų duomenų pavyzdžius, o surinkta informacija toliau naudojama klasifikavimo, regresijų ir kitiems uždaviniams spręsti. Šie metodai reikalauja didžiulio duomenų kiekio, kurie padengtų kiek įmanoma daugiau skirtingų atvejų, nes neradus atitinkamų pavyzdžių algoritmo našumas ženkliai mažėja, tuomet reikia pakartoti mokymą įtraukiant naujus duomenis.

Neprižiūrimasis mokymas sprendžia kitas problemas. Neprižiūrimojo mokymo metodai

didžiąją dalimi yra skirti nustatyti duomenų panašumus (skirtumus), todėl yra plačiai taikomi klasterizavimo, anomalijų atpažinimo ir kitas problemas spręsti.

Skatinamasis mokymas apibrėžia mokymosi sąveikaujant su aplinka idėją idealizuojant su mokymusi susijusius procesus ir matematiškai tai aprašant. Skatinamasis mokymas priešingai nei kitos dvi mašinų mokymo šakos nereikalauja pavyzdinių duomenų, bet nuolat fiksuodamas aplinką mokosi iš praeities rezultatų – atnaujina savo atminį ir todėl ateityje priima geresnius sprendimus. Skatinamojo mokymo metodai yra orientuoti į tikslą pasiekti kuo didesnį grįžtamąjį ryšį atliekant veiksmus, todėl gerai tinka sprendimo priėmimo uždaviniams spręsti. Šie metodai neretai yra naudojami automatizuojant žmogiškosios sąveikos reikalaujančius procesus.

Šiame darbe toliau nagrinėsime skatinamojo mokymo metodus ir juos taikysime automatizuojant vertybinių popierių prekybą.

Vertybinių popierių biržos prognozavimas yra sudėtinga užduotis, labiausiai dėl neaiškių faktorių darančių įtaką rinkos svyravimams. Daug faktorių lemia biržos pasikeitimus įskaitant politinius įvykius, bendras ekonomines sąlygas bei prekeivių viltis. Taigi nuspėti kainos judėjimą yra itin sunku. Visgi atsižvelgiant į akademinius tyrimus, vertybinių popierių kainų judėjimas nėra atsitiktinis [CG08].

Biržų analitikai naudoja techninę analizę [BLL92], kurią sudaro įvairūs rodikliai. Techniniai indikatoriai yra išvedami iš rinkos duomenų. Techninė analizė nagrinėja biržos svyravimus pasitelkiant praeities kainų įverčiais ir prekybos apimtimis, tai panaudojant bandant nuspėti ateities kainas. Techninė analizė teigia, kad kainų svyravimai juda tendencingai, o kriterijai, kurie paveikia kainų pasikeitimą pakeičia rinką per pamatuojamą laiko tarpą, bet ne iš karto.

Pagrindinis vertybinių popierių prekyboje taikomų mašininio mokymo metodų tikslas yra nuspėjant vertės pasikeitimus tinkamu laiku atlikti pirkimus ir pardavimus taip siekiant generuoti didžiausią grąžą ilguoju laikotarpiu.

Darbo tikslas. Šio darbo tikslas – ištirti giliojo skatinamojo mokymo metodų tinkamumą vertybinių popierių portfelio optimizavimo uždaviniui. Siekiant įgyvendinti užsibrėžtą tikslą, darbe išsikelti šie uždaviniai:

1. Ištirti skatinamojo mokymo metodus bei išrinkti tris modelius, kurie pagal atliktus tyrimus generuoja didžiausią investicijų grąžą vertybinių popierių prekybos uždaviniui;

2. Sukurti apsimokančią programų sistemą, vykdančią sprendimų priėmimą naudojant finansinius duomenis;

3. Realizuoti bei optimizuoti giliojo skatinamojo mokymo modelius;

4. Empiriniu tyrimu įvertinti algoritmų efektyvumą naudojant istorinius finansinius duomenis, palyginti prekybos rezultatus su baziniais modeliais, kitais moksliniais tyrimais bei pateikti rekomendacijas.

1. Susijusių darbų apžvalga

[DBK+16] pateikiami tyrimo rezultatai, kuriame palyginami skirtingi giliojo skatinamojo mokymo algoritmai. Apytikslio giliojo tiesioginio paskatinimo (angl. Fuzzy deep direct reinforcement arba FDDR) metodas pripažintas geriausiai veikiančiu algoritmu naudojant trumpo periodo rinkos duomenis. Tyrime taip pat atliekamas bandymas naudojant kelių rinkų duomenis ir yra vadinamas Multi–FDDR. Pastebima, kad Multi–FDDR metodas prasčiau veikia nei FDDR iki 2010 sausio mėnesio, tačiau naudojant naujesnius istorinius duomenis (nuo Sausio 2010) Multi–FDDR metodas veikia daug efektyviau. Galima priežastis yra ta, kad daugiau algoritminės prekybos kompanijų pradėjo veiklą po 2010 metų [DBK+16].

[ZZR20] tyrime vertybinių popierių prekybos uždaviniui buvo taikomi trys skirtingi giliojo skatinamojo mokymo algoritmai: DQN (angl. Deep Q-learning Network), strategijos gradiento (angl. Policy Gradient arba PG) ir A2C (angl. Advantage Actor-Critic). Tyrime buvo prekiaujama žaliavomis, akcijomis, obligacijomis ir valiutų rinkų vertybiniais popieriais. Tyrimas parodė giliojo skatinamojo mokymo pranašumą klasikinių prekybos strategijų (pirkti ir laikyti, Sign(R) [LZR19; MOP12] bei MACD signalo [BGH+15]) atžvilgiu visose turto klasėse išskyrus akcijų rinkas. Skatinamojo mokymo metodas nepasiekė geresnio našumo negu pasyvi strategija pirkti ir laikyti. Prekiaujant visomis turto klasėmis bendrai, pasiekta 4,4% geresnė nei lyginamojo indekso metinė investicinė grąža bei daugiau nei 22 kartus geresnis Šarpo rodiklis.

[LYC+20] tyrime akcijų prekybos uždaviniui buvo taikomi giliojo skatinamojo mokymo, aktoriaus ir kritiko principu paremti, modeliai DDPG ir TD3. Tyrimas rodo abiejų metodų pranašumą prieš lyginamąjį indeksą. Rezultatai rodo iki 17,61% metinę grąžą ir 1,03 Šarpo rodiklį, kai lyginamojo indekso metinė grąža siekia 10,61%, o Šarpo rodiklis 0,48. Svarbu tai, kad yra įtraukti 2020 metų akcijų rinkos nuosmukio rezultatai. Išanalizavus tyrimo autorių pateiktą programinį kodą matyti, kad prekyboje nėra įvertinami prekybos mokesčiai ar kiti su prekyba susiję kaštai.

[ZJS21] tyrime nagrinėjamas DDPG modelio našumas akcijų prekyboje. Prekyba vykdoma pasirenkant 8 skirtingas akcijas, iš kurių 4 mažo kintamumo ir 4 didelio kintamumo. Prekybos bandymas atliktas įtraukiant 2008 ir 2020 metų Covid-19 akcijų rinkos nuosmukius. Naudojant DDPG modelį gauta 14,12% investicinė metinė grąža ir 0,59 Šarpo rodiklis, kai pasyvi strategija, pirkti ir laikyti, tuo laikotarpiu generavo 4,37% metinę grąžą ir 0,27 Šarpo rodiklį.

[XLZ+18] tyrimo rezultate DDPG modelis sugeneravo 57% didesnę investicinę grąžą nei lyginamasis indeksas ir 40% geresnį Šarpo rodiklį.

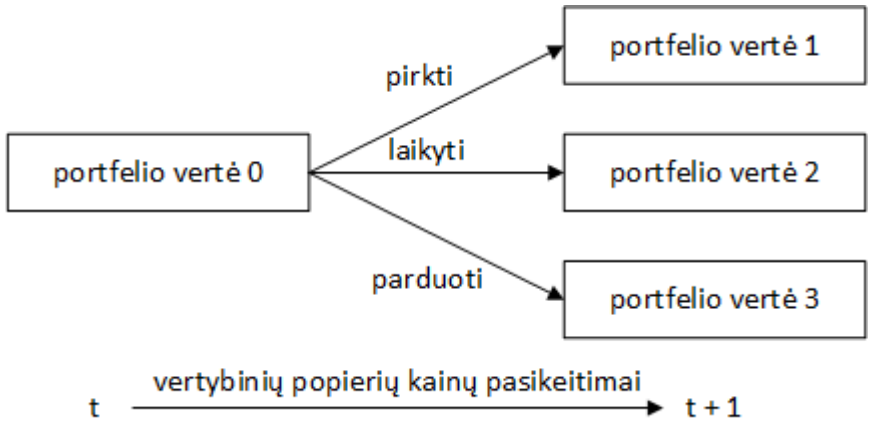
[Lee01; LYC+20; XLZ+18; ZZR20] tyrimuose vertybinių popierių prekybos problema suformuluota Markovo sprendimų procesu. Būsenų aibė įprastai sudaroma nurodant vertybinių popierių vertę, portfelio sudėtį ir laisvųjų lėšų balansą.

2. Vertybinių popierių prekybos uždavinys

Vertybiniai popieriai tai nematerialaus turto klasė. Šiame darbe nagrinėsime akcijų rūšies vertybinius popierius. Pagrindinė akcijų vertybinių popierių paskirtis yra verslui surinkti kapitalą, tuo tarpu investuotojams suteikiant teisę į įmonės turtą. Galime išskirti du pagrindinius finansinius siekius investuojant į akcijas: dividendai, kurie gali būti išmokami investuotojams nuo sukaupto pelno ir akcijų vertės augimas. Įmonių akcijų, kurios yra viešai platinamos, vertę nusako pasiūlos ir paklausos balansas. Akcijų prekyba laikysime akcijų vertybinių popierių pirkimą arba pardavimą. Vertybinių popierių prekyba yra taikoma siekiant turto vertės išlaikymo bei auginimo, rinkos likvidumo užtikrinimo ar kitų tikslų. Kadangi akcijų vertę nustato pasiūla bei paklausa, natūralu, kad sumažėjus pasiūlai ar padidėjus paklausai akcijų vertė didėja ir priešingai, didėjant pasiūlai ar mažėjant paklausai vertė mažėja. Investuojant į akcijas galima padidinti arba sumažinti turtą.

2.1. Vertybinių popierių portfelis

Vertybinių popierių portfeliu vadinsime vienos ar daugiau įmonių akcijų vertybinius popierius pinigus bei pinigus, skirtus pirkti vertybinius popierius. Portfelio vertę sudaro visų akcijų vertės ir grynųjų pinigų suma. Likvidžioje akcijų rinkoje pirkimo-pardavimo sandoriai, su tam tikromis išimtimis, vyksta nuolatos, todėl, nors ir nežymiai, akcijų vertė nuolatos keičiasi taip keičiant ir portfelio vertę. Taigi kaip parodyta 2 paveikslėlyje portfelio vertė kinta akcijas perkant, laikant ar parduodant. Perkant ir parduodant vertybinius popierius akcijų brokeriui yra mokami mokesčiai, kurie priklauso nuo biržos, kurioje tai daroma ir kitų kriterijų.



2 pav. Portfelio vertės kitimas
[XLZ+18]

Šiame tyrime vertybinių popierių prekyboje yra naudojami *Dow Jones Industrial Average (DJIA)* akcijų indeksą atitinkančių 30 įmonių akcijų vertybiniai popieriai. Pagrindinė indeksų paskirtis – suteikti investuotojams galimybę lengvai įvertinti ne atskirų akcijų bet visos rinkos, sektoriaus ar regiono būklę.

Dow Jones Industrial Average yra 1896 metais dviejų finansų analitikų *Charles Dow* ir *Edward Jones* įkurtas akcijų indeksas, tuo metu apėmęs 12 didžiausių Amerikos įmonių akcijas iš didžiau-

1 lentelė. *Dow Jones Industrial Average* indekso akcijos

Įmonės pavadinimas	Akcijos trumpinys (simbolis)	Akcijos svoris indekse
3M Company	MMM	3.84%
American Express	AXP	2.88%
Amgen	AMGN	4.87%
Apple Inc.	AAPL	2.57%
Boeing	BA	4.92%
Caterpillar Inc.	CAT	4.54%
Chevron Corporation	CVX	2.03%
Cisco Systems	CSCO	1.00%
The Coca-Cola Company	KO	1.04%
Dow Inc.	DOW	1.25%
Goldman Sachs	GS	6.54%
The Home Depot	HD	6.24%
Honeywell	HON	4.47%
IBM	IBM	2.59%
Intel	INTC	1.25%
Johnson & Johnson	JNJ	3.12%
JPMorgan Chase	JPM	2.95%
McDonald's	MCD	4.49%
Merck & Co.	MRK	1.49%
Microsoft	MSFT	4.98%
Nike	NKE	2.58%
Procter & Gamble	PG	2.64%
Salesforce	CRM	4.45%
The Travelers Companies	TRV	3.03%
UnitedHealth Group	UNH	7.33%
Verizon	VZ	1.13%
Visa Inc.	V	4.33%
Walgreens Boots Alliance	WBA	1.06%
Walmart	WMT	2.72%
The Walt Disney Company	DIS	3.66%

sių Amerikos pramonės verslo sektorių. Šiandien indeksas apima 30 Amerikos įmonių akcijas pateikiamas 1 lentelėje. Paprastai tariant indekso vertė apskaičiuojama išvedant tą indeksą sudarančių įmonių akcijų kainų aritmetinį vidurkį, tačiau, atsižvelgiant į akcijų padalijimus, įmonių apsijungimus ir kitus veiksnius, įmonių svoris tame indekse yra skirtingas.

2.2. Prekybos našumo vertinimas

Investuotojai priimdami sprendimus įprastai siekia dviejų dalykų: gauti didžiausią įmanomą grąžą prisiimant mažiausią įmanomą riziką neapibrėžtumui [Sha70].

Realią faktinę prekybos vertybiniais popieriais naudą galima įvertinti apskaičiavus metinę grąžą. Žinant metinę grąžą galima nesunkiai sužinoti pelną proporcingai nuo investicijos dydžio. Metinė grąža yra apskaičiuojama formule:

$$M = (1 + R_n)^{(1/n)} - 1 \tag{1}$$

kur R_n – portfelio investicinė grąža po n metų, o

$$R_n = \frac{P_n - P_0}{P_0} \tag{2}$$

kur P_0 – pradinė portfelio vertė, P_n – portfelio vertė po n metų.

Metinė grąža parodo prekybos rezultatą, tačiau tai nesuteikia informacijos apie tai kokios buvo priimtos rizikos. Vidutiniškai didesnę investicinę grąžą lemia didesnė rizika ir atvirkščiai. Rizika sukelia neužtikrintumą, todėl prisiimant didesnę riziką gautą didesnę investicinę grąžą bus sunkiau pakartoti, o rizikoms išsipildžius investuotojui gali tapti neracionalu nuvertėjusias akcijas parduoti - apribojami investuotojo pasirinkimai. Remiantis istoriniais duomenimis galima teigti, kad rizikingesnes investicijas reikėtų rinktis ilgesniam investavimo laikotarpiui darant prielaidą, kad prekybos realizavimo laiką galima lengviau keisti dėl rizikų, kurios gali nutikti.

Statistiškai didesnis vertybinių popierių portfelio vertės kintamumas indikuoja apie didesnę prekybos riziką ir atvirkščiai. Šarpo rodiklis yra skirtas įvertinti prekybos našumą atsižvelgiant į vertės kintamumą. Šarpo rodiklis yra apskaičiuojamas formule:

$$S = \frac{R_n}{\sigma} \tag{3}$$

kur σ – portfelio vertės standartinis nuokrypis.

2.3. Lyginamasis indeksas

Finansų rinkoms būdingas svyravimas, kuomet visos akcijų rinkos ar tam tikro sektoriaus vertė kažkurį laikotarpį didėja arba mažėja. Svyravimai būna įvairaus masto ir priklausomai nuo vidutinės vertės skaičiavimo apimties gali tęstis nuo sekundžių iki keleto metų, pastarieji, jeigu kintamumas neigiamas, vadinami ekonominėmis recesijomis. Prekybos bandymai atliekami naudojant istorinius duomenis, todėl galima pasirinkti bet kurį laiko intervalą, tačiau priklausomai nuo pasirinkto laiko intervalo gali smarkiai skirtis gauta metinė grąža ir Šarpo rodiklis dėl tam laikotarpiui būdingo teigiamo ar neigiamo ekonominio ciklo, todėl rezultatus reikėtų lyginti su lyginamojo indekso rezultatais.

Analizuojant istorines indeksų reikšmes ir pokyčius galima lengvai sužinoti, koks tikėtinas pasyviai valdomo akcijų portfelio pelningumas per metus. Toks pasyviai valdomas portfelis būtų sudarytas iš indekso įmonių akcijų, o kiekvienos akcijų pozicijos vertė portfelyje būtų proporcinga tos įmonės svoriui indekse [Mal03]. Tokią investavimo strategiją vadinsime "Pirkti ir laikyti".

Kadangi indeksas atitinka tas akcijas, kuriomis prekiaujama, toks indeksas yra tinkamas su juo lyginti prekybos rezultatus o tokio palyginimo rezultatas laikomas normalizuotu vertinimu. Taigi prekybos rezultatus vertiname santykinai nuo indekso. Santykinis prekybos rezultatų vertinimas parodo gebėjimą valdyti portfelį efektyviau negu pačios rinkos kitimas. Indeksas, su kuriuo lyginami prekybos rezultatai yra vadinamas lyginamuoju indeksu. Šiame darbe naudojamas *Dow Jones Industrial Average* lyginamasis indeksas.

2.4. Prielaidos ir kiti ribojimai

Šiame darbe daromos prielaidos:

- Akcijų kainos kitimas nėra atsitiktinis, o remiantis techninės analizės duomenimis galima bent dalinai numatyti akcijų vertę;
- Rinka, kurioje prekiaujama, yra likvidi, t.y. bet kuriuo metu galime parduoti visas turimas akcijas, taip pat galime pirkti akcijų jeigu pakanka lėšų;
- Mūsų vykdoma prekyba nedaro įtakos vertybinių popierių kainai;
- Prekyboje naudojami tik neatidėto pirkimo ar pardavimo sandoriai bei daroma prielaida, kad sandoris bus įvykdomas visada, o pirkimo ar pardavimo kaina yra paskutinė paskelbta uždarymo kaina. Nustatomas 0,1% sandorio mokestis, kuris numatytas apimti visus sandorio kaštus įskaitant kainos poslinkį nuo paskutinės uždarymo kainos.

Kiti ribojimai:

- Prekyboje galimi akcijų pirkimo ir pardavimo sandoriai jeigu turima tos rūšies akcijų. Neturimų akcijų pardavimas (angl. short selling) nėra numatytas;
- Įmonių mokamų dividendų išmokas vertinsime kaip atitinkamą akcijos kainos padidėjimą;
- Akcijų padalijimai įskaičiuojami į akcijos kainą, t.y. istorinė praeities akcijų kaina apskaičiuojama pagal dabartinę kainą padauginant iš akcijų padalijimo koeficiento.

3. Markovo sprendimų procesai

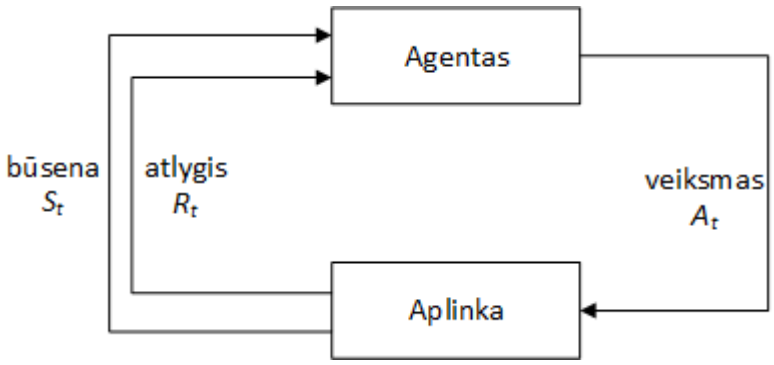
Šiame skyriuje aprašomi Markovo sprendimų procesai (angl. Markov decision processes arba MDP). MDP matematiniu pagrindu formuoja skatinamojo mokymo problemą ir jos teorinius aspektus. Šis procesas matematinėmis struktūromis apibrėžia grįžtamojo ryšio aspektą, vertės funkcijas bei Belmano lygtis, kurios formuoja esminius skatinamojo mokymo elementus. MDP sprendžia klasikinį sprendimų priėmimo modelį, kai veiksmai daro įtaką ne tik tiesioginiam grįžtamajam ryšiui, bet turi įtaką ir vėlesniems rezultatams. Dėl šios priežasties MDP kompromisu sprendžia tiesioginio ir atidėto grįžtamojo ryšio problemą. MDP principu sprendimų priėmime yra naudojamos vertės funkcijos leidžiančios parinkti tinkamiausius veiksmus kiekvienai būsenai.

MDP formuluotė yra skatinamojo mokymo paradigmos pagrindas. Tolesniuose poskyriuose apibrėžiami Markovo sprendimų procesuose naudojami komponentai.

3.1. Agento ir aplinkos sąveika

Markovo sprendimų priėmimo procesuose išreikštinai įtraukiami tik trys esminiai skatinamojo mokymo aspektai paprasčiausioje jų formoje: pojūčiai, veiksmai ir tikslas.

MDP procesuose turime sprendimų priėmėją, vadinamą agentu, kuris sąveikauja su aplinka. Agentas priima aplinkos būseną ir parenka veiksmą numatytoje veiksmų aibėje. Agentas gauna atnaujintą aplinkos būseną ir grįžtamąjį ryšį, kuris vadinamas atlygiu. Šį grįžtamąjį ryšį agentas naudoja siekdamas atnaujinti savo atmintį, kad ateityje priimami sprendimai sąlygotų kuo didesnį atlygį.



3 pav. Agento ir aplinkos sąveika pagal Markovo sprendimų procesus [SB18]

Markovo sprendimų procesuose agentas atlieka veiksmus A_t, iš griežtai apibrėžtos veiksmų aibės A. Kiekviena būsena S_t yra apibrėžta būsenų aibėje S ir kiekvienas atlygis R_t yra apibrėžtas atlygių aibėje R. MDP procesai apibrėžia atsitiktinio atlygio R_t ir atsitiktinės būsenos S_t priklausomybės tikimybę.

Tokios mašininio mokymo sistemos, kurios formuluojamos griežtai apibrėžiant veiksmų, būsenų ir atlygio priklausomybes bei jų tikimybes formuoja tai, ką vadiname Markovo sprendimų procesais. Baigtiniai Markovo sprendimų procesai (angl. finite Markov decision process) yra Markovo sprendimų procesai, kurių būsenų, veiksmų ir atlygių (S,A ir R) aibės turi baig-

tinius kiekius elementų. Didelė dalis dabartinės skatinamojo mokymo teorijos yra tinkama tik baigtiniams MDP procesams, tačiau metodai ir idėjos taikomos bendrai [SB18].

3.2. Tikslas ir atlygis

Skatinamojo mokymo paskirtis ir tikslas formuojamas atlygiu, kuris paduodamas agentui. Atlygis išreiškiamas realių skaičių aibėje. Agento tikslas yra gauti maksimalų atlygio kiekį ilguoju laikotarpiu. Tai reiškia, jog agento tikslas yra pasirinkti kelią, kuris sugeneruos nebūtinai didžiausią grąžą laiko momentu, tačiau bus pelningas ilguoju laikotarpiu. Tai galima aprašyti kaip atlygio hipotezę: Galutinis tikslas yra maksimizuoti sumą, kurią vadiname atlygiu.

Pavyzdžiui, įsivaizduokime robotą, kurį siekiama išmokyti vaikščioti. Tyrėjai po kiekvieno judesio numatė skirti atlygį atitinkamą roboto vietos padėties pajudėjimui į priekį. Robotas mokosi išeiti iš labirinto. Robotui skiriamas -1 taškas už kiekvieną atliktą judesį iki kol robotas išeina iš labirinto. Tokiu būdu robotas siekia kuo greičiau pasiekti finišą. Norint išmokyti robotą rinkti šiukšles robotui paėmus šiukšlę skiriamas vienas taškas, atlikus kitus judesius taškų neskiriama, tačiau robotui atsitrenkus skaičiuojamas -1 taško atlygis. Norėdami išmokyti robotą uždirbti prekiauti vertybinių popierių biržoje taškai skiriami proporcingai nuo pelno. Tokiu būdu robotas veikdamas savo galimų veiksmų aibe siekia atlikti veiksmus, kurie sąlygoja didžiausią pelną ilguoju laikotarpiu.

Mašina visuomet siekia gauti didžiausią atlygį, todėl turime atlygį skirti taip, kad jis tiesiogiai sąlygotų galutinį tikslą. Atlygio signalas yra būdas komunikuoti mašinai kas yra norima pasiekti, bet ne kaip tai norima pasiekti [SB18]. Bendru atveju atlygis neturėtų būti skiriamas už tai, kas mūsų nuomone yra sėkmingas veiksnys pasiekti tikslą. Pavyzdžiui, šachmatuose agentui turėtų būti skiriamas taškas tik už laimėjimą, bet ne už priešininko figūrų kirtimą.

3.3. Strategija ir vertės funkcijos

Vertės funkcija yra būdas pamatuoti naudą atlikti vieną ar kitą veiksmą duotuoju laiko momentu esant pagal duotąją būseną. Naudą galima išreikšti nustatant kokio atlygio galima tikėtis ateityje pasirinkus atlikti duotąjį veiksmą. Žinoma nauda ateityje priklauso ir nuo ateities veiksmų. Vertės funkcijos apibrėžiamos atsižvelgiant į konkretų elgesį priklausomai nuo situacijos, tai yra vadinama strategija.

Strategija yra išreiškiama kaip būsenų ir veiksmų rinkinys, kur kiekvienas veiksmas toje būsenoje turi numatytą tikimybę būti įvykdytas. Jeigu agentas naudoja strategiją π duotuoju laiko momentu t, tuomet $\pi(a|s)$ yra tikimybė, kad $A_t = a$ jeigu $S_t = s; a|s$ apibrėžia tikimybių pasiskirstymą veiksmui $a \in A(s)$ kiekvienam $s \in S$. Skatinamojo mokymo metodai apibrėžia kaip agento strategija keičiama pagal agento patirtis [SB18].

3.3.1. Būsenos vertės funkcija

Būsenos vertės funkcija strategijos π žymima V_π aprašo kiek gera kiekviena būsena naudojant strategiją π. Būsenos s vertė pagal strategiją π yra tikėtinas rezultatas pradedant būsenoje

s laiko momentu t ir ateityje naudojant strategiją π. Kitais žodžiais, tai parodo būsenos vertę strategijai π. Matematiškai $V_\pi(s)$, kaip aprašoma [SB18] galima išreikšti šia formule:

$$V_\pi(s) = E_\pi[G_t|S_t = s] = E_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s], \text{ visiems } s{\in}S \tag{4}$$

γ yra parametras, kuris nurodo, kiek turi būti diskontuojamos praeities vertės. Šis parametras leidžia seniau įvykusiems įvykiams teikti mažesnę svarbą nustatant vertę.

3.3.2. Veiksmo vertės funkcija

Panašiai, veiksmo vertės funkcija strategijos π žymima Q_π aprašo kiek gerai atlikti veiksmą a, kai būsena s naudojant strategiją π. Kitais žodžiais, tai parodo veiksmo vertę strategijai π. Matematiškai $Q_\pi(s,a)$ galima išreikšti šia formule:

$$Q_\pi(s,a) = E_\pi[G_t|S_t = s,A_t = a] = E_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s,A_t = a] \tag{5}$$

3.4. Optimalios strategijos paieška

Grubiai tariant, spręsti skatinamojo mokymo užduotį reiškia surasti strategiją, kuri pasiekia daug atlygio ilguoju laikotarpiu [SB18]. Baigtiniams Markovo sprendimų procesams galima tiksliai suskaičiuoti optimalią strategiją. Strategija π yra laikoma geresne, arba tokia pat kaip strategija π', jeigu tikėtina grąža strategijos π yra didesnė už π' visoms būsenoms. Išreiškus formule:

$$\pi{\geqslant}\pi' \text{tada ir tik tada, kai } V_\pi(s){\geqslant}V'_\pi(s), \text{ visiems } s{\in}S \tag{6}$$

$V_\pi(s)$ yra funkcija, kuri grąžina tikėtiną grąžą pradedant nuo pradinės būsenos s ir sekant šią strategija iki galutinės būsenos.

Optimali strategija, yra ta, kuri nėra prastesnė už bet kurią kitą strategiją [SB18].

3.4.1. Optimali būsenos vertės funkcija

Optimali strategija kiekvienos būsenos taške naudoja optimalią būsenos vertės funkciją. Optimalią būsenos vertės funkciją galima aprašyti formule:

$$V_*(s) = \max_\pi V_\pi(s), \text{ visiems } s{\in}S \tag{7}$$

V_* grąžina didžiausią tikėtiną grąžą, gaunamą naudojant bet kurią strategiją π, kiekvienai būsenai.

3.4.2. Optimali veiksmo vertės funkcija

Panašiai galima apibrėžti veiksmo vertės funkciją, dar vadinamą Q-funkcija:

$$Q_*(s,a) = \max_\pi Q_\pi(s,a), \text{ visiems } s{\in}S \text{ ir } a{\in}A \tag{8}$$

Kitais žodžiais Q_* grąžina didžiausią tikėtiną grąžą, gaunamą naudojant bet kurią strategiją π, kiekvienai būsenos ir veiksmo porai.

Kitas būdas, galėtų būti naudojamas parenkant veiksmą neatsižvelgiant į strategiją, bet renkantis geriausią tuo metu žinomą veiksmą toje pozicijoje. Toks metodas vadinamas gobšiu. Metodas parenka veiksmą, kuris turi didžiausią grąžos vidurkį.

$$Q_t(a) = \frac{\text{atlygių suma, kai buvo pasirinktas } a \text{ iki } t}{\text{kiekis atvejų, kai buvo pasirinktas } a \text{ iki } t} = \frac{\sum_{i=1}^{t-1} R_i \cdot 1A_i = a}{\sum_{i=1}^{t-1} 1A_i = a} \tag{9}$$

kur 1 yra funkcija lygi 1, kai sąlyga teisinga, ir 0, kai sąlyga neteisinga. Jeigu daliklis lygus nuliui $Q_t(a)$ apibrėžiame reikšmei 0 [SB18].

Paprasčiausia veiksmo parinkimo taisyklė yra pasirinkti veiksmą, kuris turi didžiausią vidutinę reikšmę – toks parinkimas vadinamas gobšus ir gali būti aprašomas formule:

$$A_t = \arg\max_a Q_t(a) \tag{10}$$

kur arg max_a yra funkcija, kuri išrenka a su didžiausia reikšme [SB18].

Gobšus pasirinkimas visada išnaudoja geriausią turimą informaciją siekiant gauti didžiausią atlygį, toks algoritmas neanalizuoja kitų atvejų, kurie galėtų būti geresni ilguoju laikotarpiu.

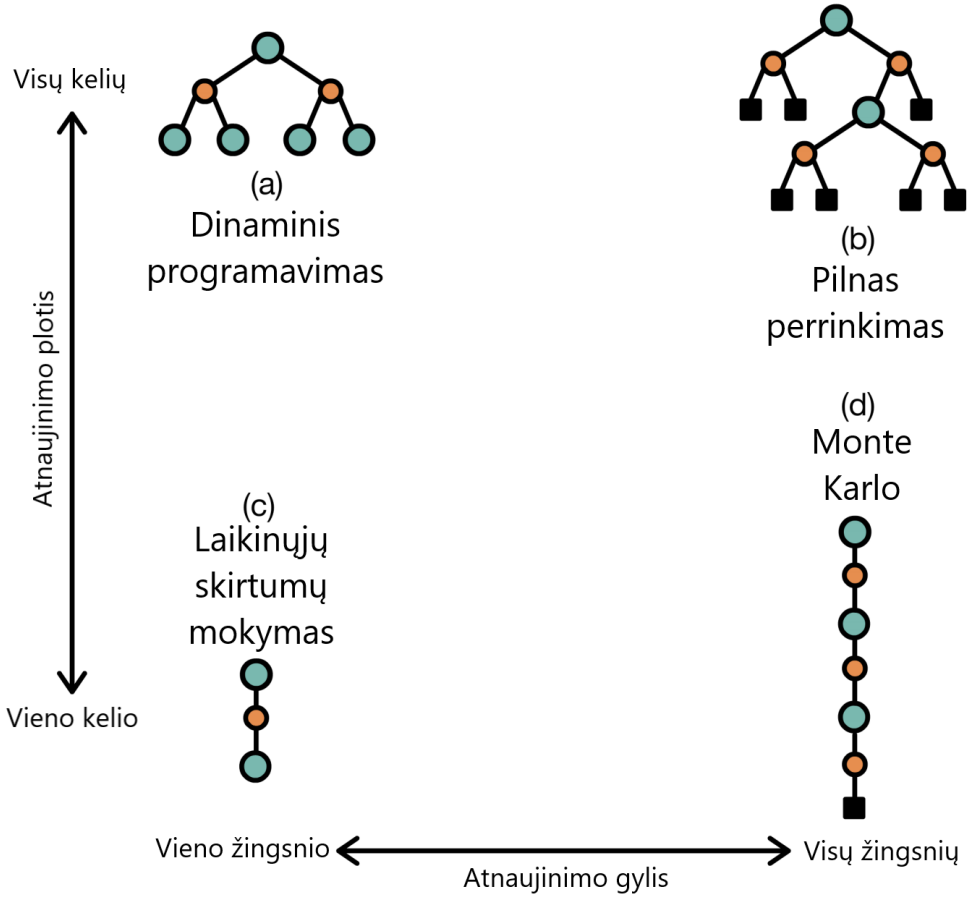
4. Skatinamojo mašininio mokymo metodai

Esminiai skatinamojo mokymo sprendimai yra paremti anksčiau minėtais Markovo sprendimų procesais. Visgi MDP apibrėžia idealizuotą teorinį modelį, tačiau realybėje tokį modelį pritaikyti neretai yra neįmanoma dėl sudėtingų užduoties sąlygų. Šiame skyriuje apibrėžiami pagrindiniai skatinamojo mokymo komponentai, problemos su kuriomis susiduriama juos realizuojant ir siūlomi sprendimo būdai.

4.1. Skatinamojo mokymo strategija

Skatinamojo mokymo strategija yra esminis skatinamojo mokymo komponentas. Strategija apsprendžia agento veiksmus ir kaupia informaciją apie patirtis. Šiame skyriuje apibrėžiamos pagrindinės skatinamojo mokymo idėjos strategijos realizacijai. 4 paveikslėlyje pateikta diagrama išskirianti dvi pagrindines dimensijas: strategijos atnaujinimo gylis, nusakantis kiek dažnai yra atnaujinama strategija, ir plotis, nusakantis kuri strategijos dalis yra atnaujinama.

Tyrimai rodo, kad geriausius rezultatus pasiekiama pasirenkant tarpinius, bet ne grynuosius šių metodų sprendimus.



4 pav. Skatinamojo mokymo metodų skirstymas išskiriant dvi svarbiausias dimensijas: atminties atnaujinimų gylis ir plotis

[SB18]

4.1.1. Dinaminis programavimas

Dinaminiu programavimu galima vadinti algoritmus, kurie sprendžia optimalios strategijos paieškos uždavinį turint tobulą, pilnos informacijos modelį, kaip Markovo sprendimų procesuose. Klasikinį dinaminio programavimo (angl. dynamic programming arba DP) pavyzdį galima apibrėžti kaip optimizuotą pilno perrinkimo algoritmą. Apskaičiavus reikšmes jos turi būti iš karto išsaugomos strategijoje, nes šios reikšmės naudojamos kitų veiksmų, kurie sudaro tą patį kelią, vertėms apskaičiuoti.

Uždavinys, sakykime optimalaus kelio radimas, skaičiuojant būsenos vertės funkciją, yra išreiškiamas padalijant jį į mažesnius uždavinius ir sprendžiant rekursyviai. Tai reikalauja didelio skaičiavimų kiekio bei pilnos informacijos apie aplinką, todėl neretai šis sprendimo būdas nėra tinkamas dėl sudėtingų aplinkos sąlygų. DP metodai nėra praktiški taikyti sudėtingoms problemoms dėl poreikio perrikti didelius kiekius galimų įvykių. Jeigu ignoruosime kelias technines detales DP metodai naudingiausio atvejo radimui pareikalaus polinominio dydžio perrinkimo pagal galimų būsenų ir galimų veiksmų kiekį [SB18], tačiau jei būsenų aibė yra labai didelė kiekvienas sprendimo priėmimas gali būti labai brangus. Pavyzdžiui „backgammon“ žaidime yra daugiau nei 10^{20} galimų būsenų. Net jeigu galėtume tikrinti reikšmes bei atnaujinti veiksmų svorius milijoną kartų per sekundę, užtruktų daugiau nei tūkstantį metų atlikti vieną ėjimą [SB18].

Visgi dinaminis programavimas aprašo esminį principą tokiems uždaviniams spręsti, todėl yra svarbus teorinis pagrindas. Dauguma kitų bandymų realizuoti skatinamojo mokymo algoritmus siekia to paties rezultato, tačiau siekiant sumažinti skaičiavimų kaštus bei sprendžiant kintamos, ne pilnos informacijos, aplinkos problemą.

4.1.2. Monte Karlo metodai

Monte Karlo metodai reikalauja tik patirties su aplinka – pavyzdinių sekų būsenų, veiksmų ir atlygių sukurtų sąveikaujant agentui su aplinka. Priešingai nei baigtiniuose MDP metoduose čia neturima pilnos informacijos apie aplinką, todėl nėra vedama visų galimų būsenų aibė.

Monte Karlo metodai sprendžia skatinamojo mokymo problemą grindžiant pavyzdiniais praeities rezultatais. Monte Karlo metodai naudojami tik epizodinėms užduotims. Sakykime, kad aplinka yra suskirstyta į epizodus ir visi epizodai nutinka nepriklausomai kokius veiksmus agentas atlieka. Epizodo pabaigoje istorija yra nuskaitoma į atmintį perskaičiuojant pavyzdinio epizodo veiksmų vertę atsižvelgiant į grįžtamąjį ryšį. Taip modelis yra atnaujinamas po kiekvieno epizodo. Sakysime, algoritmas veikia kažkurį laiko tarpą, o to laikotarpio pabaigoje atliekama retrospektyva, surenkant informaciją, kuri bus naudojama ateityje. Monte Karlo metodai dėl savo privalumų plačiai naudojami aplinkose, kurios yra ženkliai paveikiamos atsitiktinių įvykių.

4.1.3. Laikinųjų skirtumų mokymas

Visgi dėl poreikio fiksuoti pabaigos būseną grynieji Monte Karlo metodai netinka vertybinių popierių prekyboje. Laikinųjų skirtumų (angl. Temporal Difference arba TD) metoduose yra kombinuojamos Monte Karlo idėjos bei dinaminio programavimo praktikos. Laikinųjų skirtumų

metodai dažnai yra lyginami su DP bei Monte Karlo metodais. TD, taip kaip Monte Karlo metodai nereikalauja pilnos informacijos apie aplinką, t.y. nereikalingas griežtas aplinkos modelis, o algoritmas gali mokytis iš to kokia būsena yra paduodama, ši būsena gali būti dinaminė - būsenų aibė nebūtinai griežtai apibrėžta.

Jeigu Monte Karlo metoduose strategija atnaujinama tik po pilno epizodo įvertinus to epizodo grąžą, tai laikinųjų skirtumų metodo praktiką galima įsivaizduoti, kaip nuolatinį mokymąsi iš kiekvieno veiksmo įvertinant trumpalaikį poveikį. Grynuosiuose TD metoduose, panašiai kaip dinaminio programavimo atveju, strategija atnaujinama po kiekvieno žingsnio. TD metodai nereikalauja pabaigos fiksavimo ir gali būti panaudoti pastovių procesų automatizavimui, todėl, kaip rašoma [Lee01], TD metodai yra tinkami vertybinių popierių prekyboje. TD yra vieni plačiausiai naudojamų skatinamojo mokymo metodų šiomis dienomis. Vertybinių popierių prekybos uždaviniui daugiausia literatūros bei geriausi rezultatai aptinkami taikant TD metodus ir jų variacijas.

4.2. Aplinkos būsena

Panagrinėkime detaliau kas praktiškai yra skatinamojo mokymo aplinka ir kaip ji sudaroma. Kas tiksliai turi būti įtraukta, kad skatinamojo mokymo sistema galėtų generuoti siektinus rezultatus?

Dirbtinis intelektas gali dirbti tik skaitmeninėje aplinkoje, tačiau žmogus aplinka yra fizinė ir nors šios aplinkos yra susiejamos į skaitmeninę aplinką duomenys patenka tik išreikštinai. Tai kas vyksta natūralioje aplinkoje tik labai maža dalis informacijos yra įvedama į skaitmeninę aplinką.

4.2.1. Duomenų detalumo problema

Galime rasti būdų aprašyti aplinką. Fizikiniu požiūriu aplinką galima aprašyti apibūdinant fizikinius kūnus sudarančių dalelių būseną, tačiau toks detalus aprašas sudarytų didelį kiekį duomenų. Intelektas mokosi pagal pavyzdžius, o pagal pernelyg detalius aprašus yra sunku rasti pasikartojančius pavydžius, todėl duomenis reikalinga generalizuoti. Vertybinių popierių rinkos duomenų generalizavimo pavyzdys - finansinių indikatorių apskaičiavimas ir jų išrinkimas.

Viena svarbių intelekto savybių - gebėjimas generalizuoti duomenis. Žmogus pasinaudodamas sukauptomis žiniomis geba generalizuoti duomenis, taip sumažinant duomenų apimtį ir išryškinant svarbiausias, daugiausiai lemiančias, aplinkos savybes. Tokiu būdu palengvinamas mokymo uždavinys, kas leidžia turint ribotus intelekto resursus pasiekti geresnių mokymo rezultatų.

Mokymo efektyvumą gali apriboti per mažai duomenų arba per mažo detalumo duomenys. Intelektas, natūralus ar dirbtinis, negali įvertinti aplinkos reiškinių apie kuriuos duomenų neturi. Kuo labiau generalizuojama tai, kas vyksta, tuo labiau tikėtina, kad bus praleisti lemiantys reiškiniai. Generalizuojant duomenis ne tik prarandama dalis informacijos, tačiau generalizavimo išvesties rezultatas priklauso nuo generalizuojančiojo gebėjimo kokybiškai tai atlikti, taigi žmogaus klaidos generalizuojant duomenis turi tiesioginę įtaką mokymo rezultatams. Priklausomai nuo generalizuojančiojo padarytų prielaidų ir praleistos informacijos gali būti apribota galimybė pasiekti geresnių mokymo rezultatų.

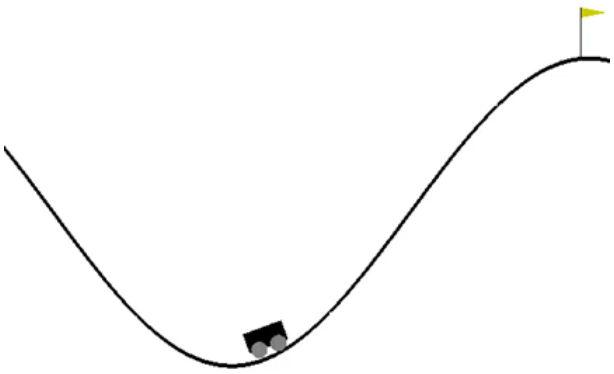
Mašininio mokymo duomenims ieškome duomenų, kurie mūsų nuomone yra turintys daugiausiai įtakos. Vertybinių popierių prekyboje sunku pasakyti kurie veiksniai turi daugiausiai įtakos kainos pasikeitimams, tačiau tyrimuose dažnai yra naudojami biržos duomenys, techniniai indikatoriai, retesniais atvejais ir kiti, su įmonėmis susiję duomenys, pavyzdžiui [WAM17] tyrime lyginama biržos duomenų naudojimo statistika, įtraukiami techniniai indikatoriai, Wikipedia naudojimo statistika, Google naujienos. Tyrime [LLD19] įtraukiami biržos duomenys, fundamentinės analizė duomenys, analizuojami ryšiai tarp įmonių bei naujienos.

Pastebime, kad iš 124 tyrimų, susijusių su vertybinių popierių prekyba, atliktų 2017 - 2019 metais, 35,83% yra naudojami tik biržos duomenys [Jia20]. Šiuose tyrimuose neįtraukiami techninės analizės rodikliai, kuriuos įprastai naudoja biržos analitikai. Tokie duomenys yra didelio detalumo, o tai, kad nėra pateikiama generalizuota informacija mašininiam mokymui tenka užduotis pačiam rasti lemiančius kriterijus.

4.3. Kredito priskyrimo problema

Prisiminkime šachmatų žaidimo pavyzdį: jeigu atlygis agentui skiriamas tik žaidimo pabaigoje, kaip parinkti tarpinius veiksmus? Dvidešimtame amžiuje vyravo du gana skirtingi požiūriai. Vienas jų buvo taikyti klaidų ir bandymų metodus, ir nenaudoti atlygio formavimo. Panaudoti kompiuteryje klaidų ir bandymų metodą buvo viena pirmųjų idėjų kaip realizuoti dirbtinį intelektą [SB18]. 1948 metais Alanas Tiuringas aprašė „Malonumo – skausmo sistemą“ (angl. pleasure – pain system), kuri veikė vadovaudamasi priežasties ir pasekmės principu: Kai konfigūracija prieina tašką kur kito veiksmo nusakyti negalima yra parenkamas atsitiktinis veiksmas. Kai yra gaunamas skausmo signalas visi iki to momento atlikti veiksmai yra atšaukiami ir jeigu gaunamas malonumo signalas visi iki to momento atlikti veiksmai yra išsaugomi.

Visgi toks metodas turi didelių trūkumų ypač naudojant sistemose, kurios reikalauja nuoseklumo. 5 paveikslėlyje pavaizduotas scenarijus: mašinėlė nėra pajėgi užvažiuoti į kalną iš rimties būsenos, tačiau įsisiūbavusi ir taip sukaupusi inercijos jėgos gali pasiekti tikslą. Įsivaizduokime, kad mašinėlę kontroliuoja skatinamojo mokymo algoritmas, kuris kas nustatytą trumpą laiką nusprendžia kontrolę važiuoti pirmyn arba atgal, o mašinėlei pasiekus tikslą skiriamas atlygis, o iki tol veiksmai parenkami atsitiktinai. Tikimybė, kad mašinėlė sėkmingai pasieks tikslą priklauso nuo to kiek dažnai bus atliekamas veiksmas bei nuo pačios trasos konfigūracijos bet galime pastebėti, kad rasti optimalų valdymo būdą bus išties sunku.



5 pav. Mašinėlė panaudoja inerciją užvažiavimui į kalną

1960 - 1980 klaidų ir bandymų metodo taikymo sumažėjo [SB18]. Didelę įtaką tam turėjo Minskio tiriamasis darbas [Min61], kuris aprašė mokymo klaidų ir bandymų metodu problemas, tokias kaip prognozavimas, lūkesčiai ir kaip jis pavadino paprasta kredito priskyrimo problema sudėtingoms skatinamojo mokymo sistemoms: Kaip reikia paskirstyti sėkmės atlygį, kai tam įtaką turėjo ne vienas, bet daug sprendimų? Metodas, kai numatomi tarpiniai sėkmės faktoriai ir už šiuos veiksmus skiriami taškai vadinamas atlygio formavimu (angl. reward shaping).

Vertybinių popierių prekyboje reikalingas nuoseklumas, nes nauda pasiekiama per ilgą laiką kai akcijų vertė kyla. Taip pat prisideda tai, kad sandoriai kainuoja, gaunami trumpalaikiai nuostoliai, o grąža atidėta. Laikinųjų skirtumų mokymas sprendžia nuoseklių veiksmų problemą, nes atlygis gali būti paskirtytas po kiekvieno veiksmo. [Lee01; LYC+20; XLZ+18; ZJS21] tyrimuose naudojami laikinųjų skirtumų mokymas, o atlygis nustatomas po kiekvieno veiksmo ir yra apskaičiuojamas pagal turimų akcijų pozicijų einamosios kainos skirtumą nuo prieš tai buvusios kainos. Kainos skirtumas tiesiogiai sąlygoja galutinį prekybos tikslą.

4.3.1. Suderinimo problema

Prisiminkime mašiną, kuri žaidžia šachmatais. Sakykime agentas gaunantis taškus už nukirstas figūras. Tikėtina, kad toks agentas išmoks nukirsti kuo daugiau figūrų, bet nebūtinai laimės žaidimą, t.y. agentas pasirinks kirsti figūras, net jeigu tai sąlygos žaidimo pralaimėjimą.

Atlygio formavimas reikalauja srities ekspertinių žinių. Panašiai kaip generalizuojant aplinkos duomenis, kadangi atsiranda žmogaus įtaka vertinanti procesą, bet ne jo rezultatą, didėja rizika, kad bus neįvertinti ir taip praleisti keliai, kurie efektyviau veda į tikslą. Kitaip tariant žmogus atlikdamas atlygio formavimą sufleruoja mašinai kaip reikia pasiekti tikslą, todėl mašinos veiksmų priėmimas tampa panašus į žmogaus, o to nepadarius mašina gali rasti labai efektyvius būdus pasiekti tikslą, apie kuriuos žmogus nė nepagalvojo.

[ZZR20] tyrime vertybinių popierių prekybai atlygis formuojamas formule pagal akcijų kainos skirtumą ir standartinę deviaciją siekiant apmokyti modelį taip, kad portfelio vertės svyravimas būtų mažesnis. Standartinė deviacija yra šalutinis prekybos tikslas, todėl tokiu metodu atsiranda rizika, jog strategija numatys optimizuoti mažesnę standartinę deviaciją vietoj pelno siekimo. [ZZR20] tyrime prekybos akcijomis rezultatas nesiekė pasyvaus investavimo rezultato.

4.4. Žvalgymas ir išnaudojimas

Gebėdami nustatyti optimalius veiksmus pagal būsenas galime tai pakartoti, tačiau kaip žinome, kad tai jau optimaliausias būdas pasiekti tikslą? Skatinamojo mokymo agentas mokosi per patirtis. Deja, mašinai diskrečiai yra apibrėžtas vienas tikslas. Panaudojus vertės nustatymo algoritmus mašina tiksliai apskaičiuoja optimaliausią žinomą veiksmą, pagal ankstesnę patirtį. Geriausio žinomo kelio naudojimą galima apibrėžti kaip tos situacijos ir informacijos išnaudojimą.

Siekiant rasti naujų kelių agentas privalo pasirinkti ne optimaliausią žinomą sprendimą, bet atlikti žvalgymąsi. Dilema ta, kad nei atliekant žvalgymąsi nei išnaudojimą negalima garantuoti geriausio rezultato. Agentas privalo bandyti įvairius veiksmus ir atsirinkti naudingiausius. Užduotims, kurios generuoja didelį atsitiktinių įvykių kiekį agentas privalo atlikti daugybę bandymų, iki kol gali nustatyti patikimą tikėtiną rezultatą.

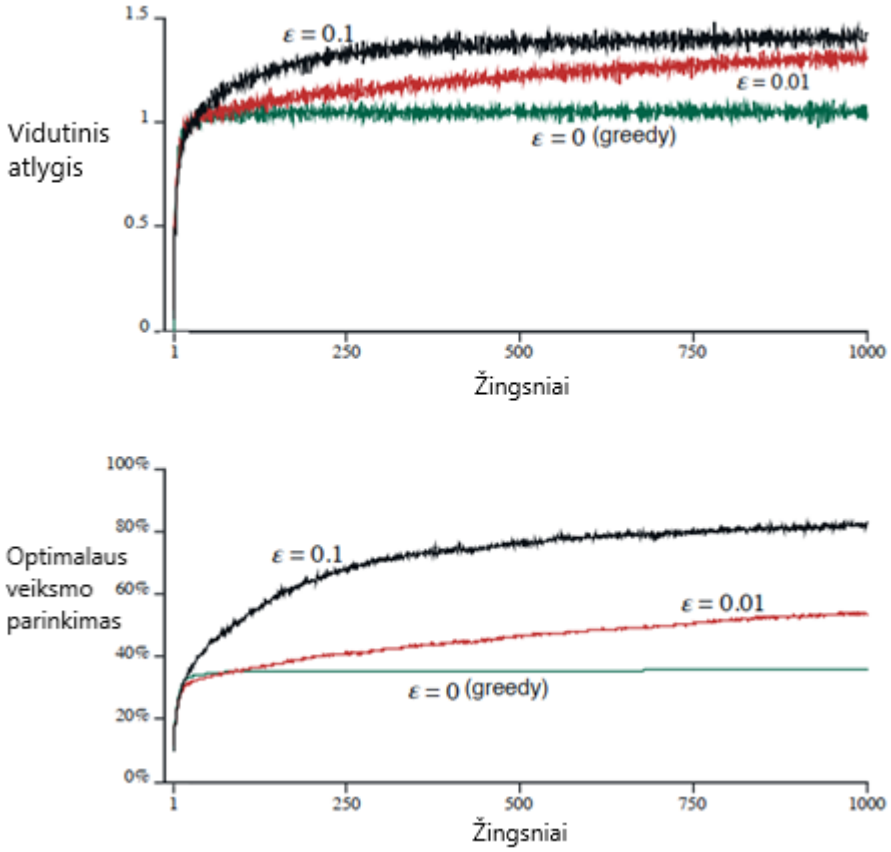
Daugybę metų matematikai bando išspręsti žvalgymosi ir išnaudojimo problemą, deja, dar niekam nepavyko. Verta paminėti, jog tiek prižiūrimajame tiek neprižiūrimajame mokyme šios problemos nepasireiškia, bent jau gryniausiose jų formose [SB18].

Yra daug būdų kaip bandoma spręsti šią problemą – ieškant balansą žvalgymosi ir išnaudojimo. Dauguma skatinamojo mokymo algoritmų parametrizuoja žvalgymosi ir išnaudojimo poveikį. Aplikacijoms, kurių aplinkos tendencijos keičiasi sparčiai, siekiant prisitaikyti prie besikeičiančios aplinkos reikalingas didesnis žvalgymasis.

4.4.1. Optimalaus veiksmo trukdymas

Algoritmas, kuris visuomet renkasi geriausius žinomus sprendimus yra vadinamas absoliučiai gobšiu. Toks metodas neleidžia atrasti optimalių sprendimų, todėl kai kuriais atvejais turime pasirinkti žvalgymąsi.

Yra daugybė būdų valdyti skatinamojo mokymo modelio žvalgymąsi. Vienas iš klasikinių sprendimų yra ϵ-gobši strategija. Sakykime norime didžiąją laiko dalį pasirinkti geriausius žinomus sprendimus, tačiau kartais pasirinkti naujus sprendimus, kad atrastume naujų kelių. Įsiveskime dydį ϵ, kur ϵ priklauso realių skaičių aibei ir $0<\epsilon<1$. Prieš pasirenkant sprendimą apskaičiuojame optimaliausią sprendimą, tačiau tik su tikimybe $1-\epsilon$ pasirinksime optimalų sprendimą. Su tikimybe ϵ pasirinksime bet kurį iš kitų galimų sprendimų. Dabar galime keisti dydį ϵ ir taip pakeisti žvalgybos poveikio tikimybę. Kai $\epsilon = 0$, parenkamas statistiškai optimaliausias žinomas sprendimas, $\epsilon = 1$, tuomet visada bus parenkamas bet kuris sprendimas, tačiau nustatę $\epsilon = 0{,}1$ turėsime veikimą, kuomet su 90 % tikimybe bus parenkamas geriausias žinomas sprendimas, bet 10 % tikimybė parinkti bet kurį sprendimą. Taip apibrėžiame ϵ-gobšią (angl. ϵ-greedy) strategiją. Šiuo dydį galima keisti norint paspartinti sistemos apsimokymo galimybes, pavyzdžiui, taikyti didelį dydį ϵ pirmosiose iteracijose ir nuosekliai jį mažinti naujoms iteracijoms. 6 Paveikslėlyje parodyti [SB18] atlikto tyrimo rezultatai taikant skirtingus dydžius ϵ, parodantys kaip vidutiniškai kinta algoritmo našumas.



6 pav. Vidutinis našumas naudojant ϵ-gobšus strategiją ir taikant skirtingus dydžius ϵ. Bandymai atlikti naudojant sistemą, kur veiksmų aibę sudaro 10 veiksmų. Duomenys gauti įvykdžius apmokymą daugiau kaip 2000 kartų sprendžiant skirtingas problemas.

[SB18]

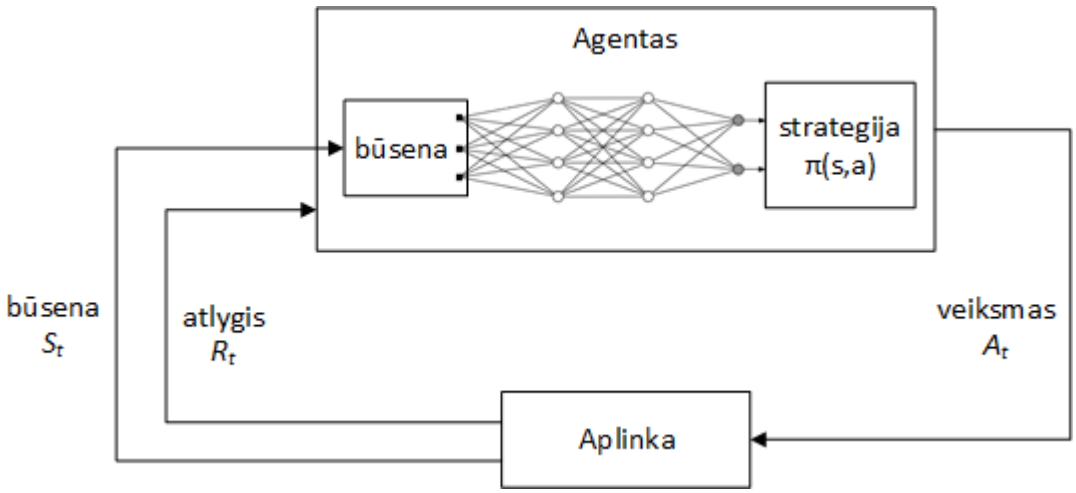
4.5. Gilusis skatinamasis mokymas

Ankstesniuose skyriuose apibrėžti metodai yra tiesiniai. Jie yra aiškūs teoriniame lygmenyje ir gerai veikia praktikoje, kai būsenos savybės suteikia domeno žinių. Būsenos savybių parinkimas yra vienas svarbiausių būdų suteikti domeno žinių skatinamojo mokymo sistemoms [SB18]. Būsenos savybes galima aprašyti polinomu, tačiau toks metodas ne visada efektyvus dėl prasto plečiamumo - reikalauja išreikštinio savybių apibrėžimo.

Tiesiniai metodai skaičiuoja absoliučias ankstesnių būsenų reikšmes, kas gali būti nepatogu kai būsenos yra išreiškiamos procentine dalimi gaunama daug būsenų, kurios visos užima skirtingą poziciją atmintyje ir nėra informacijos, kad jos yra panašios. Būsenos rezultatą galima suapvalinti taip sumažinant skirtingų būsenų kiekį, tačiau prarandamas tikslumas, o gretimos suapvalintos reikšmės išlieka vienodai skirtingos skatinamojo agento požiūriu. Pavyzdžiui, esant amplitudei 0–100 % suapvalinus būseną iki 1 % gaunama bent 100 galimų skirtingų būsenų, kurios visos agento požiūriu yra vienodai skirtingos.

Netiesiniuose metoduose, naudojami dirbtiniai neuronų tinklai siekiu suvesti didelio pasiskirstymo reikšmes į siauresnę būsenų aibę, kurią vėliau naudoja strategijoje priimant sprendimus. 7 paveikslėlyje pateikiama principinė tokių metodų schema. Netiesiniai metodai tapo labai po-

puliarūs ir yra vadinami giliuoju skatinamuoju mokymu (angl. deep reinforcement learning).



7 pav. Giliojo skatinamojo mokymo struktūrinė diagrama

Dirbtiniai neuroniniai tinklai skatinamojo mokymo sistemoje atlieka 4.2.1 skyriuje aprašytą įvesties duomenų generalizavimą. Apmokyti neuroniniai tinklai didelio detalumo duomenims suteikia prasmę.

Giliojo skatinamojo mokymo metodai taikant dirbtinių neuronų tinklus pradėti taikyti skatinamojo mokymo užduotyse leido sukurti sistemas, kurios priima didesnio detalumo duomenis. Giliojo skatinamojo mokymo proveržį lėmė DeepMind kompanijos sukurtas giliojo skatinamojo mokymo algoritmas [MKS+15], kuris parodė puikius žaidimo rezultatus įvairiuose kompiuteriniuose žaidimuose iki 25 kartų aplenkdamas profesionalius žaidėjus. Įdomu tai, kad būsena S_t pateikiama kompiuterinio žaidimo vaizdas taškų (angl. pixel) pavidalu, kas yra didelio detalumo duomenys lyginant su panašiais sprendimais, kuriems įvestis naudojamos skaitinės objektų koordinatės, atstumai, pagreitis ir t.t.

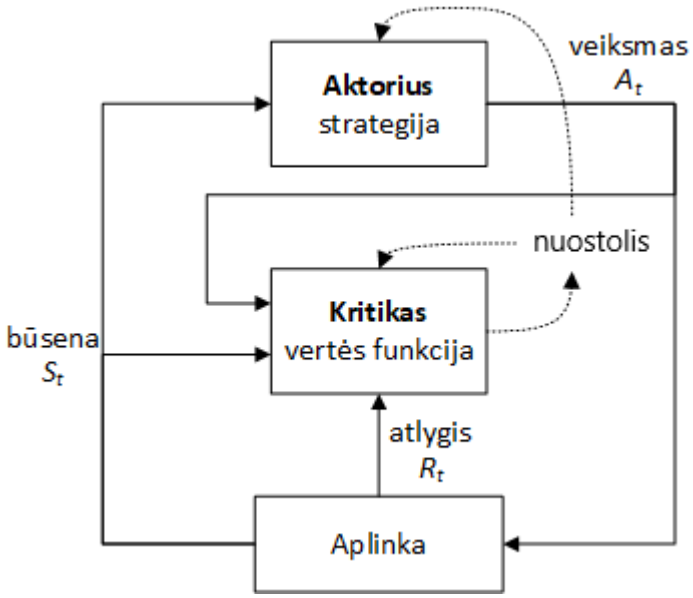
Gilusis skatinamasis mokymas sėkmingai taikomas vertybinių popierių prekybos uždaviniui, rodo tyrimai [LKK+19; XLZ+18; ZZR20]. [LKK+19] rezultatai rodo, kad šie metodai yra labiau tinkami už įprastus prižiūrimojo mokymo metodus vertybinių popierių prognozavimui. Giliojo skatinamojo mokymo metodai yra pranašesni realiose rinkos sąlygose ir siekiant priimti greitesnius ir geresnius prekybos sprendimus negu žmonės [Jia20].

Vertybinių popierių prekyboje būsenų aibė pastoviai kinta, todėl dauguma būsenų daugiau niekada nepasikartos ateityje. Tiesiniai metodai netinka vertybinių popierių prognozavimo problemai, nes ryšiai tarp įvesties ir išvesties vertybinių popierių biržoje yra labai netiesiniai [Lee01]. [Lee01] tyrime panaudotas TD metodas bei daugiasluoksnis neuroninis tinklas.

4.5.1. Aktoriaus ir kritiko metodas

[XLZ+18] taiko gilųjį deterministinį strategijos gradientą (angl. deep deterministic policy gradient arba DDPG) [LHP+15], kuris yra patobulinta deterministinio strategijos gradiento (angl. deterministic policy gradient arba DPG) algoritmo [SLH+14] versija. DPG apjungia Q-mokymo (angl. Q-learning) [SB18] ir strategijos gradiento (angl. policy gradient arba PG) idėjas [SB18].

DDPG skiriasi nuo DPG tuo, jog vertės funkciją vykdo agentas, vadinamas kritiku, o vertės funkcijos skaičiavimui naudojamas daugiasluoksnis neuroninis tinklas, vadinamas kritiko tinklu. 8 paveikslėlyje pateikta šio metodo architektūros diagrama.



8 pav. Aktoriaus – kritiko mokymo architektūros diagrama [GNC19]

4.5.2. DDPG modelis

DDPG – angl. *Deep Deterministic Policy Gradient* algoritmas aprašytas [LHP+15] mokymo metu keičia vertės funkciją ir strategiją. DDPG atveju vertės funkcijos vertė apskaičiuojama remiantis ankstesnėmis vertėmis, taip ženkliai sumažinant skaičiavimo laiką, o šį apskaičiavimą atliekanti funkcija vadinama *kritiku*. 8 paveikslėlyje pateikiama šio metodo principinė diagrama. 9 paveikslėlyje aprašytas šio metodo pseudokodas.

Algorithm 1 DDPG algorithm

Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ.
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer R
for episode = 1, M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial observation state s_1
  for t = 1, T **do**
    Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise
    Execute action a_t and observe reward r_t and observe new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in R
    Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R
    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
    Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
    Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_i}$$

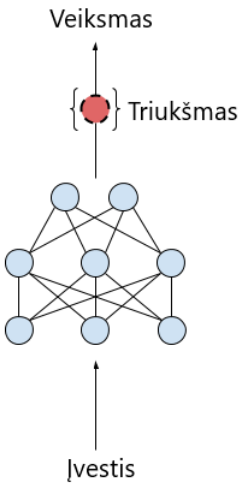
    Update the target networks:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$
$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

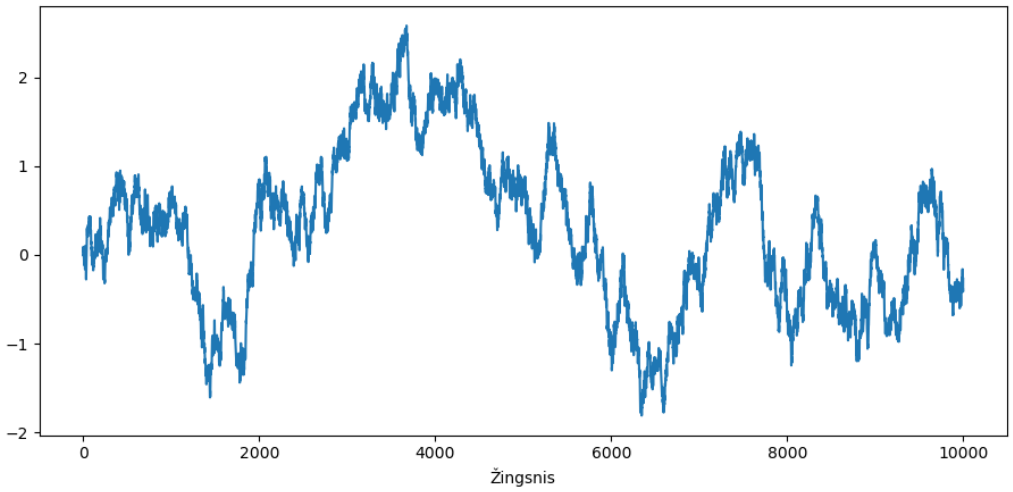
  end for
end for

9 pav. Aktoriaus – kritiko DDPG mokymo pseudokodas
[LHP+15]

DDPG strategija deterministiškai parenka optimalius veiksmus, todėl žvalgymo ir išnaudojimo problema sprendžiama pridedant triukšmą išvesties rezultatui kaip parodyta 10 paveikslėlyje. Algoritmo autorius rekomenduoja Ornstein–Uhlenbeck pasiskirtymu [UO30] orientuotą triukšmą, pavyzdys pateiktas 11 paveikslėlyje.



10 pav. Veiksmo triukšmas
[PHD+17]



11 pav. Ornstein Uhlenbeck veiksmo triukšmas

4.5.3. TD3 modelis

DDPG modelis kartais gali pasiekti puikių rezultatų, tačiau pastebima, kad pastarasis modelis labai trapus keičiant hiperparametrų ar kitus pakeitimus [Ach20]. Dažna to priežastis yra tai, kad vertės funkcija pervertina aktoriaus įvesties vertes, todėl strategija sugadinama, arba kitaip tariant įvyksta permokymas. Dvigubas atidėtasis DDPG (angl. *Twin Delayed DDPG*) arba kitaip dar vadinamas TD3 algoritmas sprendžia šią problemą trimis principiniais pakeitimais:

- Lygiagrečiai naudojamos dvi vertės funkcijos, taigi turime du kritikus dar vadinamus dvyniais. Iš šių naudojama yra vertė pagal mažesnę reikšmę.
- Politikos atnaujinimas yra atidėtas, t.y. strategija yra atnaujinama rečiau negu vertės funkcija. Algoritmo autorius [FVM18] rekomenduoja strategiją atnaujinti du kartus rečiau negu vertės funkciją.
- Vertės funkcijai paduodamos reikšmėms yra pridedamas triukšmas, taip suglotninant strategiją, išduodami veiksmai tampa pastovesni, nes sumažėja tikimybė pataikyti į smarkiai išsikreipusias vertes vertės funkcijoje.

12 paveikslėlyje pateikiamas TD3 algoritmo pseudokodas.

Algorithm 1 TD3

Initialize critic networks Q_{θ_1}, Q_{θ_2}, and actor network π_ϕ
with random parameters θ_1, θ_2, ϕ
Initialize target networks $\theta'_1 \leftarrow \theta_1$, $\theta'_2 \leftarrow \theta_2$, $\phi' \leftarrow \phi$
Initialize replay buffer $\mathcal{B}$
for $t = 1$ **to** T **do**
  Select action with exploration noise $a \sim \pi_\phi(s) + \epsilon$,
  $\epsilon \sim \mathcal{N}(0, \sigma)$ and observe reward r and new state s'
  Store transition tuple (s, a, r, s') in $\mathcal{B}$

  Sample mini-batch of N transitions (s, a, r, s') from $\mathcal{B}$
  $\tilde{a} \leftarrow \pi_{\phi'}(s') + \epsilon, \quad \epsilon \sim \text{clip}(\mathcal{N}(0, \tilde{\sigma}), -c, c)$
  $y \leftarrow r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', \tilde{a})$
  Update critics $\theta_i \leftarrow \text{argmin}_{\theta_i} N^{-1} \sum (y - Q_{\theta_i}(s, a))^2$
  if t mod d **then**
    Update ϕ by the deterministic policy gradient:
    $\nabla_\phi J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s, a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)$
    Update target networks:
    $\theta'_i \leftarrow \tau\theta_i + (1 - \tau)\theta'_i$
    $\phi' \leftarrow \tau\phi + (1 - \tau)\phi'$
  end if
end for

12 pav. TD3 pseudokodas
[FVM18]

4.5.4. SAC modelis

Švelnaus aktoriaus kritiko (angl. Soft Actor Critic arba SAC) modelis [HZA+18] taip pat paremtas aktoriaus - kritiko metodu. Esminis šio modelio skirtumas nuo jo pirmtako DDPG yra tas, jog jis reguliuoja mokymosi entropiją skatinant atsitiktinius įvykius. Jeigu strategijos svoriai yra koncentruoti ir didele tikimybe numato vienodą išvestį tai yra skaitoma maža politikos entropija, priešingai jeigu svoriai, taip pat išvestis galimų veiksmų aibėje yra tolygiai pasiskirstę tai skaitome didele entropija.

Agentas gauna papildomą paskatinimą priklausomai nuo strategijos entropijos. Tokiu metodu yra nusilpninamas lokalus maksimumas bei skatinamas žvalgymasis. SAC modeliui žvalgymasis reguliuojamas pasirenkant atlygio koeficientą α skiriamą už entropiją. Geriausias koeficientas (kuris stabiliausiai generuoja didžiausią atlygį) priklauso nuo aplinkos ir reikalauja detalaus reguliavimo [Ach20]. 13 paveikslėlyje pateikiamas SAC algoritmo pseudokodas.

Algorithm 1 Soft Actor-Critic

Input: initial policy parameters θ, Q-function parameters ϕ_1, ϕ_2, empty replay buffer $\mathcal{D}$
Set target parameters equal to main parameters $\phi_{\text{targ},1} \leftarrow \phi_1$, $\phi_{\text{targ},2} \leftarrow \phi_2$
repeat
  Observe state s and select action $a \sim \pi_\theta(\cdot|s)$
  Execute a in the environment
  Observe next state s', reward r, and done signal d to indicate whether s' is terminal
  Store (s, a, r, s', d) in replay buffer $\mathcal{D}$
  If s' is terminal, reset environment state.
  if it's time to update **then**
    for j in range(however many updates) **do**
      Randomly sample a batch of transitions, $B = \{(s, a, r, s', d)\}$ from $\mathcal{D}$
      Compute targets for the Q functions:

$$y(r, s', d) = r + \gamma(1 - d)\left(\min_{i=1,2} Q_{\phi_{\text{targ},i}}(s', \tilde{a}') - \alpha \log \pi_\theta(\tilde{a}'|s')\right), \quad \tilde{a}' \sim \pi_\theta(\cdot|s')$$

      Update Q-functions by one step of gradient descent using

$$\nabla_{\phi_i} \frac{1}{|B|} \sum_{(s,a,r,s',d) \in B} \left(Q_{\phi_i}(s, a) - y(r, s', d)\right)^2 \qquad \text{for } i = 1, 2$$

      Update policy by one step of gradient ascent using

$$\nabla_\theta \frac{1}{|B|} \sum_{s \in B} \Big(\min_{i=1,2} Q_{\phi_i}(s, \tilde{a}_\theta(s)) - \alpha \log \pi_\theta \left(\tilde{a}_\theta(s)| \, s\right) \Big),$$

      where $\tilde{a}_\theta(s)$ is a sample from $\pi_\theta(\cdot|s)$ which is differentiable wrt θ via the reparametrization trick.
      Update target networks with

$$\phi_{\text{targ},i} \leftarrow \rho \phi_{\text{targ},i} + (1 - \rho)\phi_i \qquad \text{for } i = 1, 2$$

    end for
  end if
until convergence

13 pav. SAC pseudokodas

[Ach20]

5. Vertybinių popierių prekybos automatizavimas naudojant skatinamuosius mašininio mokymo metodus

Šiame skyriuje atliekami bandymai taikant skatinamojo mašininio mokymo algoritmus. Mašininiam mokymui keliamas uždavinys – investicinių akcijų portfelio optimizavimas perkant, laikant bei parduodant vertybinius popierius siekiu maksimizuoti pelną ilguoju laikotarpiu. Ši užduotis formuluojama kaip dalinės informacijos Markovo sprendimų priėmimo procesų uždavinys.

Skatinamojo mokymo agentas sąveikauja su aplinka. Kiekviena sąveika tai atskiras žingsnis. Pirmiausia prieš atliekant žingsnį agentas gauna informaciją apie aplinkos būseną kaip numato mašininio mokymo aplinka: turimas portfelio pozicijas, akcijų kainą bei techninius indikatorius. Remiantis pateikta informacija agentas atlieka veiksmą.

Veiksmų aibė Markovo sprendimų procesuose sudaroma trijų galimų veiksmų aibėje: pirkti, laikyti ir parduoti. Kaip pateikiama 2 paveikslėlyje, taip pat kaip tyrime [XLZ+18], agentas atlikdamas šiuos veiksmus keičia portfelio vertę. Agentas pasirenka pirkti, parduoti arba laikyti pozicijas. Atliekant šiuos veiksmus portfelio vertė kinta. Laikant pozicijas portfelio vertė kinta dėl besikeičiančios vertybinių popierių kainos. Priklausomai nuo portfelio vertės agentas gauna paskatinimą. Taip formuojamas agento tikslas maksimizuoti portfelio vertę.

Kadangi vertybinių popierių biržoje didelė dalis veiksnių nėra žinoma, o techninė analizė negarantuoja rezultatų ateityje, laikykime vertybinių popierių prekybą ne pilnos informacijos aplinka, o kainos pokyčiai yra iš dalies atsitiktiniai įvykiai. Monte Karlo metodai sėkmingai panaudojami ieškant efektyvių kelių aplinkoje, kurioje gausu atsitiktinių įvykių. Toks metodas siekia maksimalios grąžos balansuodamas grąžą ir riziką.

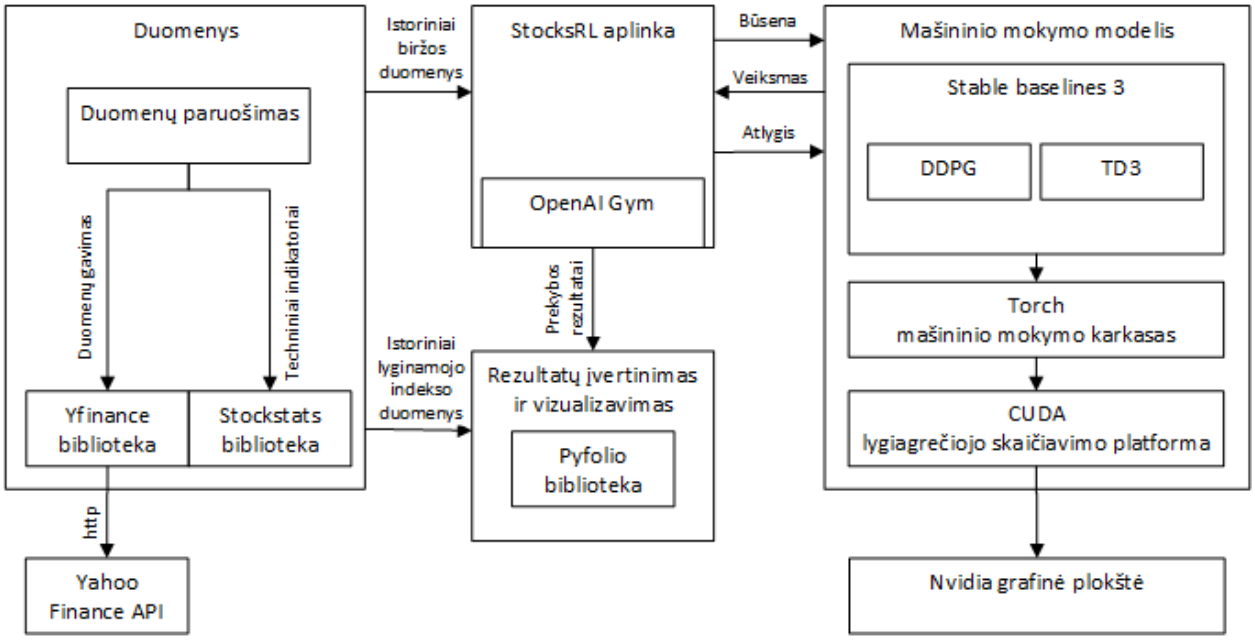
5.1. Naudotų technologijų aprašas

Metodų realizacijai naudojama *Python v3.6.12* programavimo kalba, skatinamojo mašininio mokymo modeliavimui – *Gym v0.15.3* [BCP+16], istoriniams vertybinių popierių duomenims gauti – *yfinance v0.1.55*, techniniams indikatoriams apskaičiuoti – *stockstats v0.3.2*, duomenų normalizavimui – *scikit-learn v0.21.0*, aritmetiniams veiksmams su masyvais – *pandas v1.1.4*, prekybos rezultatų statistikai apskaičiuoti ir grafikų kūrimui – *pyfolio v0.9.2*, bazinių strategijų prekybos testavimui – *Quantopian zipline v1.4.1*[Edd20] ir *PyPortfolioOpt v1.4.1*[Mar21], paveikslų generavimui – *Matplotlib v3.2.1*, optimalių hiperparametrų paieška vykdoma naudojantis *Optuna v2.7.0* biblioteka.

Mašininio mokymo algoritmų realizacija naudojama – *stable-baselines3 v0.10.0* [RHE+19], kuris atsirado kaip atvirojo kodo išsišakojimas nuo *OpenAI baselines* [DHK+17]. *stable-baselines3* yra paremtas *torch* mašininio karkaso pagrindu. Šiame darbe naudojamas *torch v1.7.0* mašininio mokymo karkasas. Skaičiavimams naudojama lygiagrečiojo skaičiavimo platforma – *CUDA v10.0* (angl. Compute Unified Device Architecture), kuri skaičiavimus atlieka naudojantis grafine plokšte. Šiame darbe tyrimas atliekamas naudojant Nvidia GeForce GTX 1080 grafinę plokštę.

Programa sukompiliuota *Windows 10 Pro build-19042* operacine sistema naudojant *Visual Studio Code v1.52.1* programinę įrangą.

14 paveikslėlyje pateikiama aušto lygio programinė architektūra atvaizduojanti pagrindinius programinius komponentus ir jų tarpusavio sąveiką. Pateikiami komponentai detaliau nagrinėjami vėlesniuose skyriuose.



14 pav. Programinė architektūra

5.2. StocksRL mašininio mokymo aplinka

Skatinamojo mašininio mokymo agentas apmokomas remiantis istoriniais vertybinių popierių biržos duomenimis simuliacinėje aplinkoje. Tam tikslui sukurta StocksRL mašininio mokymo aplinka paremta *OpenAI Gym* platforma [BCP+16].

5.2.1. Agento veiksmai

Agento pateikiamas veiksmas į mašininio mokymo aplinką apima visų turto vienetų procentinį pasiskirstymą ir taip inicijuoja portfelio perbalansavimą. Portfelio perbalansavimui yra įvertinamos turimos pozicijos ir apskaičiuojamas siektinas akcijų kiekis. Pozicijoms, kurių siektinas akcijų kiekis yra didesnis nei turima tos pozicijos akcijų, yra inicijuojamas akcijų pirkimas. Priešingai, pozicijoms, kurių siektinas akcijų kiekis yra mažesnis nei turima tos pozicijos akcijų yra inicijuojamas pilnas arba dalinis tos pozicijos akcijų pardavimas.

Atliekant akcijų pirkimą ir pardavimą yra daroma prielaida, kad rinka yra likvidi, pirkimo ir pardavimo sandoriai įvykdomi iš karto, o agento veiksmai nepaveikia rinkos kainos. Akcijų pirkimui ir pardavimui yra taikomas 0.1% mokestis, kaip ir [XLZ+18].

Agento veiksmų aibę sudaro 1 + n ilgio vektorius

$$\{a_0, a_1, a_2, ..., a_n | \sum_{i=0}^{n} a_i = 1; a_i \in [0,1]\} \tag{11}$$

kur a_0 – siektina grynųjų procentinė dalis portfelyje, a_1, a_2, ... a_n – siektina akcijų pozicijų

procentinė dalis portfelyje, n – akcijų pozicijų kiekis.

5.2.2. Aplinkos būsena

Skatinamojo mokymo aplinkos parametrai yra akcijų portfelio procentinis pasiskirstymas tarp visų turto vienetų ir prekiaujamų vertybinių popierių techniniai indikatoriai. Taigi aplinkos būsenos aibę sudaro $1 + n + n + n * m$ ilgio vektorius

$$\{a_0, a_1, a_2, ..., a_n, k_1, k_2, ..., k_n, x_{1,1}, x_{1,2}, ..., x_{1,n}, ..., x_{2,1}, x_{2,2}, ..., x_{2,n}, ..., x_{m,1}, x_{m,2}, ..., x_{m,n}\} \tag{12}$$

kur a_0 – grynųjų procentinė dalis portfelyje, a_1, a_2, ... a_n – akcijų pozicijų procentinė dalis portfelyje, n – akcijų pozicijų kiekis, k – vertybinių popierių vieneto kaina, x – techniniai indikatoriai, m – techninių indikatorių kiekis.

5.2.3. Atlygis

Atlygis apskaičiuojamas vertinant portfelio vertės pasikeitimą pagal formulę

$$R = P_t - P_{t-1} \tag{13}$$

kur P_t – einamojo žingsnio portfelio vertė, P_{t-1} – buvusio žingsnio portfelio vertė.

Portfelio vertė skaičiuojama sumuojant visų turimų turto vienetų vertę, akcijų vertė nustatoma remiantis paskutine uždarymo kainą pagal formulę

$$P_t = g_t + \sum_{i=1}^{n} v_{i,t} \cdot k_{i,t} \tag{14}$$

kur t – einamojo žingsnio numeris, g – grynieji, v – akcijų turimas vienetų kiekis, n – akcijų pozicijų kiekis, k – akcijos uždarymo kaina.

5.3. Naudotų duomenų aprašas

Šiame tyrime aplinkos aibė yra sudaroma naudojant istorinius vertybinių popierių biržos duomenis ir yra paremta finansinių rinkos duomenų technine analize.

Istoriniai vertybinių popierių biržos duomenys gauti naudojantis *yFinance* python biblioteka per *Yahoo Finance API (https://finance.yahoo.com/)*. Duomenų rinkinį sudaro intervalinės laiko eilutės.

Laiko eilutę sudaro 7 atributai:

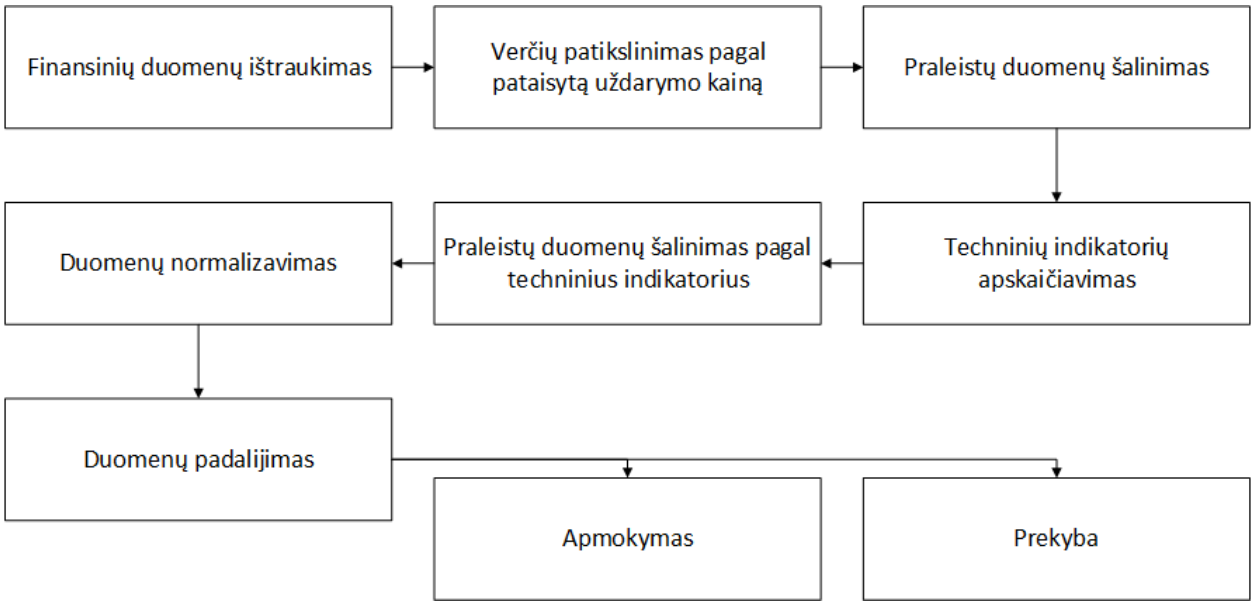
- intervalo pradžios laiko žyma;
- atidarymo kaina (angl. *open*) – vertybinių popierių vertė intervalo pradžioje;
- aukščiausia kaina (angl. *high*) – didžiausia vertybinių popierių vertė laiko intervale;

- žemiausia kaina (angl. *low*) – mažiausia vertybinių popierių vertė laiko intervale;
- uždarymo kaina (angl. *close*) – vertybinių popierių vertė intervalo pabaigoje;
- pataisyta uždarymo kaina (angl. *adjusted close*); – vertybinių popierių vertė intervalo pabaigoje atsižvelgiant į akcijų padalijimus ir dividendų išmokas;
- prekybos apimtis (angl. *volume*) – skaičius nurodantis kiek apsikeitimo sandorių įvykdyta biržoje per duotą laiko intervalą;

Šiame tyrime naudojami *Dow Jones Industrial Average* akcijų indeksą atitinkančių 30 įmonių akcijų pateikiamų 1 lentelėje vertybiniai popieriai. Rezultatų įvertinimui naudojamas DJIA akcijų lyginamasis indeksas. Duomenų aibę sudaro 22 metų finansiniai duomenys nuo 1999 sausio 1 dienos iki 2021 metų sausio 1 dienos.

5.4. Duomenų paruošimas

Pilnas duomenų paruošimas apima techninių indikatorių pridėjimą, duomenų valymą, normalizavimą ir kitus veiksmus. Pilnas duomenų apdorojimo kelias pavaizduotas 15 paveikslėlyje.



15 pav. Duomenų apdorojimo schema

Dėl įmonių mokamų dividendų bei akcijų padalijimų tikroji akcijų vertė gali kisti arba kisti jų kiekis. *Yahoo Finance* duomenyse šie du kriterijai jau yra įskaičiuoti pataisytoje uždarymo kainoje. Taigi akcijų vertei nustatyti naudosime pataisytą uždarymo kainą.

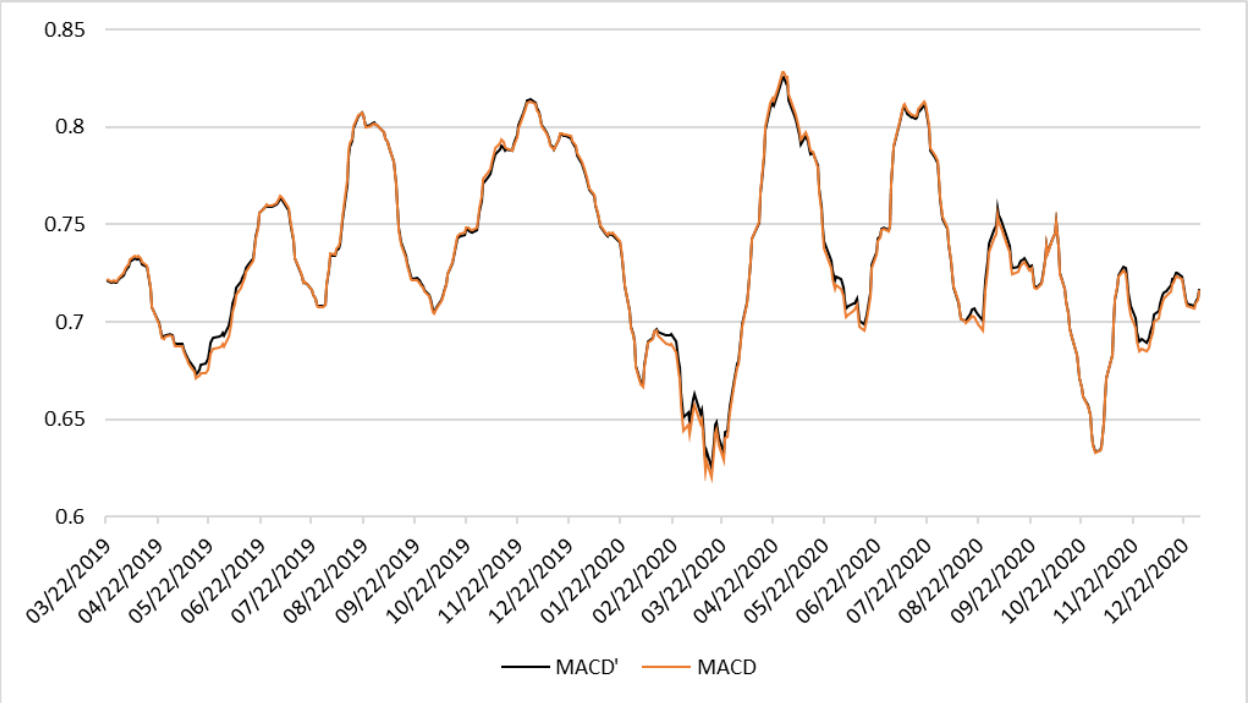
Atidarymo, aukščiausia, žemiausia ir uždarymo kainos yra pateikiamos originalios, kokios jos buvo tuo metu, t.y. neįskaičiuojant dividendų ar akcijų padalijimų, o dėl akcijų padalijimo kitą dieną akcijos nominali vertė gali būti kelis kartus mažesnė. Kadangi kai kurie techniniai indikatoriai yra apskaičiuojami pagal aukščiausią ar žemiausią kainą tai yra galimi techninių indikatorių

iškraipymai. Siekiant išvengti techninių indikatorių iškraipymo reikalinga, atidarymo, aukščiausios, žemiausios kainos bei prekybos apimties rodiklius perskaičiuoti atsižvelgiant į dividendus bei akcijų padalijimus. Perskaičiavimui naudojame formulę:

$$X'_t = X_t \cdot K_t / P_t \tag{15}$$

, kur K_t – uždarymo kaina laiko momentu t, P_t – pataisyta uždarymo kaina laiko momentu t, X_t – bet kuris kitas rodiklis laiko momentu t, X'_t – perskaičiuotas rodiklis.

16 paveikslėlyje pavaizduotoje diagramoje matomas skirtumas po pataisymo.



16 pav. AMGN akcijos MACD indikatorius prieš ir po pataisymo

5.4.1. Techniniai indikatoriai

Šiame darbe naudojami 5 techninės analizės indikatoriai:

- MACD (angl. Moving Average Convergence Divergence), kuris apskaičiuojamas pagal uždarymo kainą. Šis indikatorius parodo slenkantį vidurkį ir yra vienas dažniausiai naudojamų kainos pagreičio indikatorių [CNL14].
- RSI (angl. Relative Strength Index), kuris apskaičiuojamas pagal uždarymo kainą. Šis indikatorius skirtas parodyti trumpalaikį vertybinių popierių pervertinimą arba nuvertinimą nuo įprastos kainos [CNL14]. Vertės svyruoja griežtai apibrėžtame intervale nuo 0 iki 100.
- CCI (angl. Commodity Channel Index), kuris apskaičiuojamas pagal aukščiausią, žemiausią ir uždarymo kainas. Einamoji kaina yra palyginama su kainos vidurkiu einamajame laiko intervale [MPC+16].

2 lentelė. Duomenų statistinė analizė prieš normalizavimą

	Min	Q1	Vidurkis	Mediana	Q3	Max	Standartinis nuokrypis
Atidarymo kaina	0,200	22,463	53,376	37,447	67,702	435,564	47,988
Aukščiausia kaina	0,203	22,739	53,893	37,836	68,288	435,564	48,414
Žemiausia kaina	0,196	22,185	52,848	37,052	67,137	429,880	47,542
Uždarymo kaina	0,202	22,461	53,381	37,439	67,723	430,300	47,986
Prekybos apimtis	58575	$3{,}271{\cdot}10^6$	$2{,}480{\cdot}10^7$	$5{,}740{\cdot}10^6$	$1{,}138{\cdot}10^7$	$6{,}392{\cdot}10^9$	$9{,}939{\cdot}10^7$
Pataisytos kainos skirtumas	0,372	0,675	0,774	0,787	0,873	1,000	0,131
Kainos pokytis	-51,869	-0,791	0,055	0,043	0,893	34,755	1,948
ATR	0,006	0,497	1,125	0,803	1,300	23,289	1,220
MACD	-56,995	-0,228	0,151	0,114	0,525	18,665	1,301
RSI	0,000	46,936	52,471	52,429	58,022	100,000	8,074
CCI	-869,44	-65,921	18,968	32,510	104,123	666,866	112,577
ADX	3,211	19,843	32,276	29,047	42,161	100,000	15,811

- ADX (angl. Average Directional Index), kuris apskaičiuojamas pagal aukščiausią, žemiausią ir uždarymo kainas. Šis indikatorius parodo kainos judėjimo tendencijos stiprį ta pačia kryptimi [Gur+18]. Vertės svyruoja griežtai apibrėžtame intervale nuo 0 iki 100.
- ATR (angl. Average True Range), kuris apskaičiuojamas pagal aukščiausią, žemiausią ir uždarymo kainas. Šis indikatorius indikuoja kainos nepastovumą [Bru17].

5.4.2. Duomenų normalizavimas

Techninių indikatorių duomenys, kurie yra griežtai apibrėžtame intervale (RSI ir ADX) yra išreiškiami intervale nuo 0 iki 1. Kiti indikatoriai normalizuojami naudojantis *scikit-learn* įrankių paketo tiekiama funkcija *StandardScaler* [GM13]. Normalizavimo funkcija:

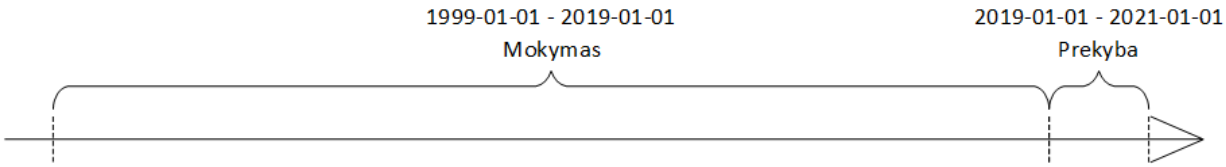
$$z = (x - u)/s \tag{16}$$

, kur x – duomuo, u – visų duomenų aritmetinis vidurkis, s – standartinis nuokrypis.

Kai kuriais atvejais dalies įmonių vertybinių popierių duomenys pagal tam tikras laiko žymas yra praleisti. Tokias laiko žymas turintys duomenys yra pašalinami iš duomenų aibės, t.y. tuose intervaluose akcijomis nėra prekiaujama.

5.4.3. Mokymo ir prekybos duomenys

Bendra duomenų aibė sudaro 143286 įrašų. Duomenų aibė D padalijama į du poaibius D_m ir D_v. D_m poaibis skirtas modelio apmokymui, o poaibis D_v - įvertinti apmokytą modelį. 17 paveikslėlyje pateikiama duomenų padalijimo schema.



17 pav. Duomenų padalijimas

5.5. Našumo vertinimo kriterijai

Prekybos siekiamybė yra gauti didžiausią investicinę grąžą prisiimant mažiausią riziką. Skatinamojo mokymo užduoties tikslas - rasti strategiją, kuri kiek galima geriau išpildytų šią siekiamybę. Šiame tyrime pagrindiniais prekybos našumo vertinamo kriterijais laikomas Šarpo rodiklis ir metinė grąža.

Rezultatus lyginame su pasyvia investavimo strategija [Mal03] atitinkamai lyginamąjį indeksą *Dow Jones Industrial Average*. Taip pat pateikiami palyginimai su kitomis bazinėmis investavimo strategijomis: OLMAR [LH12] ir mažiausio kintamumo (angl. Min Variance) [Ang12].

5.6. Hiperparametrai

Optimalių hiperparametrų paieška atliekama naudojantis biblioteka *Optuna*. Optimaliems hiperparametrams rasti 100 kartų vykdomas modelio apmokymas taikant skirtingas hiperparametrų reikšmes nustatytuose rėžiuose. Rėžiai parinkti pagal pradines rekomenduojamas reikšmes, o pradinės parametrų reikšmės nustatomos pagal reikšmes naudojamas [HIB+18; ZJS21; ZZR20].

DDPG ir TD3 metodams žvalgymui ir išnaudojimui valdyti naudojame veiksmo triukšmą. Ornstein–Uhlenbeck triukšmas, kur vidurkis lygus nuliui, o pasiskirstymo plotis σ lygus 0,1. SAC metodui veiksmo triukšmas nėra taikomas. Pradinis entropijos reguliavimo koeficientas α nustatomas 0,02, bet yra apmokomas. Naudotų hiperparametrų reikšmės pateikiamos 3 lentelėje.

Mokymas vykdomas iki pasiekiama 100000 žingsnių, o pirmus 1000 žingsnių agentas renkasi veiksmus atsitiktinai galimų veiksmų aibėje.

5.7. Modelio apmokymas

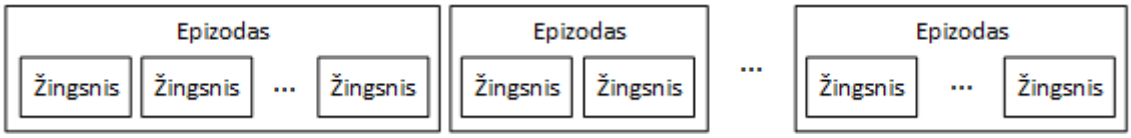
Skatinamojo mašininio mokymo agentas mokosi atlikdamas veiksmus ir gaudamas atlygį. Mokymas vykdomas iki tol, kol pasiekiamas nustatytas žingsnių kiekis. Mokymo sesiją sudaro epizodai. Epizodais yra nepertraukiamos agento darbo sesijos. Epizodą sudaro nuo 1 iki n kiekis žingsnių. Epizodo pradinėje būsenoje portfelio pasiskirstymas 100% pasiskirstęs grynaisiais, akcijų pozicijų dalis portfelyje lygi 0.

3 lentelė. Hiperparametrai

	DDPG	TD3	SAC
Mokymo žingsnis (angl. learning rate)	0.002	0.002	0.0005
Duomenų rinkinio dydis (angl. batch size)	128	128	128
Buferio dydis (angl. buffer size)	10000	10000	100000
τ	0.01	0.01	0.005
γ	0,99	0,99	0.99
Veiksmo triukšmas	OU (σ=0,1)	OU (σ=0,1)	-
α	-	-	0,02 (auto)
Strategijos atidėjimas (angl. policy delay)	-	2	-
Optimizatorius	Adam	Adam	Adam

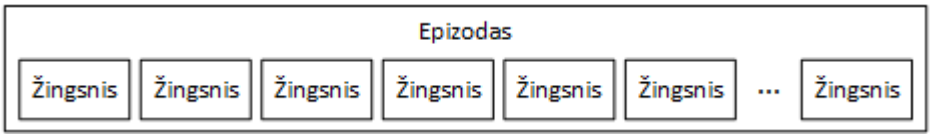
OU – Ornstein–Uhlenbeck triukšmas.

Epizodo vykdymo eigoje duomenų eilutės agentui pateikiamos iš eilės. Epizodas baigiamas išnaudojus visas tam epizodui skirtas duomenų eilutes. Pasibaigus epizodui būsenos reikšmė nustatoma į pradinę ir pradedamas naujas epizodas. Atliekant mokymą tos pačios duomenų eilutės gali būti pateikiamos kelis kartus, tačiau siekiant minimizuoti permokymo riziką epizodo pradžios ir pabaigos duomenų rėžiai parenkami atsitiktinai iš visos mokymui skirtų duomenų aibės.



18 pav. Mokymo epizodai

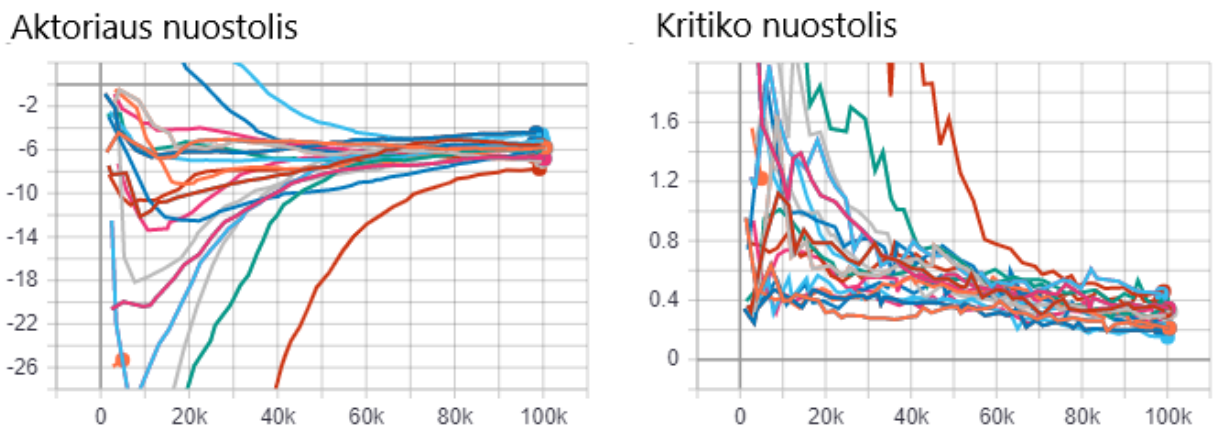
Apmokyto agento įvertinimui skiriamas vienas prekybos epizodas, kuriam skiriamos visos prekybai skirtos duomenų eilutės. Epizodo pradinėje būsenoje portfelio pasiskirstymas kaip ir mokymo epizoduose 100% pasiskirstęs grynaisiais, akcijų pozicijų dalis portfelyje lygi 0. Atliekant prekybą agentui paduodamos duomenų eilutės paduodamos tik vieną kartą. Taigi prekybai naudojami duomenys nėra priskiriami mokymo duomenų imčiai, tad prekybos epizodas parodo agento našumą su naujais duomenimis.



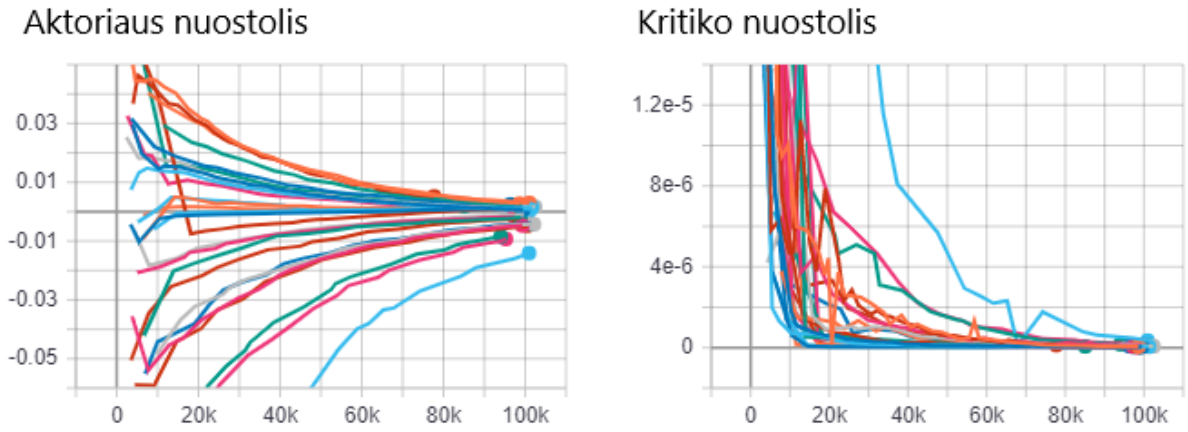
19 pav. Prekybos epizodas

Tyrime buvo apmokyti trys giliojo skatinamojo mokymo modeliai DDPG, TD3 ir SAC. 20 21 22 paveikslėliuose pateiktos *tensorboard* aplinkos sugeneruotos modelių mokymosi diagramos. Kiekviena kreivė vaizduoja vienos mokymo sesijos rezultatą. Apmokymui numatytas 100 tūkstančių žingsnių kiekis pateiktas horizontalioje ašyje. Nuostolio vertė grafikuose pateikta vertikalioje ašyje. Grafikuose matyti, kad dauguma atvejų aktoriaus ir kritiko nuostolis konverguoja, tačiau SAC modelio mokymo rezultatai nėra nuoseklūs. TD3 modelis bendru atveju konvergavo

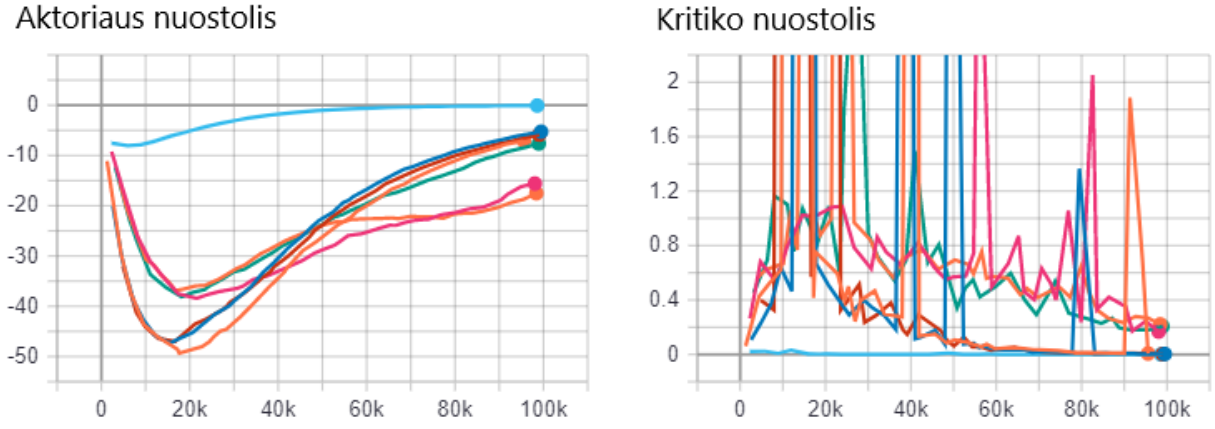
geriau negu DDPG arba SAC modeliai. TD3 modelis bandymų metu buvo mažiausiai jautrus hiperparametrų pokyčiams.



20 pav. DDPG modelio apmokymo nuostolis



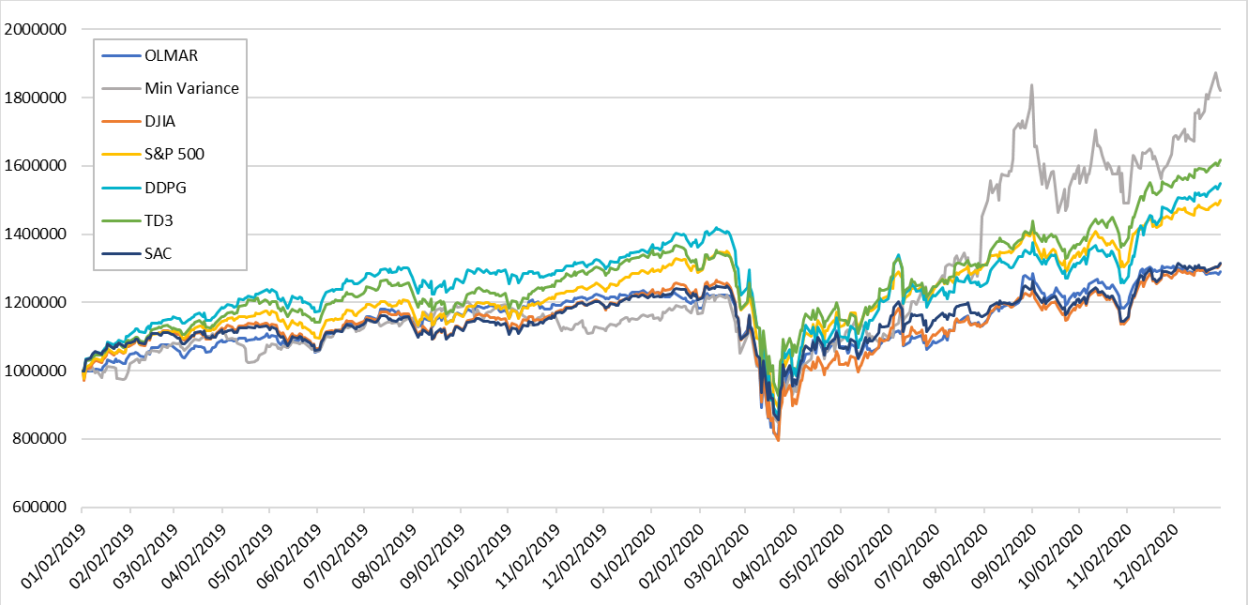
21 pav. TD3 modelio apmokymo nuostolis



22 pav. SAC modelio apmokymo nuostolis

5.7.1. Prekybos rezultatai

Apmokius skatinamojo mokymo modelius iš naujo inicijuojama skatinamojo mokymo aplinka ir pateikiant prekybai skirtus duomenis ir vykdomas vienas epizodas. Pradinis prekybos portfelio dydis 1000000 Jungtinių Valstijų dolerių. 23 paveikslėlyje pateikiamas grafikas iliustruojantis portfelio vertės kitimą. Papildomai, palyginimui įtrauktas geltona spalva pažymėtas *S&P 500* akcijų indeksas, sudarytas iš 500 didžiausių, viešai platinamų, Amerikos įmonių akcijų.



23 pav. Prekybos našumas

4 lentelėje pateikiami tyrimo rezultatai pateikiant trijų tirtų giliojo skatinamojo mokymo strategijų prekybos rezultatus, bazinių strategijų ir lyginamojo indekso *DJIA* rezultatus. Paveikslėliuose 24 - 29 pateikiami išsamūs DDPG, TD3 ir SAC modelių prekybos rezultatai. Paveikslėliuose 30 - 32 pateikiamos šių modelių atlygio, sugeneruoto prekybos metu, histogramos.

4 lentelė. Prekybos našumo palyginimas (2019-01-01 - 2021-01-01)

	DDPG	TD3	SAC	OLMAR	Min Variance	DJIA
Metinė grąža	24,64%	27,28%	14,80%	13,79%	35,46%	14,48%
Metinė grąža^{+}	70,16%	88,4%	2,2%	-4,77%	144,89%	-
Šarpo rodiklis	0,91	1,07	0,68	0,63	1,09	0,63
Šarpo rodiklis^{+}	44%	70%	7,9%	0%	1,73	-
Metinė standartinė deviacija	28,93%	25,67%	25,1%	26,03%	32,76%	27,43%
Didžiausias nuosmukis	38,85%	32,19%	31,38%	34,4%	30,63%	37,09%

Metinė grąža^{+} – metinės grąžos skirtumas nuo lyginamojo indekso;
Šarpo rodiklis^{+} – Šarpo rodiklio skirtumas nuo lyginamojo indekso.

Tyrimo rezultatai parodė, kad naudojant DDPG ir TD3 modelius pasiektas geresnis prekybos našumas negu pasyvi investavimo strategija. *Min Variance* strategijos rezultatai didžiąją dalį laiko atsiliko nuo kitų strategijų, tačiau nuo 2020 metų balandžio mėnesio, kai buvo didžiausias rinkos smukimas vertė laikėsi pagal vidurkį. Nuo 2020 birželio mėnesio šios strategijos portfelio

pasiskirstymas tapo 100% investuoti į AAPL kompanijos akcijas. Ši strategija remiasi geresniu Šarpo rodikliu, taigi portfelio perskirstymą lėmė nuoseklus AAPL akcijų kainos augimas iki 2020 birželio. Nuo to laiko matomas didelis portfelio vertės kintamumas: metinė standartinė deviacija nuo 29% pakilo iki 39%, didžiausias nuosmukis nuo -20% padidėjo iki -30%. Portfelio vertė tapo priklausoma nuo vienos akcijos kainos svyravimo, o rodikliai indikuoja padidėjusią prekybos riziką. Tuo tarpu TD3 modelio portfelio pasiskirstymas apėmė 11 įmonių akcijas.

[XLZ+18] tyrime, naudojant DDPG algoritmą buvo gauta 57% didesnė metinė grąža ir 40% geresnis Šarpo rodiklis nuo lyginamojo indekso. [ZJS21] tyrime naudojamas DDPG modelis pasiekė 14,12% metinę grąžą ir 0,6 Šarpo rodiklį, kai pasyvi strategija siekė 4,37% metinė grąžą ir 0,27 Šarpo rodiklį. [LYC+20] tyrime DDPG modelis pasiekė 49% geresnę metinė grąžą ir 104% geresnį Šarpo rodiklį negu lyginamasis indeksas. Šiame tyrime DDPG modelis pasiekė 70% didesnę metinę grąžą ir 44% geresnį Šarpo rodiklį negu lyginamasis indeksas.

TD3 prekybos rezultatas geriausias iš šiame tyrime naudotų skatinamojo mokymo metodų. Metinė grąža 88%, o Šarpo rodiklis 70% viršijo lyginamojo indekso rezultatą. Iki 2020 kovo mėnesio TD3 modelio metinė grąža buvo 18%, DDPG - 19%, Šarpo rodiklis atitinkamai 1,26 ir 1,33. TD3 modelio pranašumas pasirodė nuo 2020 metų kovo mėnesio, prasidėjus su Covid-19 susijusiam finansinių rinkų nuosmukiui.

Rezultatai ir išvados

Darbo rezultatai:

1. Sukurta, skatinamuoju mokymu paremta, apsimokanti programų sistema automatizuojanti vertybinių popierių prekybą sprendimo priėmimo aspektu:
    - Parengta siūloma programinė architektūra;
    - Sukurta programinė biblioteka lygiagrečiais skaičiavimais paruošianti istorinius finansinius duomenis mašininiam mokymui;
    - Atlikus [LYC+20] tyrimo kodo analizę nustatyta, kad finansinių indikatorių apskaičiavimui yra naudojamos vertės, kuriose neatsižvelgta į dividendus ir akcijų padalijimus, kurie iškraipo techninius indikatorius. Šiame darbe pasiūlytas metodas panaikinantis techninių indikatorių iškraipymus;
    - Sukurta StocksRL aplinka, paremta MDP principais ir OpenAI Gym platforma. Aplinka nėra priklausoma nuo konkrečių metodų realizacijos ir yra skirta bandyti skirtingus skatinamojo mokymo algoritmus;
    - StocksRL aplinkai pritaikytos ir optimizuotos OpenAI Baselines pagrindu sukurtos stable-baselines3 skatinamojo mokymo algoritmų realizacijos;
    - Sukurta programinė biblioteka leidžianti įvertinti StocksRL aplinkoje vykdomos vertybinių popierių prekybos našumą ir vizualiai atvaizduojant prekybos rezultatus;
2. Atliekant literatūros analizę bei empirinius bandymus ištirtas skirtingų skatinamojo mokymo metodų tinkamumas naudoti finansiniams duomenims, pasirinkti efektyviausi metodai bei pasiūlyta metodika šių skatinamojo mokymo modelių optimaliam veikimui pasiekti:
    - Išnagrinėti naujausi tyrimai rodo geresnį DDPG giliojo skatinamojo mokymo metodo efektyvumą vertybinių popierių prekybos uždaviniui negu pasyvi investavimo strategija;
    - Išnagrinėti ir praktiškai pritaikyti DDPG, TD3 ir SAC giliojo skatinamojo mokymo metodai;
    - DDPG metodas iki 70%, o SAC iki 7,9% efektyvesnis negu pasyvaus investavimo bei OLMAR investavimo strategijos;
    - Naudojant to paties laikotarpio duomenis TD3 modelis sugeneravo 10,7% didesnę metinę grąžą ir 17,6% didesnį Šarpo rodiklį negu DDPG modelis. TD3 modelis buvo iki 84% efektyvesnis negu SAC modelis;
    - TD3 giliojo skatinamojo mokymo metodas labiau konverguoja ir yra mažiau jautrus hiperparametrų pokyčiams negu DDPG metodas;
    - Nuo 2019 sausio iki 2020 birželio duomenimis TD3 ir DDPG modeliai buvo efektyvesni negu Min Variance bazinis modelis. Nuo 2020 birželio iki 2020 gruodžio

Min Variance modelis perskirstė portfelį vienoje AAPL akcijų pozicijoje. Nuo 29% iki 39% padidėjusi standartinė deviacija ir nuo -20% iki -30% padidėjęs portfelio vertės didžiausias nuosmukis indikuoja rizikos padidėjimą. Min Variance modelis šiame tyrime parodė geriausius prekybos rezultatus.

Išvados ir rekomendacijos:

1. Šiame darbe pasiūlyta skatinamojo mokymo sistema leidžia tirti, Markovo sprendimų procesų principu orientuotus, skatinamojo mokymo modelius, juos taikant vertybinių popierių prekybos automatizavimo uždaviniui. Pademonstruotas šios sistemos veikimas;

2. DDPG ir TD3 giliojo skatinamojo mokymo metodai tinka vertybinių popierių prekybos uždaviniui ir didžiąją dalį laiko veikia efektyviau negu pasyvios prekybos, OLMAR ir Min Variance baziniai modeliai. Min Variance modelis diversifikuotame portfelyje yra mažiausiai efektyvus iš šiame tyrime naudotų strategijų. Tyrimo rezultatai rodo, kad Min Variance strategija gali generuoti gerus prekybos rezultatus, tačiau tik sutelkiant portfelį vienoje akcijų pozicijoje bei ženkliai padidinant prekybos riziką;

3. TD3 modelio apmokymas vertybinių popierių prekybai yra stabilesnis negu DDPG ar SAC modelių;

4. TD3 modelis geriau tinka vertybinių popierių prekybos uždaviniui negu DDPG arba SAC modeliai.

Ateities tyrimų gairės

Šiame darbe buvo nagrinėjami ir praktiškai išbandyti metodai, kurie leidžia optimizuoti vertybinių popierių portfelį remiantis biržos duomenų technine analize. Šiame tyrime nebuvo naudojama fundamentalioji įmonių analizė, profesionalių analitikų išvados ar žiniasklaidos teikiama informacija. Ateities tyrimuose siūloma atlikti portfelio optimizavimą naudojant giliojo skatinamojo mokymo agentus bei jiems suteikiant informaciją apie įmonių finansines ataskaitas, fundamentalią analizę, taip pat nustatyti daugiausiai įtakos finansų rinkoms darančius asmenis bei plačiosios žiniasklaidos šaltinius ir įtraukti jų deklaruojamą informaciją į agento aplinkos būsenos aprašą.

Šiame darbe sukurta StocksRL skatinamojo mokymo aplinka yra pritaikyta dirbti su skirtingo laiko intervalo duomenimis, pavyzdžiui valandos arba minutės intervalo, tačiau duomenų aibė yra apribota programinės atminties kiekiu. Siūloma realizuoti dinaminį duomenų nuskaitymą iš disko pagal poreikį.

Realioje akcijų biržoje kai kurios šiame darbe išsikeltos prielaidos nėra pilnai išpildomos: nėra garantijų, kad paskelbus pirkimo ar pardavimo sandorį jis bus įvykdytas pagal paskutinę uždarymo kainą. Siūloma atlikti papildomą prekybos rezultatų vertinimą modelį pritaikant realioje akcijų biržoje naudojant virtualią sąskaitą. Toks vertinimas tikėtina užtruks ilgai, tačiau tiksliau parodytų modelio našumą realiose sąlygose.

Literatūra

[Ach20] Joshua Achiam. Spinning up in deep reinforcement learning. *Dosegljivo: spinningup. openai. com.[Dostopano 12. 7. 2020]*, 2020.

[Ang12] Andrew Ang. Mean-variance investing. *Available at SSRN 2131932*, 2012.

[BCP+16] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang ir Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[BGH+15] Jamil Baz, Nicolas Granger, Campbell R Harvey, Nicolas Le Roux ir Sandy Rattray. Dissecting investment strategies in the cross section and time series. *Available at SSRN 2695101*, 2015.

[BLL92] William Brock, Josef Lakonishok ir Blake LeBaron. Simple technical trading rules and the stochastic properties of stock returns. *The Journal of finance*, 47(5):1731–1764, 1992.

[Bru17] Renato Bruni. Stock market index data and indicators for day trading as a binary classification problem. *Data in brief*, 10:569–575, 2017.

[CG08] Rohit Choudhry ir Kumkum Garg. A hybrid machine learning system for stock market forecasting. *World Academy of Science, Engineering and Technology*, 39(3):315–318, 2008.

[CNL14] Terence Tai-Leung Chong, Wing-Kam Ng ir Venus Khim-Sen Liew. Revisiting the performance of macd and rsi oscillators. *Journal of risk and financial management*, 7(1):1–12, 2014.

[DBK+16] Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren ir Qionghai Dai. Deep direct reinforcement learning for financial signal representation and trading. *IEEE transactions on neural networks and learning systems*, 28(3):653–664, 2016.

[DHK+17] Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol ir k.t. Openai baselines, 2017.

[Edd20] Joe Jevnik Eddie Hebert Richard Frank. Zipline. *GitHub repository*, 2020.

[FVM18] Scott Fujimoto, Herke Van Hoof ir David Meger. Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*, 2018.

[GM13] Raul Garreta ir Guillermo Moncecchi. *Learning scikit-learn: machine learning in python*. Packt Publishing Ltd, 2013.

[GNC19] Luis A Garrido, Rajiv Nishtala ir Paul Carpenter. Continuous-action reinforcement learning for memory allocation in virtualized servers. *International Conference on High Performance Computing*, p. 13–24. Springer, 2019.

[Gur+18] Ikhlaas Gurrib ir k.t. Performance of the average directional index as a market timing tool for the most actively traded usd based currency pairs. *Banks and Bank Systems*, 13(3):58–70, 2018.

[HIB+18] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup ir David Meger. Deep reinforcement learning that matters. *Proceedings of the AAAI Conference on Artificial Intelligence*, tom. 32 numeris 1, 2018.

[HZA+18] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel ir Sergey Levine. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. *International Conference on Machine Learning*, p. 1861–1870. PMLR, 2018.

[Jia20] Weiwei Jiang. Applications of deep learning in stock market prediction: recent progress. *arXiv preprint arXiv:2003.01859*, 2020.

[LCM+04] Daeyeol Lee, Michelle L Conroy, Benjamin P McGreevy ir Dominic J Barraclough. Reinforcement learning and decision making in monkeys during a competitive game. *Cognitive brain research*, 22(1):45–58, 2004.

[Lee01] Jae Won Lee. Stock price prediction using reinforcement learning. *ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No. 01TH8570)*, tom. 1, p. 690–695. IEEE, 2001.

[LH12] Bin Li ir Steven CH Hoi. On-line portfolio selection with moving average reversion. *arXiv preprint arXiv:1206.4626*, 2012.

[LHP+15] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver ir Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[LYC+20] Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao ir Christina Dan Wang. Finrl: a deep reinforcement learning library for automated stock trading in quantitative finance. *arXiv preprint arXiv:2011.09607*, 2020.

[LKK+19] Jinho Lee, Raehyun Kim, Yookyung Koh ir Jaewoo Kang. Global stock market prediction based on stock chart images using deep q-network. *IEEE Access*, 7:167260–167277, 2019.

[LLD19] Jue Liu, Zhuocheng Lu ir Wei Du. Combining enterprise knowledge graph and news sentiment analysis for stock price prediction. *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.

[LZR19] Bryan Lim, Stefan Zohren ir Stephen Roberts. Enhancing time-series momentum strategies using deep neural networks. *The Journal of Financial Data Science*, 1(4):19–38, 2019.

[Mal03] Burton G Malkiel. Passive investment strategies and efficient markets. *European Financial Management*, 9(1):1–10, 2003.

[Mar21] Robert Andrew Martin. Pyportfolioopt: portfolio optimization in python. *Journal of Open Source Software*, 6(61):3066, 2021. DOI: 10.21105/joss.03066. URL: https://doi.org/10.21105/joss.03066.

[Min61] Marvin Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, 1961.

[MKS+15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu ir k.t. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[MOP12] Tobias J Moskowitz, Yao Hua Ooi ir Lasse Heje Pedersen. Time series momentum. *Journal of financial economics*, 104(2):228–250, 2012.

[MPC+16] Mansoor Maitah, Petr Prochazka, Michal Cermak ir Karel Šrédl. Commodity channel index: evaluation of trading rule of agricultural commodities. *International Journal of Economics and Financial Issues*, 6(1):176–178, 2016.

[PHD+17] Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y Chen, Xi Chen, Tamim Asfour, Pieter Abbeel ir Marcin Andrychowicz. Parameter space noise for exploration. *arXiv preprint arXiv:1706.01905*, 2017.

[RHE+19] Antonin Raffin, Ashley Hill, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto ir Noah Dormann. Stable baselines3. *GitHub repository*, 2019.

[SB18] Richard S Sutton ir Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[Sha70] William F Sharpe. *Portfolio theory and capital markets*. McGraw-Hill College, 1970.

[SLH+14] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra ir Martin Riedmiller. Deterministic policy gradient algorithms. 2014.

[UO30] George E Uhlenbeck ir Leonard S Ornstein. On the theory of the brownian motion. *Physical review*, 36(5):823, 1930.

[WAM17] Bin Weng, Mohamed A Ahmed ir Fadel M Megahed. Stock market one-day ahead movement prediction using disparate data sources. *Expert Systems with Applications*, 79:153–163, 2017.

[XLZ+18] Zhuoran Xiong, Xiao-Yang Liu, Shan Zhong, Hongyang Yang ir Anwar Walid. Practical deep reinforcement learning approach for stock trading. *arXiv preprint arXiv:1811.07522*, 2018.

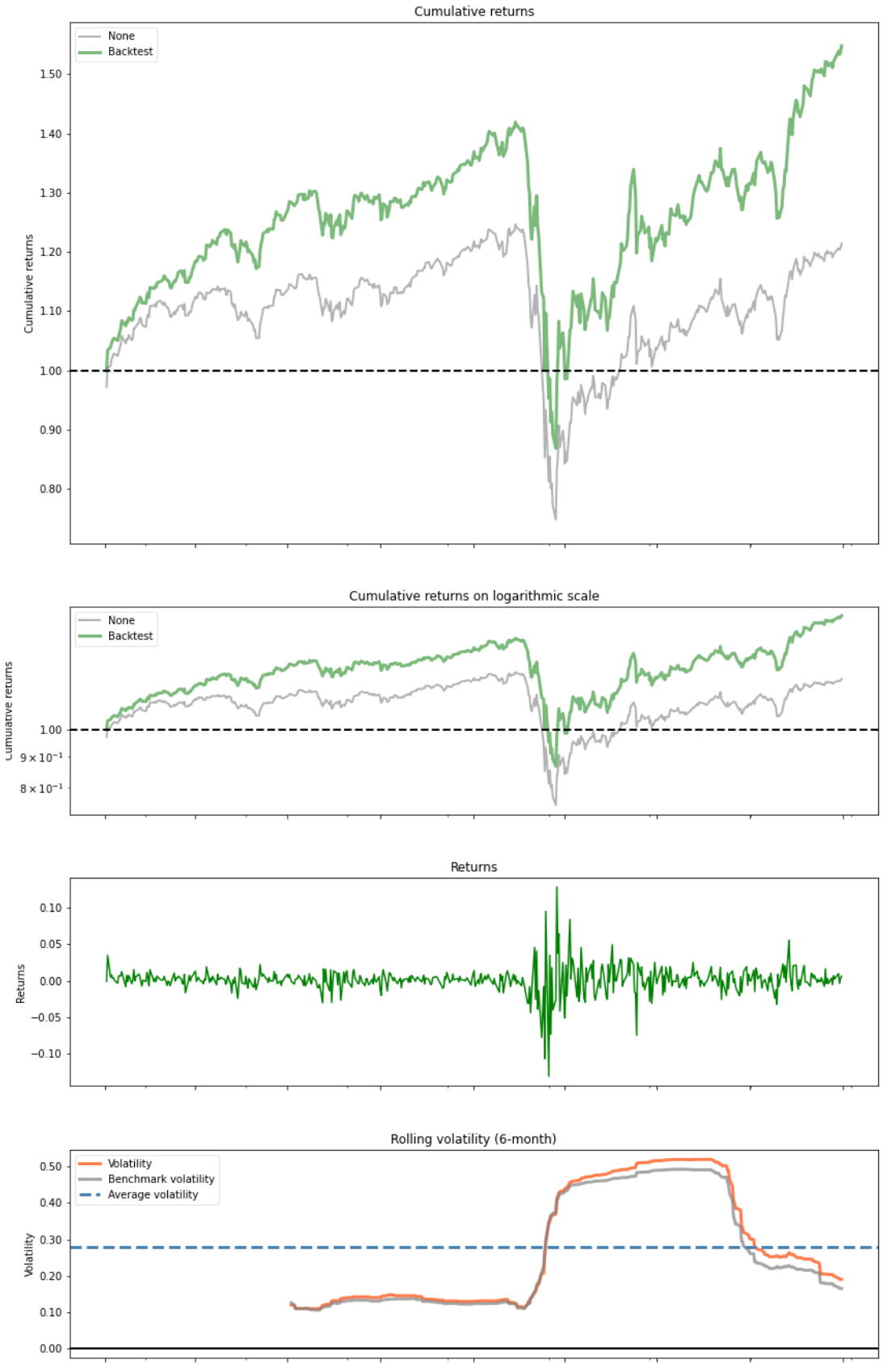
[ZJS21] Huanming Zhang, Zhengyong Jiang ir Jionglong Su. A deep deterministic policy gradient-based strategy for stocks portfolio management. *arXiv preprint arXiv:2103.11455*, 2021.

[ZZR20] Zihao Zhang, Stefan Zohren ir Stephen Roberts. Deep reinforcement learning for trading. *The Journal of Financial Data Science*, 2(2):25–40, 2020.

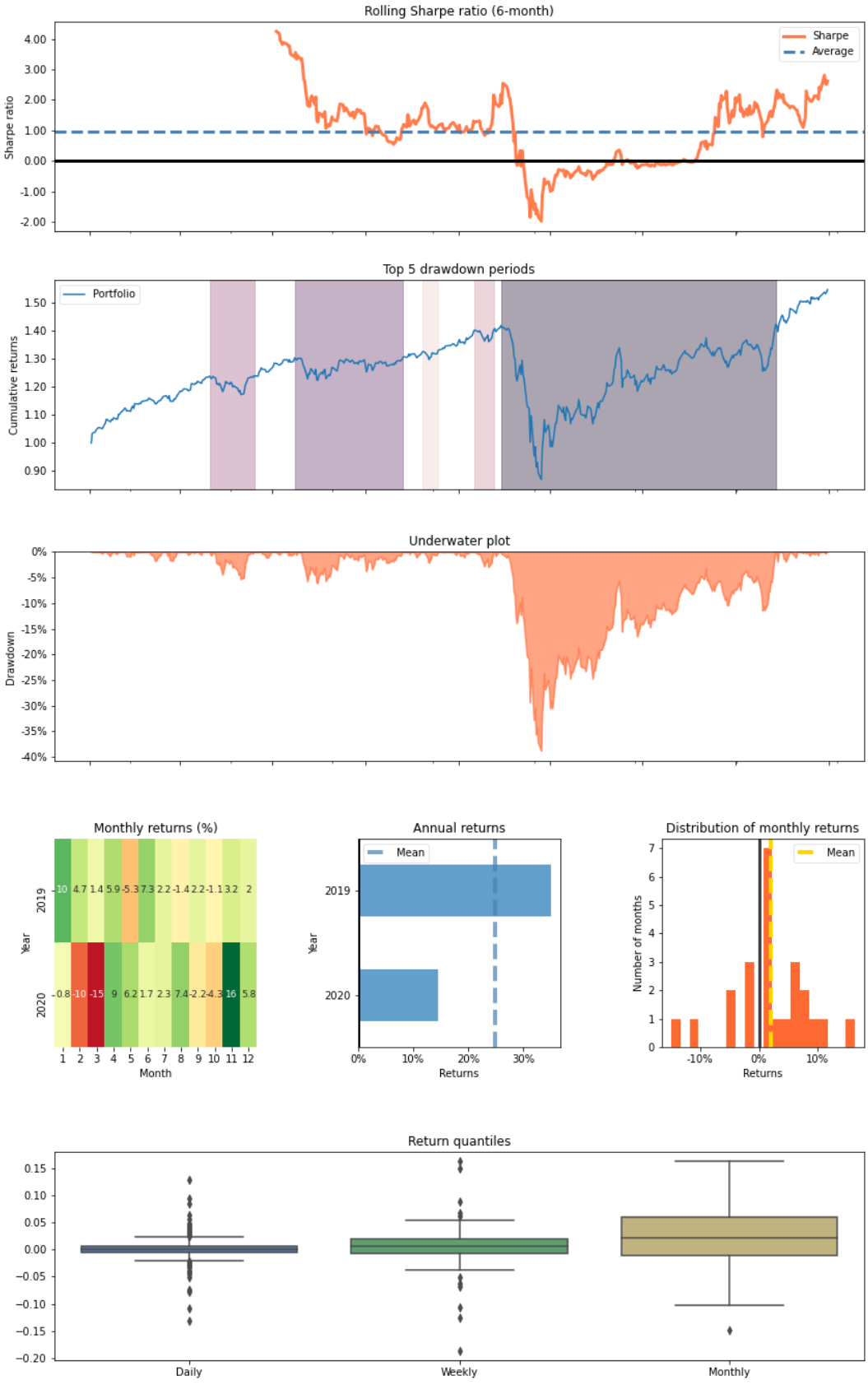
Santrumpos

s	būsena
S	būsenų aibė
a	veiksmas
A	veiksmų aibė
k	veiksmų skaičius
t	diskrečiai pamatuojamas laiko momentas
γ	diskonto norma (parametras)
A_t	veiksmas laiko momentu t
R_t	atlygis laiko momentu t
G_t	nauda (diskontuota atlygių suma) laiko momentu t
$E[X]$	atsitiktinio kintamojo X labiausiai tikėtinas dydis

A priedas

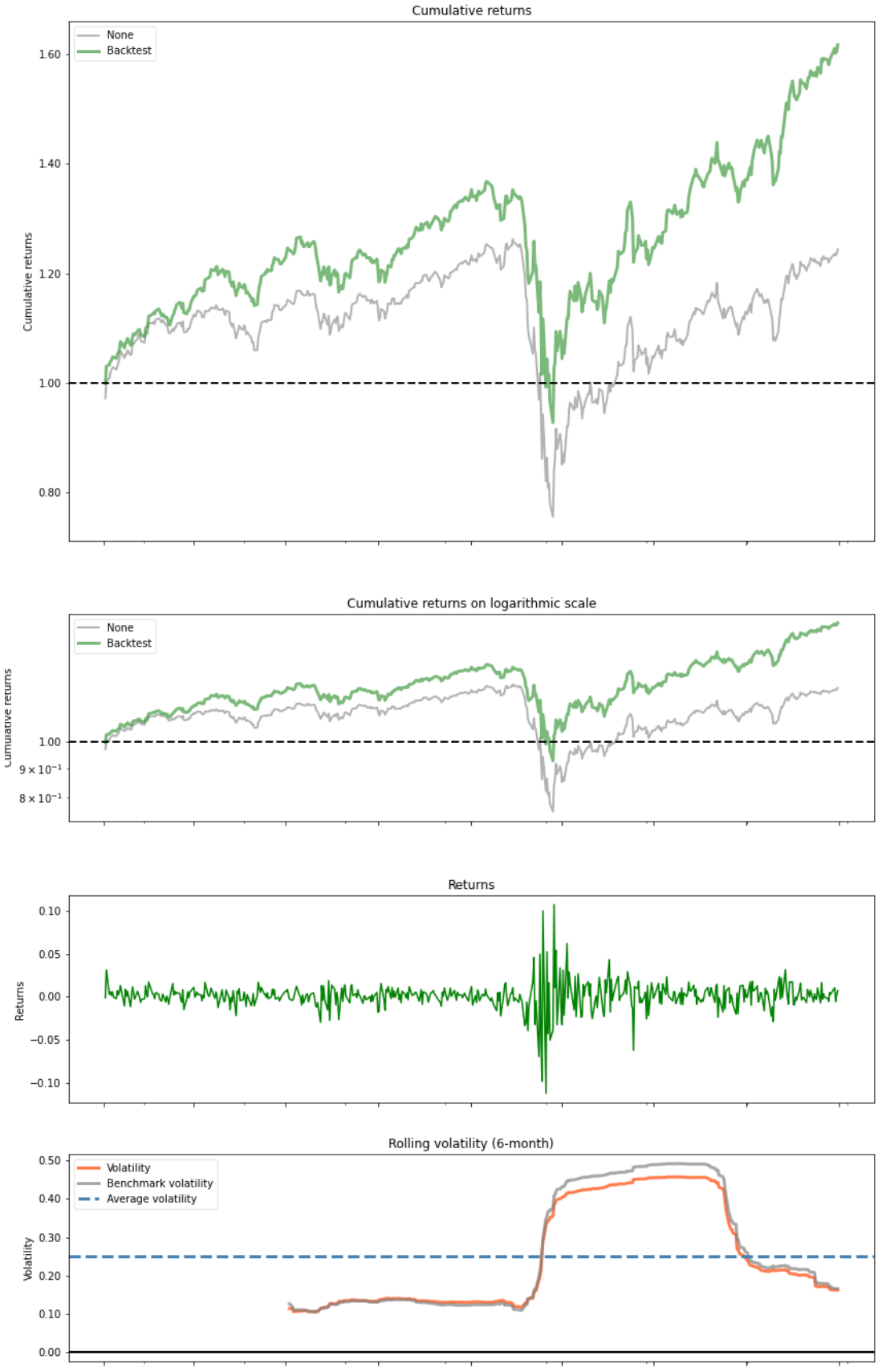


24 pav. DDPG modelio prekybos rezultatai (1)

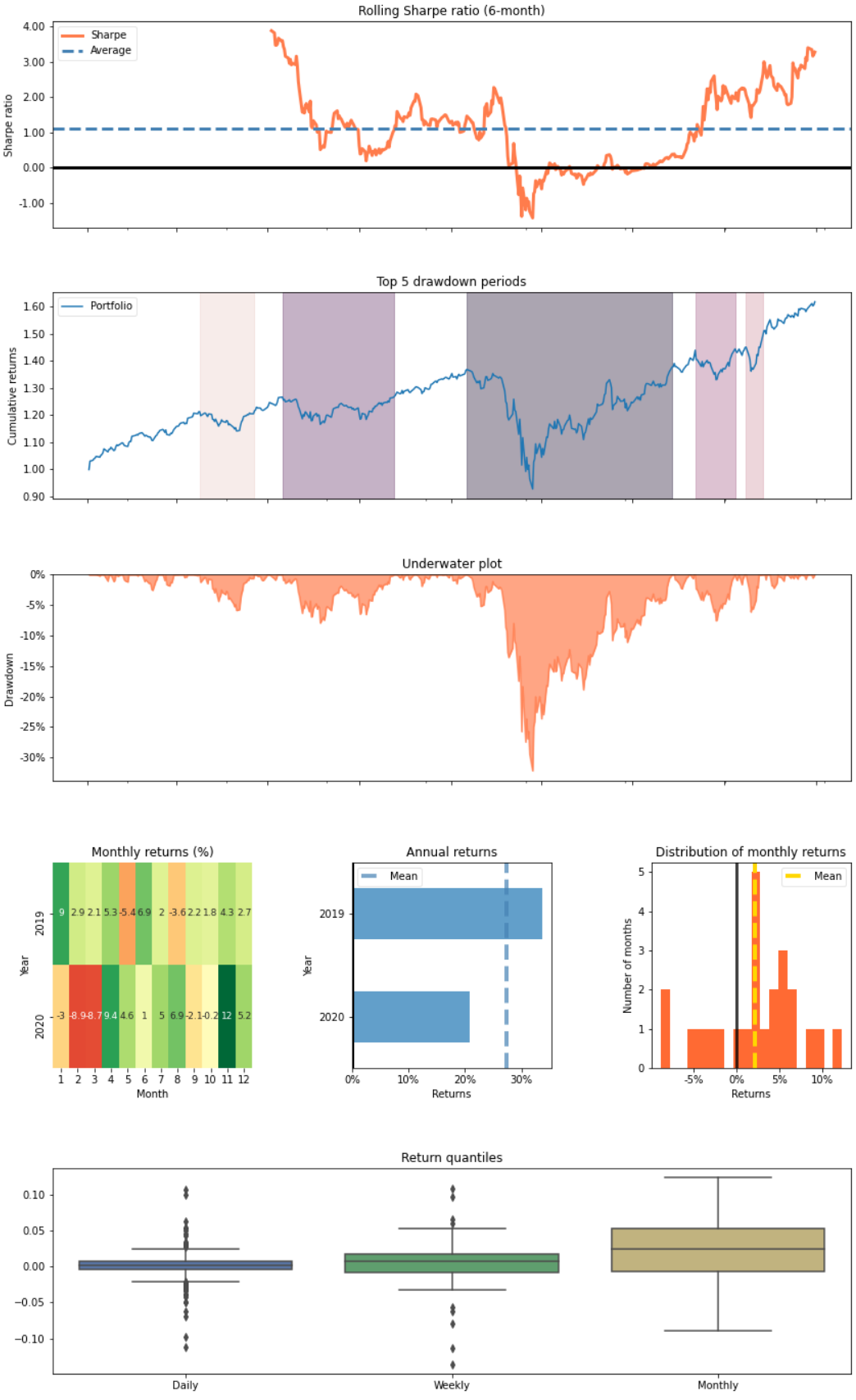


25 pav. DDPG modelio prekybos rezultatai (2)

B priedas

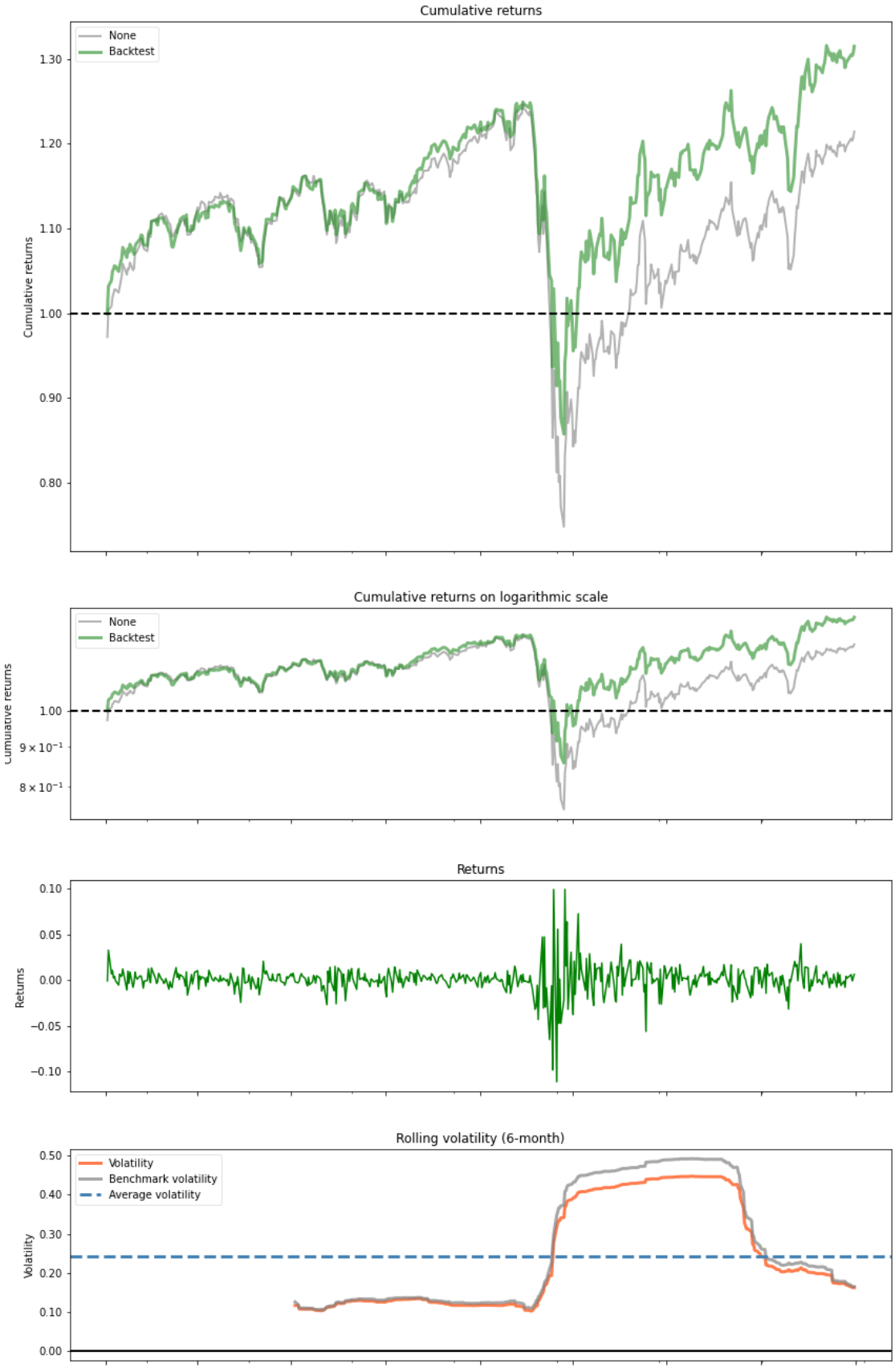


26 pav. TD3 modelio prekybos rezultatai (1)

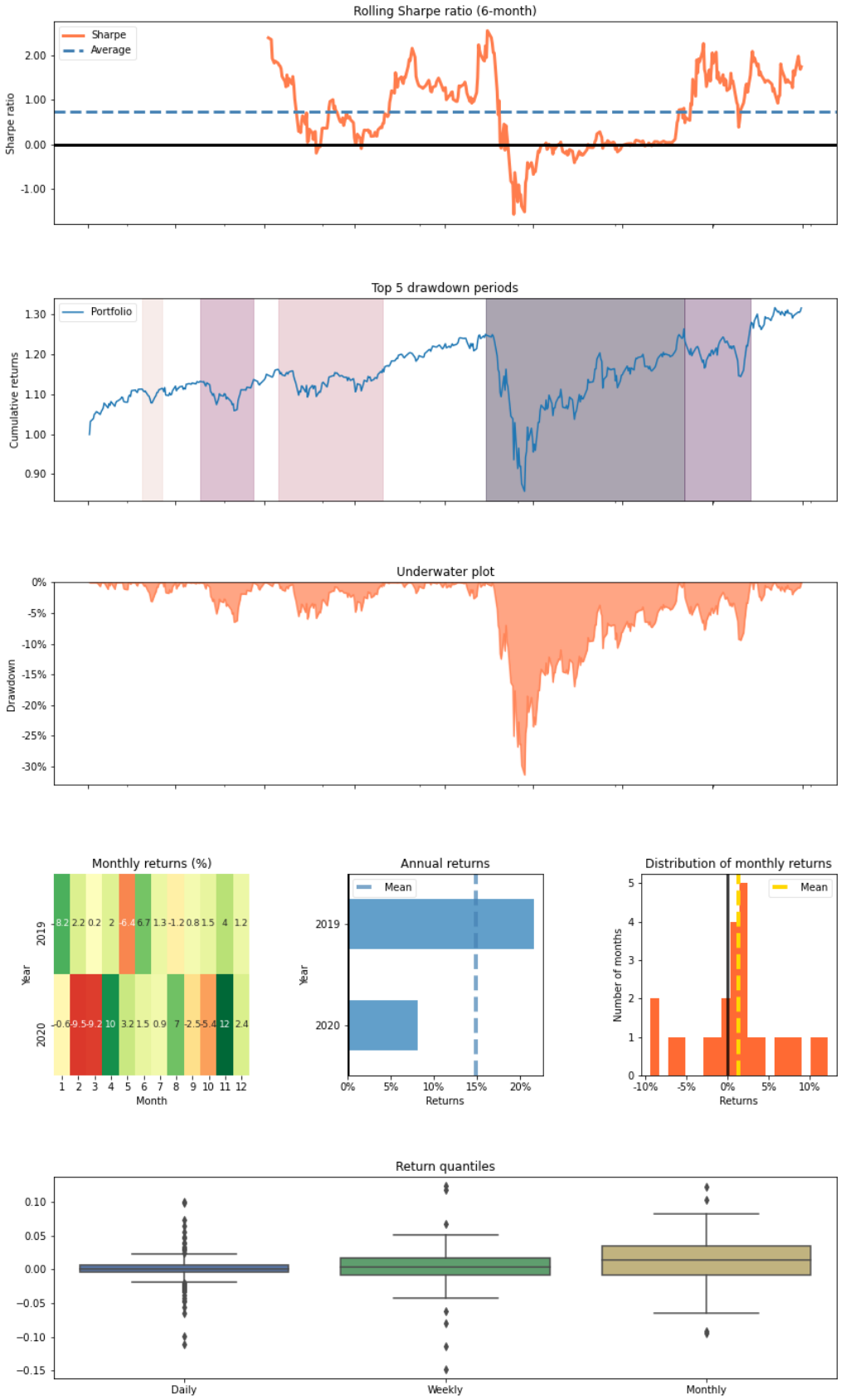


27 pav. TD3 modelio prekybos rezultatai (2)

C priedas

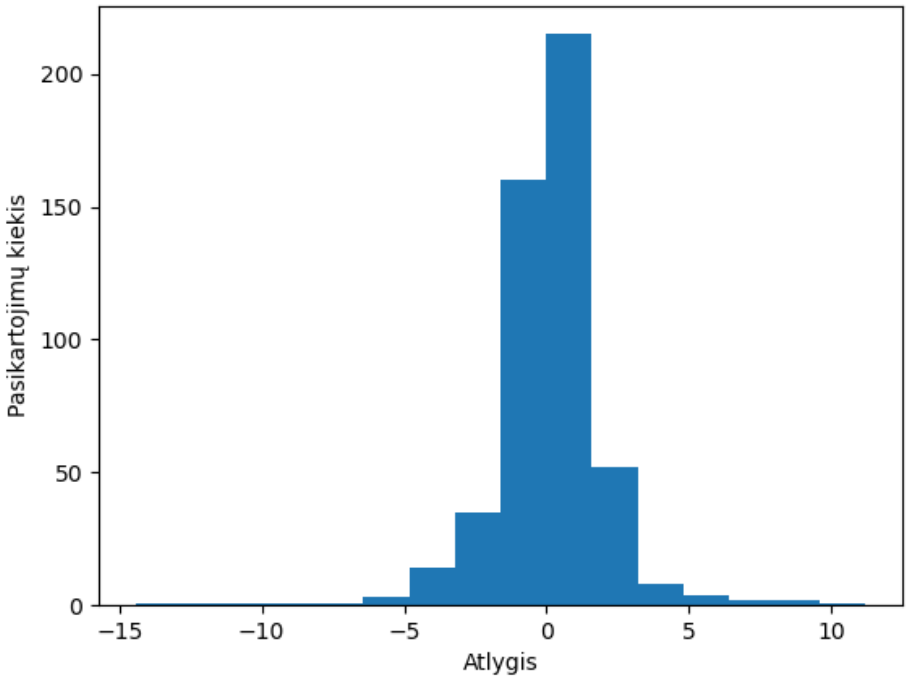


28 pav. SAC modelio prekybos rezultatai

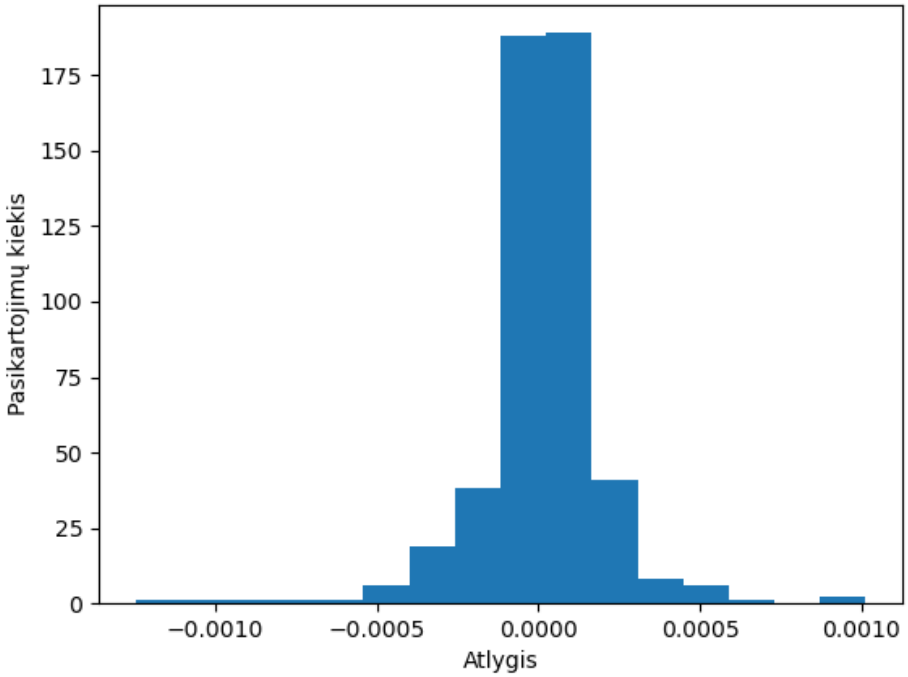


29 pav. SAC modelio prekybos rezultatai

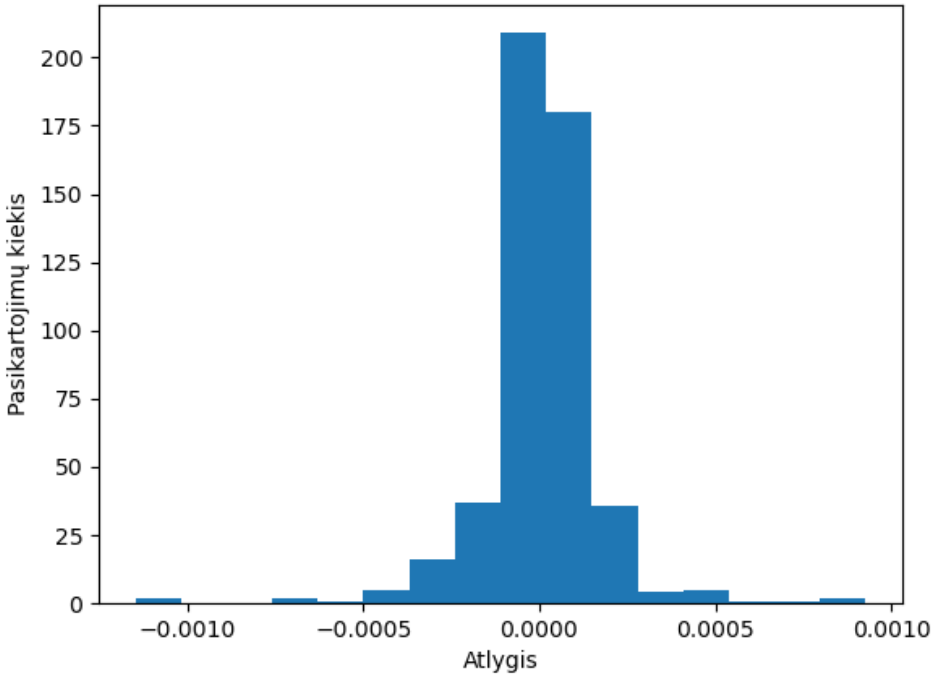
D priedas



30 pav. DDPG prekybos atlygis



31 pav. TD3 prekybos atlygis



32 pav. SAC prekybos atlygis