



**Kauno technologijos universitetas**  
Matematikos ir gamtos mokslų fakultetas

**Paslėptųjų Markovo grandinių taikymas kriptovaliutų kainų  
prognozei**

Baigiamasis magistro projektas

---

**Lina Sakalavičiūtė**  
Projekto autorė

**Assoc. prof. dr. Audrius Kabašinskas**  
Vadovas

---

**Kaunas, 2020**



**Kauno technologijos universitetas**  
Matematikos ir gamtos mokslų fakultetas

# **Paslėptųjų Markovo grandinių taikymas kriptovaliutų kainų prognozei**

Baigiamasis magistro projektas  
Taikomoji Matematika (6211AX006)

---

**Lina Sakalavičiūtė**  
Projekto autorė

**Assoc. prof. dr. Audrius Kabašinskas**  
Vadovas

**Lekt. dr. Mindaugas Bražėnas**  
Recenzentas

---

**Kaunas, 2020**



**Kauno technologijos universitetas**  
Matematikos ir gamtos mokslų fakultetas  
Lina Sakalavičiūtė

## **Baigiamojo projekto pavadinimas**

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Linos Sakalavičiūtės, baigiamasis projektas tema „Paslėptųjų Markovo grandinių taikymas kriptovaliutų kainų prognozei“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

---

(vardą ir pavardę įrašyti ranka)

---

(parašas)

## Turinys

<b>IVADAS</b> .....	<b>9</b>
1. LITERATŪROS APŽVALGA.....	11
1.1. Kriptovaliutų prekyba.....	11
1.2. Kriptovaliutų kainoms įtakos turintys veiksniai .....	12
1.3. Kriptovaliutų prognozavimo metodai.....	14
1.4. Paslėptųjų Markovo modelių taikymų apžvalga .....	16
1.4.1. Paslėptųjų Markovo grandinių taikymas režimų identifikavimui .....	19
1.5. Atliekamo tyrimo metodika.....	20
2. TYRIMO METODOLOGIJA .....	21
2.1. Paslėptieji Markovo modeliai.....	21
2.1.1. Paslėptojo Markovo modelio apibrėžimas .....	22
2.2. Pirmyn ir atgal algoritmai.....	25
2.3. Viterbi algoritmas.....	26
2.4. Baum-Welch algoritmas.....	26
2.5. Modelio patikimumo vertinimas .....	27
2.5.1. Sumaišymo matrica.....	27
2.5.2. Pirmos eilės Markovo grandinės testas .....	29
2.6. HMM pritaikymo kriptovaliutų kainoms tyrimo eiga .....	29
3. TYRIMO REZULTATAI IR JŲ APIBENDRINIMAS .....	31
3.1. Duomenys ir aprašomoji statistika .....	31
3.2. HMM modelio pritaikymas paslėptųjų būsenų (režimų) nustatymui kriptovaliutų rinkoje .....	33
3.2.1. HMM modelio optimalaus būsenų skaičiaus radimas.....	35
3.2.2. Kriptovaliutų rinkos režimų nustatymas naudojant 3 būsenų HMM .....	37
3.3. Kainos krypties prognozė dviejų būsenų paslėptuoju Markovo modeliu.....	40
3.3.1. Pirmos eilės Markovo grandinių testas .....	41
3.3.2. Sudaryto modelio pritaikymo eiga .....	43
3.3.3. Kriptovaliutų kainos krypties prognozės HMM modeliu įvertinimas.....	44
3.3.4. HMM modelio tikslumo priklausomybė nuo stebėjimų sekos dydžio.....	46
3.3.5. Agregavimo lygio įtaka rezultatams .....	47
3.3.6. HMM ir kitų mašininio mokymosi metodų tikslumo palyginimas .....	49
3.4. Tyrimo apibendrinimas ir rekomendacijos.....	50
3.5. Programinė realizacija .....	51
<b>IŠVADOS</b> .....	<b>52</b>
<b>LITERATŪROS SARAŠAS</b> .....	<b>53</b>
<b>PRIEDAI</b> .....	<b>56</b>

## Paveikslų sąrašas

<b>1 pav.</b> Paslėptojo Markovo modelio dviejų būsenų perėjimo schema, kai turimi trys stebėjimų simboliai (Toledo ir Katz 2009) .....	23
<b>2 pav.</b> Paslėptojo Markovo modelio bendroji struktūra (Kim, Jeong ir Lee 2019) .....	24
<b>3 pav.</b> Sumaišymo matrica .....	27
<b>4 pav.</b> HMM modeliavimo schema .....	30
<b>5 pav.</b> Bitkoino, eterio, laitkoino ir teterio kainos kitimo (kairėje) ir gražų grafikai .....	31
<b>6 pav.</b> AIC ir BIC reikšmių priklausomybė nuo stebėjimų skaičiaus .....	36
<b>7 pav.</b> Bitkoino 3 būsenų HMM modelio ATR, gražų, posteriorinių tikimybių ir būsenų grafikai .	38
<b>8 pav.</b> Eterio 3 būsenų HMM modelio ATR, gražų, posteriorinių tikimybių ir būsenų grafikai.....	38
<b>9 pav.</b> Laitkoino 3 būsenų HMM modelio ATR, gražų, posteriorinių tikimybių ir būsenų grafikai	39
<b>10 pav.</b> Teterio 3 būsenų HMM modelio ATR, gražų, posteriorinių tikimybių ir būsenų grafikai .	40
<b>11 pav.</b> Bitkoino apmokymo duomenų (kairėje) ir testavimo duomenų sumaišymo matrica .....	44
<b>12 pav.</b> Eterio apmokymo duomenų (kairėje) ir testavimo duomenų sumaišymo matrica.....	45
<b>13 pav.</b> Laitkoino apmokymo duomenų (kairėje) ir testavimo duomenų sumaišymo matrica.....	45
<b>14 pav.</b> Modelio tikslumo ( <i>angl.</i> accuracy) priklausomybė nuo stebėjimo sekos skaičiaus .....	47

## Lentelių sąrašas

<b>1 lentelė.</b> Pagrindinės HMM charakteristikos .....	22
<b>2 lentelė.</b> Bitkoino, eterio, lairkoino ir teterio gražų pagrindinės charakteristikos .....	32
<b>3 lentelė.</b> HMM modeliai pagal mažiausias AIC ir BIC informacinius kriterijus .....	35
<b>4 lentelė.</b> 3 būsenų HMM parametrų reikšmės .....	37
<b>5 lentelė.</b> Pirmos eilės Markovo grandinių testo $p$ -reikšmių rezultatai.....	42
<b>6 lentelė.</b> Markovo savybių testų rezultatai .....	43
<b>7 lentelė.</b> HMM prognozės metodo tikslumo rodiklių reikšmės.....	45
<b>8 lentelė.</b> Modelio tikslumas agreguotiems duomenims .....	48
<b>9 lentelė.</b> Skirtingų metodų vidutiniai prognozės tikslumo rodikliai.....	49

Sakalavičiūtė, Lina. Paslėptųjų Markovo grandinių taikymas kriptovaliutų kainų prognozei. Magistro baigiamasis projektas / vadovas assoc. prof. dr. Audrius Kabašinskas; Kauno technologijos universitetas, Matematikos ir gamtos mokslų fakultetas.

Studijų kryptis ir sritis: matematikos mokslai (A), taikomoji matematika (A02).

Reikšminiai žodžiai: paslėptieji Markovo modeliai, Baum-Welch, Viterbi, kriptovaliutos, prognozė. Kaunas, 2020. 86 p.

## Santrauka

Kriptovaliutų kainos pasižymi itin dideliu kintamumu lyginant su tradicinėmis finansinėmis laiko eilutėmis, todėl ne visada pavyksta pasiekti gerų ir patikimų kainos prognozės rezultatų įprastais modeliais. Šio darbo tikslas yra pritaikyti paslėptojo Markovo grandinių metodą nepastoviųjų kriptovaliutų kainos krypties prognozei bei nustatyti kainas veikiančius režimus.

Kriptovaliutų kainoms įtakos turi įvairūs politiniai ir ekonominiai veiksniai. Apskaičiavus logaritmines grąžas ir pritaikius 2, 3, 4 ir 5 būsenų paslėptuosius Markovo modelius nustatyta, kad geriausia naudoti 3 paslėptąsias būsenas režimų identifikavimui. Šis modelis suskirstė kriptovaliutų grąžas pagal jų kintamumo dydį į mažo, vidutinio ir ekstremalaus kintamumo režimus.

Kainų krypties prognozei sudarytas paslėptasis Markovo modelis sudiskretizavus kriptovaliutų kainų grąžas į 5 intervalus. Pagal juos sudaryta stebėjimų seka 3 metų duomenims. Atliktas modelio apmokymas naudojant *Baum-Welch* algoritimą ir laikant, kad yra dvi paslėptosios būsenos atitinkančios kainos kilimą ir kainos kritimą. Atlikus 3 mėnesių kainos krypties prognozė *Viterbi* algoritmu įvertintas modelio tikslumas, kuris palygintas su mašininio mokymo metodais (atsitiktinių miškų algoritmu, dirbtinių neuroninių tinklų metodu ir naiviu Bajeso klasifikatoriumi). Vidutiniškai paslėptųjų Markovo modeliu atlikta prognozė yra tikslesnė kai turimi kiekvienos dienos stebėjimai.

Sakalavičiūtė, Lina. The Application of Hidden Markov Chains for Cryptocurrency Price Prediction. Master's Final Degree Project / supervisor assoc. prof. dr. Audrius Kabašinskas; The Faculty of Mathematics and Natural Sciences, Kaunas University of Technology.

Study field and area: Applied Mathematics (A01), Mathematical Sciences (A).

Keywords: Hidden Markov Model, Baum-Welch, Viterbi, cryptocurrencies, prediction.

Kaunas, 2020. 86 pages.

### **Summary**

Cryptocurrency prices have high volatility compared to traditional financial time series. This is the reason why it is hard to predict prices properly using standard models. The aim of this work is to apply the hidden Markov chain model to predict cryptocurrency price direction and to determine regime switching impact to cryptocurrency prices.

Various political actions and economic regimes affect cryptocurrency prices. Logarithmic returns used for 2-, 3-, 4-, 5-state hidden Markov model application. It was found that 3-state model is the best to find hidden regimes in cryptocurrency price changes. This model divided cryptocurrency returns according to their volatility into low, medium, and extreme volatility regimes.

For the price direction forecast the hidden Markov model was constructed by discretizing the cryptocurrency prices returns into 5 intervals. Based on them, a sequence of observations was formed for 3-year data. The assumption was made that there are two hidden states which represents increasing and decreasing prices. Baum-Welch algorithm was used for learning the model. Then predicted price directions for 3 months data and calculated the accuracy of the model, which was compared with machine learning methods (random forest algorithm, artificial neural network method, and naive Bayesian classifier). In average the prediction made by the hidden Markov model is more accurate than other methods when daily observations are considered.



## Įvadas

Pastaraisiais metais itin išpopuliarėjusi investicinė naujovė – kriptovaliutos. Kriptovaliutos yra skaitmeninis finansinis turtas, kurio nuosavybę ir nuosavybės perdavimą garantuoja decentralizuota kriptografinė technologija, o ne bankas ar kita patikima trečioji šalis (Giudici, Milne ir Vinogradov 2019).

Kriptovaliutos atsirado visai neseniai. Bitkoinas, pirmoji decentralizuota kriptovaliuta, viešai pristatytas dar 2009 m., o didesnio susidomėjimo sulaukė tik 2017 m. Tuomet bitkoino kaina pradėjo sparčiai augti ir pakilo beveik iki 20 000 dolerių. Tačiau dėl savo nepastovumo, kelių dienų laikotarpiu kaina nukrito 30 %. Dėl didėjančio investuotojų pasitikėjimo kriptovaliutomis bei galimybės sukurti naujų, šiuo metu pasaulyje yra apie 5 000 kriptovaliutų, kuriomis yra prekiaujama. Klestint kriptovaliutos prekybai pastebėtas didesnis susidomėjimas bei aktyvumas atliekant įvairius tyrimus. Daugiau kaip 85 % naujų straipsnių, susijusių su kriptovaliutų prekyba, atsirado per pastaruosius dvejus metus (Fang et. al. 2020). Tai rodo, kad turima nauja tyrimų kryptis finansų prekyboje bei vis užsimenama, kad kriptovaliutos gali tapti ateities pinigais.

Lietuvoje taip pat siekiama įdiegti naujas finansines technologijas, turinčias reikšmingą poveikį finansų rinkoms. Šiuo metu yra bandoma nauja blokų grandinės technologija, pagrįsta technologinė platforma *LBCChain*, kurios startas numatytas dar šiais metais. Be to, Lietuvos bankas ketina išleisti pirmąją pasaulyje skaitmeninę kolekcinę monetą *LBCoin*, kuri ir yra kuriama taikant blokų grandinės technologiją (Lietuvos bankas 2020).

Nors kriptovaliutos yra apibrėžiamos kaip finansinis turtas, reali ekonominė vertė lieka neaiški, kadangi sandoriuose yra nurodomas tik perduodamas kriptovaliutų kiekis. Rinkos vertę galima susidaryti įvertinus valiutų kursus esamų dekretinių<sup>1</sup> valiutų atžvilgiu (Giudici, Milne ir Vinogradov 2019). Tai įmanoma dėl nuolat vykstančios kriptovaliutų prekybos, o norint įsigyti bitkoiną, galima pirkti šimtają ar dešimtają vieno bitkoino dalį.

Vis tik šis investavimo būdas yra rizikingas dėl didelių vertės svyravimų. Kita vertus tai sukuria ir prielaidas dideliame uždarbiui gauti. Pagrindiniai būdai gauti pinigine naudą iš kriptovaliutų yra diversifikuoti investicinius portfelius, užsiimti kriptovaliutų „kasyba“ arba išnaudoti didelius svyravimus – aktyviai prekiaujant kriptovaliutų rinkoje.

---

<sup>1</sup> Dekretiniai pinigai (angl. Fiat money) - tai yra dekretu išleisti valstybiniai popieriniai pinigai, nepadengti aukso atsargomis, nekonvertuojami į auksą ir naudojami tik kaip mainų priemonė.

Pagrindinė problema, susijusi su investavimu į kriptovaliutas, yra rinkos nepastovumas, dėl kurio tradiciniai finansinių laiko eilučių prognozės metodai nėra tinkami naudojimui. Pavyzdžiui, akcijų kainų kitimui taikomas Markovo grandinių modelis realiose sistemose tik iš dalies ir su tam tikra paklaida atitinka teorinius modelius. Dėl tos priežasties geriau yra taikyti paslėptuosius Markovo grandinių modelius, kuriuose yra naudojami du stochastiniai procesai (abu atskirai): be pagrindinio, kuris yra stebimas (pvz., kriptovaliutos kainos), yra ir kitas, apibūdinantis „sistemos būseną“, kuri nėra tiesiogiai stebima. Paslėptas stochastinis procesas, apibūdinantis sistemos būseną, yra Markovo grandinė (Erlwein, Mitra ir Roman 2012).

**Darbo tikslas** – nustatyti, ar kintant kriptovaliutos kainai egzistuoja nežinoma (paslėpta) Markovo grandinės dalis bei atlikti kriptovaliutos kainos krypties prognozę.

**Darbo uždaviniai:**

- optimalus paslėptųjų būsenų skaičiaus nustatymas;
- pritaikyti paslėptąjį Markovo modelį, kad nustatyti paslėptųjų režimų poveikį kriptovaliutų kainoms;
- sudaryti diskretųjį HMM modelį kriptovaliutų kainų krypties prognozei;
- įvertinamas modelio tikslumas apskaičiuojant sumaišymo matricą;
- palyginti prognozės tikslumą su kitais metodais.

Visi darbe reikalingi skaičiavimai atliekami R programine įranga.

## 1. Literatūros apžvalga

Skaitmeninės technologijos ir jų naudojimas mokslinei, ekonominei ar socialinei informacijai apdoroti praplėtė žmogaus galimybes. Vis daugiau procesų yra perkeliama į virtualią erdvę. Keičiasi ir požiūris į pinigus. Žmonės ima rečiau atsiskaitinėti grynaisiais pinigais ir palaipsniui pereina prie bankinių kortelių, elektroninių pinigų ar netgi kriptovaliutų naudojimo. Pasaulyje daugėja vietų, kuriose galima atsiskaityti už prekes ir paslaugas kriptovaliutomis. Šiuo metu bikoinus priima tokios pasaulinės kompanijos kaip *Wikipedia*, *Microsoft*, *Burger King*, *KFC*, *Subway*, *Twitch* ir daugelis kitų (Beigel, 2020). Lietuvoje *Varle.lt* ir *Topocentras.lt* internetinėse parduotuvėse taip pat galimas atsiskaitymas kriptovaliutomis (bitkoinais, laitekoinais, eteriais). Tačiau kitos didžiosios įmonės nėra įsidedusio sistemų tokiam atsiskaitymo būdui.

Investavimas į kriptovaliutos pastaruoju metu ypač aktualus tyrimų objektas. Pats paprasčiausias investavimo būdas yra tiesiog pirkti, tikintis, kad kriptovaliutos pabrangs ir vėliau bus galima jas pelningai parduoti. Lygiai taip pat, kaip iki šiol daroma su investiciniais fondais ar įmonių akcijomis.

### 1.1. Kriptovaliutų prekyba

Kriptovaliutos apibrėžiamos kaip blokų grandinėje (*angl.* blockchain) esantis turtas, kurį galima mainyti ar perduoti kitiems tinklo vartotojams ir kuris neteikia jokios kitos naudos išskyrus tai, kad yra naudojamas kaip mokėjimo priemonė (Giudici, Milne ir Vinogradov 2019). Jos gali būti dviejų rūšių:

1. Kintamos vertės (*angl.* floating). Tokio tipo kriptovaliutų kiekis yra fiksuotas, o rinkos kainą nustato pasiūla ir paklausa.
2. Stabilios vertės (*angl.* stablecoins). Fiksuota kaina išlaikoma kriptovaliutas padengiant realiu turtu: grynaisiais pinigais, vertybiniais popieriais. Pavyzdžiui, vienas teteris (*angl.* Tether, USDT) kriptovaliutos vienetas yra padengtas 1 JAV doleriu (Tether.to 2020).

2020 m. balandžio 18 d. duomenimis egzistuoja 21 434 kriptovaliutų prekybos vietų, kuriuose prekiaujama 5 361 kriptovaliuta. Bendra rinkos vertė siekia 208 milijardų dolerių (CoinMarketCap 2020), kuri apskaičiuojama sudedant visų kriptovaliutų vertes doleriais. Spalio mėnesį 10 populiariausių kriptovaliutų sudarė 85 % bendros rinkos vertės (Yahoo Finance, 2019). Šiuo metu bitkoinas išlieka dominuojanti kriptovaliuta ir sudaro 63,5 % rinkos vertės.

Kripto valiutų prekyba – tai kripto valiutų pirkimas ir pardavimas, siekiant gauti pelno. Dėl naudojamos blokų grandinės technologijos galima išskirti tokius kripto valiutų prekybos privalumus (Fang et. al. 2020):

- **24 valandų prekyba.** Kripto valiutų rinka yra prieinama 24 valandas per parą, 7 dienas per savaitę.
- **Decentralizacija.** Vienas didžiausių kripto valiutų pranašumų yra tas, kad juose nėra finansų tarpininkų, tai leidžia sumažinti operacijų sąnaudas. Be to, ši ypatybė gali patikti vartotojams, kurie nepasitiki tradicinėmis sistemomis. Kripto valiutos veikimo principai, vystymasis ir funkcionavimas priklauso nuo jomis besinaudojančios bendruomenės daugumos sprendimo.
- **Anonimiškumas.** Sandoriams kripto valiuta nereikalinga jokia konfidenciali informacija (vardas, pavardė, adresas ir kt.), tačiau kiekvienas sandoris registruojamas ir yra viešai žinomas pats sandorio sudarymo faktas. Todėl kiekvienas sandoris kripto valiuta gali būti identifikuotas, tačiau sandoriai iš esmės išlieka anonimiški.
- **Skaidrumas.** Kripto valiutos užtikrina skaidrumą, nes įprastais atvejais kiekvienas gali pamatyti bet kurį kiekvieno vartotojo sandorį, kadangi jis yra registruojamas viešame decentralizuotame registre.
- **Patikimumas.** Duomenų bazėje suklastoti ar pakeisti sandorį yra praktiškai neįmanoma, nes tokiu atveju pakeitimus reiktų atlikti visiems tinklo vartotojams.
- **Drastiški svyravimai.** Kripto valiutų nepastovumas dažnai pritraukia spekuliacinį susidomėjimą ir investuotojus. Ryškūs kainų svyravimai gali suteikti prekybininkams puikias galimybes užsidirbti pinigų.

Kita vertus, drastiški svyravimai apima ir didesnę riziką. Kripto valiutos taip pat nėra teisiškai reguliuojamos ar prižiūrimos atsakingų institucijų (Kripto valiutos ir žetonų apskaitos rekomendacijos 2018), tai prideda tokioms investicijoms papildomos rizikos. Be to, reikia turėti pakankamai žinių ir suprasti veiklos principus, norint pasirinkti tinkamą kripto valiutą investavimui, kadangi valiutos vertė yra nepastovi.

## 1.2. Kripto valiutų kainoms įtakos turintys veiksniai

Išaugęs kripto valiutų populiarumas paskatino žmones domėtis virtualia valiuta ir tirti jos charakteristikas. Ne tik potencialūs investuotojai, bet ir mokslo srities žmonės bandė nustatyti veiksnius, lemiančius kripto valiutų kainų kitimą. Yra daugybė priežasčių, lemiančių didelį kintamumą ir nestabilumą. Galima išskirti tokius pagrindinius veiksnius (Master The Crypto 2019):

- **Nauja rinka.** Naujoms technologijoms reikia laiko, kad jas patobulintų ir pritaikytų plačiajai auditorijai, ir yra didelė nesėkmės rizika. Rinka paremta nauja technologija yra daug nepastovesnė už tradicines, brandžias ir laiko patikrintas. Kriptovaliutų rinka šiuo metu eina panašiu ciklu, kaip internetinių bendrovių atveju, kai iš pradžių įmonės uždirbo reikšmingą gražą, tačiau pernelyg didelis optimizmas ir spekuliaciniai motyvai išpūtė interneto bendrovių burbulą. O po kurio laiko šis burbulas pradėjo bliuksti.
- **Nereguliuojama rinka.** Kriptovaliutų rinka iš esmės nereguliuojama dėl sudėtingumo ir sunkumų reguliuojant atvirojo kodo decentralizuotą technologiją. Kadangi nėra reglamentų, atsiranda blogų veikėjų, kurie užsiima rinkos manipuliacija.
- **Spekuliacijos.** Naujoje rinkoje, kurioje nėra jokių taisyklių, vienintelis dalykas, lemiantis kriptovaliutų vertes, yra spekuliacija. Paprastai bet kurio turto vertė priklauso nuo jo naudingumo. Šiuo metu spekuliacijos yra paplitusios, nes labai sunku – beveik neįmanoma kiekybiškai įvertinti bet kurią kriptovaliutos vertę, remiantis tradicine fundamentalia analize.

Akademikai aktyviai atlieka įvairius tyrimus, kurių metu siekia ištirti kriptovaliutų rinką ir nustatyti, kokie išorės faktoriai turi įtakos kriptovaliutų kainų pokyčiams. Choi ir Shina'as (2020) pritaikė vektorinės autoregresijos (toliau VaR) modelį ir nustatė, kad bitkoino graža nėra susijusi su tradicinėmis turto klasėmis, tokiomis kaip akcijos, obligacijos ir biržos prekės. Be to, autoriai taip pat nerado reikšmingo sąryšio tarp bitkoino ir aukso kainų.

Dauguma tyrimų sutelkti į trumpalaikius ryšius, naudojant tik dienos duomenis, Konradas, Kustovičius ir Ghysels'as (2018) pabrėžia, kad taip pat svarbu ištirti ryšį tarp ilgalaikio bitkoino kintamumo ir jo ekonominių veiksnių. Tirdami bitkoino kintamumą ilguoju laikotarpiu, autoriai pritaikė dviejų komponentų GARCH-MIDAS metodą. Mišrių duomenų atrinkimo (*angl.* Mixed Data Sampling) technika siūlo unikalią sistemą potencialių bitkoino kintamumo veiksniams ištirti. Makroekonominių ir finansinių elementų duomenys yra imami retesni (kas mėnesį), lyginant su bitkoino gražomis. Rezultatai parodė, kad S&P 500 kintamumas daro neigiamą ir labai reikšmingą poveikį ilgalaikiam bitkoino kintamumui, o S&P 500 kintamumo rizikos premija priešingai – teigiamą. Be to, skirtingai nuo akcijų rinkos kintamumo, bitkoino kintamumas elgiasi procikliškai, t. y., didėja esant aukštesniam pasaulinės ekonominės veiklos lygiui (Conrad, Custovic ir Ghysels 2018).

Kriptovaliutų kainai įtaką daro daugelis įvykių pasaulyje. Vienas svarbiausių momentų 2019-aisiais – bitkoinų ir eterių (*angl.* Ethereum) indeksų įtraukimas į NASDAQ biržą. Tuomet bitkoino kaina padidėjo 300 doleriais. 2019 m. viduryje buvo fiksuota ir kita tendencija: bitkoino kaina smuko 360 dolerių, kai Donaldas Trumpas „Twitter“ programėlėje paskelbė, jog jis nėra bitkoinų ir kitų

kripto valiutų gerbėjas (Finblog 2020). Filipas ir Gorse (2017) parodė, kad diskusijas *Reddit* socialiniame tinkle ir kripto valiutų kainas sieja stiprus ryšys. Remiantis paskelbtų žinučių apimtimi, per dieną padaugėjusių sekėjų skaičiumi ir per dieną į *subreddit* forumą prisijungusių naujų narių skaičiumi autoriai nustatė, kad socialiniai tinklai gali pateikti reikšmingos informacijos kripto valiutų kainos krypties prognozavimui (Phillips ir Gorse 2017).

### 1.3. Kripto valiutų prognozavimo metodai

Siekiant išsiaiškinti, ar socialiniuose tinkluose pasirodanti informacija gali lengvai paveikti kripto valiutų kainas, pradėti kurti įvairūs prognozavimo modeliai.

Jau minėtieji Filipas ir Gorse (2017) savo atliktame tyrime pritaikė paslėptuosius Markovo modelius. Šio darbo tikslas buvo parodyti, kaip metodai naudojami epidemijos protrūkiams identifikuoti gali būti pritaikomi kripto valiutų kainų „burbulų“ prognozei atsižvelgiant į socialinių tinklų informaciją. HMM metodas buvo naudojamas aptikti epidemijos ir ne epidemijos būsenas ne tik tokių kripto valiutų kaip bitkoino, laitkoino (*angl.* Litecoin, LTC), eterio ir monero (*angl.* Monero) kripto valiutų prekybos apimtyse, bet ir socialinio tinklo diskusijų srauto apimtyje. Galima pažymėti, kad prekybos strategijoje nebuvo naudojami jokie su kaina susiję prekybos signalai, ir atsižvelgta tik į socialinių tinklų naudojimą ir prekybos apimtį. Darbe pademonstruota, kad paslėptieji Markovo modeliai yra naudingi nustatant į „burbulus“ panašų elgesį laiko eilutėse, ir kad socialinė žiniasklaidos teikiama informacija yra susijusi kripto valiutų kainų pokyčiais (Phillips ir Gorse 2017).

Kitas tyrimas (Kraaijeveld ir De Smedt 2020) buvo atliktas pasitelkiant *Twitter* duomenis. Žodynu paremta sentimentinė analizė kartu su *Grangerio* priežastinio ryšio testu parodė, kad *Twitter* programėlėje skelbiamos nuomonės gali būti naudojamos bitkoino, bitkoino pinigų (*angl.* Bitcoin Cash) ir laitkoino kainų grąžai nustatyti. O žinučių apimtys gali būti naudojamas kaip laitkoino ir *Ripple* kripto valiutų grąžų prognozavimo rodiklis. Atvirkštinis ryšys siejamas su kitomis kripto valiutomis, kur kainų grąžos padeda prognozuoti žinučių kiekį. Yra keletas atvejų, kai kainų grąža paveikia *Twitter* naujienų srautą dienos eigoje, tačiau autoriai teigia, kad vis dėlto *Twitter* turi didesnę poveikį kripto valiutų rinkai, o ne atvirkščiai. Be to, pritaikius euristinę rinkinį nustatyta, kad 1 – 14% žinučių (*angl.* Tweets) susijusių su kripto valiutomis paskelbia robotai (Kraaijeveld ir De Smedt 2020).

Taip pat buvo atliktas tyrimas (Dipple et. al 2020), kuriame analizavo kelių pagrindinių kripto valiutų (bitkoino, eterio, laitkoino, monero ir *Ripple*) kainų svyravimų ir socialinių tinklų (*Twitter*, *Reddit*, *GizHub*) veiklos koreliaciją. Koreliuotų paklaidų stochastinių diferencialinių lygčių pagalba sukūrė

prognozavimo modelį. Kriptovaliutų kursams naudojo standartinio geometrinio Brauno judesio modeliavimą, o socialinės žiniasklaidos veiklai ir kriptovaliutų prekybos apimtims - geometrinį Ornstein-Uhlenbeck procesą. Atitinkamų kriptovaliutų ir socialinių tinklų modelio parametrų įverčiai rasti didžiausio tikėtimumo būdu. Atsižvelgiant į pastebėtas koreliacijas buvo atliktas laiko eilučių prognozavimas. Pasak autorių gautas įspūdingas tikslumas numatant, ar kriptovaliutų rinka kitą dieną padidės ar sumažės, o tai reikšmingas rezultatas investuojant ir prekiaujant kriptovaliutomis (Dipple et. al 2020).

Šiuose tyrimuose prognozė remiasi išoriniu regresoriumi, tačiau taip pat yra svarbu sudaryti modelius, kurie efektyviai įvertintų pačių kriptovaliutų kainų dinamikos tendencijas ir kuriuos būtų galima naudoti kainų prognozei.

Giameras (2020) sukonstravo modelį naudojant variacinių modų dekompoziciją (*angl.* Variational Mode Decomposition - VMD) ir apibendrintą adityvų modelį (*angl.* Generalized Additive Model – GAM), kuris prognozavo bitkoino vienos dienos kainų seką minutės intervalu. Sukurto modelio (VMD-GAM) efektyvumas naudojant apmokymo duomenis buvo palygintas su empirinės modų dekompozicijos (*angl.* Empirical Mode Dekompozicija - EMD) ir GAM hibridiniu metodu. Empiriniai rezultatai naudojant tris vertinimo metrikas (MAPE, RMSE ir Bias) parodė, kad VMD-GAM modeliu buvo pasiektas geresnis prognozavimo našumas, palyginti su EMD-GAM ansamblio modeliu. Į tolesnius tyrimus autorius rekomenduoja įtraukti neuroninius tinklus (Gyamerah, 2020).

Prognozę neuroninių tinklų metodu atliko Maiti's ir kt (2020). Jie pritaikė neuroninius tinklus septynių kriptovaliutų kiekvienos dienos duomenims. Analizuoti duomenys apie atidarymo kainą, aukščiausią dienos kainą, žemiausią dienos kainą, uždarymo kainą, apyvartą ir rinkos kapitalizaciją nuo 2013 m. balandžio 28 d. iki 2019 m. sausio 23 d. Visų pirma atlikta koreliacinė analizė parodė, kad tarp atidarymo, aukščiausios, žemiausios ir uždarymo kainos yra ryškus tiesinis ryšys. Žemiausia priklausomybe pasižymi apyvartos rodiklis visoms kriptovaliutomis. Nepaisant aukšto tiesinio modelio koreliacijos koeficiento, linijiniai modeliai nėra tinkami chaotiškos prigimties kriptovaliutų prognozei. Neuroniniai tinklai turėjo daug mažesnę paklaidą atliekant skaičiavimus ir prognozę (Maiti, Vyklyuk, ir Vuković 2020). Didelio kintamumo kriptovaliutų kainų laiko eilutės dažnai yra netiesinės ir nestacionariosios, todėl naudojant įprastus statistinius modelius kainų prognozei gaunamos didelės paklaidos.

Kriptovaliutų rinka yra dar visai nauja, todėl akademikai lygina metodų tikslumą pritaikius modelius tiek kriptovaliutų, tiek akcijų kainų prognozei. Sudas ir Spiteris (2019) analizavo bitkoino dienos uždarymo kainą nuo 2016 m. sausio 1d. iki 2019 m. sausio 28 d. Kriptovaliutų rinkos valdymo modeliams pasirinko naudoti paslėptuosius Markovo modelius (*angl.* Hidden Markov Models –

HMM) ir paslėptuosius pusiau Markovo modelius (*angl.* Hidden semi-Markov Model – HSMM). Atlikę tyrimą autoriai padarė išvadą, kad geriausia naudoti keturių būsenų normalųjį HSMM modelį su neigiamu binominiu pasiskirstymu norint efektyviai fiksuoti bulių (*angl.* bull) ir lokių (*angl.* bear) rinkos dinamiką. Bulių rinka laikoma tada, palyginti ilgą laiką kaina kyla aukštyn, o kai kainos stipriai krenta atsižvelgiant į to meto aukštas kainas, laikoma, kad tai yra lokio rinka. Nors gautas modelio pritaikomumas yra prastesnis, nei atlikus modeliavimą S&P 500 akcijų kainai, jis yra tinkamas naudoji kriptovaliutų kainų prognozei (Suda ir Spiteri 2019).

Apibendrinant kriptovaliutų kainų analizei taikomus metodus, galima teigti, kad HMM pritaikymas kriptovaliutoms yra vertas dėmesio ir tolimesnių tyrimų dėl savo universalumo, t. y. jį galima pritaikyti tiek tiesiogiai - stebimoms kainoms, tiek netiesiogiai - per išorinį regresorių, kuris turi įtakos kainų pokyčiams. Sekančiame skyrelyje apžvelgiami HMM taikymo pavyzdžiai.

#### **1.4. Paslėptųjų Markovo modelių taikymų apžvalga**

Paslėptosios Markovo grandinės arba dar vadinamieji paslėptieji Markovo modeliai pradėti naudoti 1970 metais. Pirmiausia HMM buvo pritaikytas kalbos atpažinimui. Vėliau naudotas ne tik garso signalams klasifikuoti ir prognozuoti, bet ir DNA sekų analizei, ranka rašytų simbolių atpažinimui ir t.t. (Hassan and Nath, 2005). Ilgainiui HMM pradėtas taikyti akcijų rinkos analizei bei kainų prognozei.

Lietuvoje HMM finansinėms laiko eilutėms kol kas nėra plačiai taikomas, tačiau nemažai HMM modeliavimo tyrimų atlikta automatiniam šnekamosios kalbos atpažinimui. HMM panaudojimas šnekai atpažinti remiasi prielaida, kad kalbos signalas yra atsitiktinis procesas, kurio parametrus galima nustatyti. HMM visam žodžiui sukurti sujungiami atskirų fonemų<sup>2</sup> (*angl.* phone) HMM, apskaičiuojamos žodžio sekos tikimybės bei surandamas geriausias kelias, atitinkantis optimalią žodžio seką. Atskiriems žodžiams ir fonemoms apmokyta HMM sujungiant yra gaunamas kelių žodžių ar fonemų sekos HMM (Vaičiulytė 2015). Vis tik vėliau atliktame tyrime nustatyta, kad HMM apmokyti trifonemomis<sup>3</sup>, leidžia pasiekti geresnių atpažinimo rezultatų nei apmokant fonemomis. (Vaičiulytė ir Felinskas 2016). Vartotojo identifikavimo naudojant balso atpažinimą HMM modelis savo tikslumu palenkia Bajeso ir neuroninių tinklų klasifikatorius (Barauskaitė 2017).

---

<sup>2</sup> Fonema – mažiausias kalbos vienetas, turintis skiriamąją reikšmę

<sup>3</sup> Trifonema – tai trijų fonemų grupė



Viename Vaičiulytės ir Sakalausko (2017) straipsnyje yra pristatytas rekurentinis paslėptųjų Markovo modelių parametrų įvertinimo algoritmas. Paslėptuosius Markovo modelius modeliavo Gauso skirstiniu, kurių nežinomi įverčiams gauti panaudotas didžiausio tikėtimumo metodu ir klasikiniu EM (*angl.* Expectation-Maximization - EM) algoritmu.

HMM taip pat pritaikytas vairuotojų elgsenai iširti. Artūras Jakštas (2008) sudarė du HMM modelius iš kurių vienas modeliuoja, kad posūkis įveiktas gerai, o kitas, kad blogai. Šio tyrimo tikslas buvo parodyti, kad sistema sugeba identifikuoti ir įspėti vairuotoją dar prieš nelaimei įvykstant, kad jis įvažiuoja į posūkį blogai. Pats modelis apskaičiuoja tikimybes, koks bus posūkio įveikimas (geras ar blogas).

Tačiau Lietuvoje atliktų ir laisvai prieinamų tyrimų, kuriuose HMM būtų pritaikytas finansinėms laiko eilutėms, surasti nepavyko, todėl tyrimui paremti naudojami užsienio literatūros šaltiniai, kuriuose akcijų ir kriptovaliutų kainoms prognozuoti buvo naudojamas analizuojamas modelis. Pastaraisiais metais paslėptieji Markovo modeliai tampa vis populiariesni modeliuojant finansines laiko eilutes. Viena pagrindinių šių modelių idėjų yra ta, kad tam tikra rinkos informacija yra paslėpta stebint triukšmą turinčiuose stebėjimuose, pvz. kainų pokyčiuose. Pagrindinė paslėpta informacija gali simbolizuoti įvairius verslo ciklo etapus, tokius kaip kilimas, pikas, nuosmukis, mažiausia vertė ir atsigavimas, kurie daro įtaką kainų pokyčiams (Erlwein, Mitra ir Roman 2012). Paslėptosios arba nežinomos Markovo grandinės dalys neretai gali paaiškinti atsiradusias paklaidas ir procesų vidinę struktūrą.

Vienas iš pavyzdžių, kad HMM modelis yra tinkamas prognozuoti akcijų kainą yra Hasano ir Nath'o (2005) atliktas tyrimas. Autoriai analizavo duomenis pagal keturis akcijų požymius: pradinę kainą, didžiausią kainą, mažiausią kainą ir uždarymo kainą. Šie keturi ankstesnės dienos atributai buvo naudojami numatyti sekančios dienos uždarymo kainą. Tyrimas atliktas tokiu principu, kad yra randami tokie istoriniai duomenys, kurių dinamika (*angl.* pattern) yra panaši į dabartinius pokyčius, ir tada panaudoja istorinių duomenų paskutinės dienos kainos pokytį ir dabartinę dienos kainą, kad galėtų numatyti būsimą uždarymo kainą. Autoriai palygino pasiūlyto metodo ir ANN rezultatus vidutinės absoliutinės paklaidos (MAPE) pagalba ir gavo, kad vertės yra gana panašios (Hassan ir Nath 2005).

Šį metodą modifikavo Nguyen'as (2017) ir į modelį įtraukė informacinius kriterijus (AIC, BIC ir kt.), kad surastų optimalų būsenų skaičių su kurio HMM yra efektyviausias. Pats modelis apribotas iki šešių būsenų, kad būtų paprastas ir įgyvendinamas esant cikliškiems ekonomikos režimams. Sekančiu žingsniu parinktas geriausias HMM modelis ir sudėtiniai (*angl.* multiple) nepriklausomi kintamieji: atidarymo, žemiausia, aukščiausia ir uždarymo kaina, kad būtų nustatyta būsima JAV indekso S&P

500 uždarymo kaina. Patikimumas įvertintas lyginant HMM rezultatus su vidutiniais istoriniais duomenimis panaudojant R kvadrato (*angl.* R-Squared) ir suminės kvadratinė prognozavimo paklaidos rodiklius.

HMM naudotas prognozuoti *Oil India ltd* akcijų kainai. 7 būsenoms nusakyti sudaryti trys stebėjimai: akcijų kainos kilimas (*angl.* increasing), nekintanti akcijos kaina ir akcijų kainos kritimas (*angl.* decreasing). Perėjimo ir emisijos matricos buvo sudarytos apskaičiuojant vienos, dviejų ir trijų dienų kainų skirtumus. Iš stebėjimų ir būsenų sekų buvo prognozuotos akcijų kainų reikšmės. Išvadose rašoma, kad šis modelis nėra tobulas prognozuoti akcijos kainoms, dėl rinkos svyravimų, kuriuos lemia daugybės faktorių įtaka (Sasikumar ir Abdullah 2016). Šiame straipsnyje nėra atliktas modelio patikimumo tyrimas, todėl lieka neaišku, kiek sudarytas modelis yra efektyvus.

Panašiu į pastarąjį principu Kavitas ir kt. (2013) pritaikė paslėptuosius Markovo modelius akcijų rinkos trendo analizei. Savo darbe analizavo *IBM* akcijų kiekvienos dienos uždarymo kainas mėnesio laiko periodu. Pasirinkti du stebimi simboliai I ir D, atitinkamai akcijų kainų kilimui ir kritimui pažymėti. Buvo spėjama, kad procesą sudaro 6 būsenos ir apskaičiuoti akcijų uždarymo kainų skirtumai vienai – šešioms dienoms. Atsitiktinai sugeneruoti stebėjimų simbolių (*angl.* emission symbols) ir būsenų sekos. Remiantis tikslo (*angl.* fitness) funkcijos didžiausia reikšme nustatyta, kad geresniam rezultatui gauti optimalu naudoti vienos dienos skirtumų seką (Kavitha, Udhayakumar ir Nagarajan 2013).

Erlwein'as, Mitras ir Romanas (2012) modeliavo finansines laiko eilutes naudodamiesi HMM metodu, pagal kurį stebimo proceso parametrai (laikyti diskretizuotais geometriniais Browno judesiais) gali varijuoti tarp skirtingų ekonominių režimų, kurie nėra tiesiogiai stebimi. Paslėpto Markovo modelio parametrai įvertinami naudojant filtravimo metodą. Atlikus parametrų ir Markovo grandinės įvertinimus sekančiam laikotarpiui nustatyta, kad modelis gali būti naudojamas generuoti ateities realizacijų scenarijus finansinėms laiko eilutėms. Sugeneruoti scenarijai *FTSE100* ir aukso būsimoms grąžoms bei patikrinta jų kokybė atsižvelgiant į portfelio optimizavimo uždavinį, kuriam išspręsti naudojamas *vidutinis-CVaR* metodas, t. y. portfelio *CVaR* yra sumažinamas iki minimumo, kai reikalaujama minimalios numatomos grąžos iš portfelio. Galiausiai buvo padaryta išvada, kad siūlomas scenarijų generavimo metodas yra pakankamai stabilus (Erlwein, Mitras ir Romanas 2012).

Be tradicinio HMM modeliavimo, taip pat bandoma atrasti įvairių modelių modifikacijų pagerinančių prognozės rezultatus. Gupta's ir Dhingra'as (2012) naudodamiesi istorinėmis akcijų kainomis pritaikė maksimalų posteriorinį HMM metodą sekančios dienos akcijų kainos prognozavimui. Apmokydami tolydųjį HMM jie atsižvelgė ne tik į aukščiausią ir žemiausią akcijų vertę dienos metu, bet ir dalinius akcijų vertės pokyčius. Šis HMM naudotas atlikti maksimalius posteriorinius sprendimus visoms

galimoms sekančios dienos akcijų kainoms. Modelis pritaikytas keturių įmonių akcijų kainoms ir gauti rezultatai palyginti su kai kuriais esamais metodais, tokiais kaip HMM, ARIMA ir dirbtiniais neuroniniais tinklais. Paklaidoms įvertinti pasirinktas vidutinės absoliutinės procentinės paklaidos MAPE rodiklis. Pasiūlytas HMM modelis buvo pranašesnis *Apple Inc.* ir *IBM Corporation* akcijų kainų prognozės atveju, o *Dell Inc.* kainų prognozės efektyvumas apylygis. Be to, autoriai išskyrė HMM modelio paprastumą, palyginti su kitais esamais metodais (Gupta ir Dhingra 2012).

Apibendrinant HMM modeliavimo rezultatus, galima daryti tokias išvadas, kad finansinėms laiko eilutėms reiktų rinktis ne daugiau 7 paslėptųjų būsenų ir kad modelis efektyviausiai imant vienos dienos duomenų skirtumus. Šiomis išvalgomis remtasi pritaikant HMM modelį kriptovaliutų kainoms.

#### **1.4.1. Paslėptųjų Markovo grandinių taikymas režimų identifikavimui**

Paslėptieji Markovo modeliai taip pat gali būti naudojamas identifikuojant režimų perėjimus (*angl.* regime switching). Toks pritaikymas skirtas suprasti akcijų ar kriptovaliutų rinkos elgesį.

Paprastai yra nežinoma, kiek laiko truks vienas ar kitas režimas. Angas ir Timermanas (2012) teigė, kad režimo perėjimo modeliai gali fiksuoti finansinio turto kainos dinamikos pokyčius. Bene svarbiausias šio modelio aspektas yra išsiaiškinti optimalų būsenų skaičių modeliuojant režimo perėjimo modelius.

Geriausias modelis įprastai pasirenkamas atsižvelgiant į informacinių kriterijų reikšmes. Labiausiai paplitęs AIC ir BIC kriterijų naudojama informacija, skirta maksimaliai padidintų tikėtinumo funkciją. Dažniausiai pasirenkamas paslėptųjų būsenų skaičius kinta nuo 2 iki 7. Tačiau su kiekviena papildoma būsena padidėja perėjimo tikimybių ir stebėjimo pasirodymo tikimybių skaičiavimų sudėtingumas. Pavyzdžiui, pasirenkant 6 ar 7 būsenas galima modelį tiesiog permokinti (Chappell 2018).

Čapelis (2018) pritaikė HMM ir atliko optimalaus būsenų skaičiaus tyrimą bitkoino kainų duomenims. Autorius nustatė, kad labiausiai tinkamas yra 5 būsenų paslėptasis Markovo modelis. Be to pritaikius režimo perėjimo analizę Markovo modeliais su skirtingu būsenų skaičiumi užfiksavo, kad bitkoino kainoms būdingi kintamumo šuoliai, kai yra pasirinktas didesnis paslėptųjų būsenų skaičius. O kai yra pasirinktas 2, 3 arba 4 būsenų modelis režimai yra suklasterizuojami pagal kainos kintamumą (Chappell 2018).

Analizuojant akcijų kainas jos yra siejamos su bulių ir lokių režimais, kai turimas 2 būsenų Markovo modelis, o kai svarstomas 3 būsenų modelis įtraukiama ir neutrali rinka. Esant daugiau paslėptųjų būsenų interpretuojant duomenis atsižvelgiama į režimų vidurkių ir standartinių nuokrypių reikšmes. *FTSE 100* indeksui atliktas tyrimas parodė, kad geriausiai realius duomenis atitinka 2 būsenų modelis. Tuo tarpu 3 ir 4 būsenų modeliai atmesti, kaip nepakankamai tikslūs (Yuan ir Mitra 2016) .

Taigi, šis analizės tipas naudingas norint išsiaiškinti rinkos poveikį analizuojamoms kainoms.

### **1.5. Atliekamo tyrimo metodika**

Remiantis atlikta literatūros analize tiriamajame darbe paslėptasis Markovo modelis pritaikomas nustatyti paslėptuosius rinkos režimus ir jų poveikį kriptovaliutų kainų grąžoms. O kainų krypties prognozei atlikti sudarytas diskretusis HMM modelis. Šiems tikslams pasiekti išsikelti uždaviniai pateikti išsamiau nurodant taikytą metodologiją:

- apskaičiuojamos kriptovaliutų grąžos ir atliekama jų statistinė analizė;
- nustatomas optimalų paslėptųjų būsenų skaičių remiantis AIC ir BIC informaciniais kriterijais;
- pritaikomas 3 būsenų HMM nustatyti paslėptųjų režimų poveikį kriptovaliutų kainoms;
- sudarytas HMM modelis kriptovaliutų kainų grąžas suskirsčius į 5 stebėjimų intervalus ir darant prielaidą, kad egzistuoja 2 paslėptosios būsenos nusakančios kainos kryptį;
- pritaikomas *Baum-Welch* algoritmas modelio parametrus nustatyti;
- naudojant *Viterbi* algoritmą prognozuota kainos kryptis pagal sudarytą stebėjimų seką;
- įvertinamas modelio tikslumas apskaičiuojant sumaišymo matricą;
- prognozės tikslumas palyginamas su mašininio mokymosi metodais.

## 2. Tyrimo metodologija

Šiame skyriuje pateiktas paslėptųjų Markovo modelių apibrėžimas, aprašyti tyrimo metu naudojamų algoritmų veikimo principai ir modelio patikimumo vertinimo kriterijai bei pateikiama tyrimo schema.

### 2.1. Paslėptieji Markovo modeliai

HMM yra tikimybinis modelis, kuris yra paremtas Markovo grandinių procesu. Markovo grandinė yra naudinga, kai turime apskaičiuoti stebimų įvykių sekos tikimybę. Markovo grandinių kiekviena būseną atitinka kažkokį stebimą, realų įvykį. Tačiau daugeliu atvejų dominantys įvykiai yra paslėpti: mes jų tiesiogiai nestebime. Dėl tos priežasties Markovo grandinių modelis yra ribotas daugelio uždavinių sprendimo atžvilgiu. Išplėtus Markovo grandinių sampratą, kai stebėjimas yra tikimybinių būsenų funkcija, turimas vadinamasis paslėptųjų Markovo grandinių modelis.

Taigi, HMM yra stochastinis procesas, kurio pagrindinis procesas negali būti tiesiogiai stebimas, nes jo būsenos yra paslėptosios. Toks procesas gali būti pastebėtas kito stochastinio proceso turinčio stebėjimų seką. Stebimas reiškinys (*angl.* the observed event) dar yra vadinamas simboliu, o nežinomas faktorius sudarantis stebėjimo pagrindą – būseną (Sasikumar ir Abdullah 2016).

HMM privalumai (Hassan ir Nath 2005):

- HMM turi stiprius statistikos pagrindus
- Sugeba patikimai apdoroti naujus duomenis
- Skaičiavimai atliekami efektyviai tiek programuojant, tiek įvertinant modelį (dėl nustatytų apmokymo algoritmų)
- Galima nesunkiai prognozuoti esant panašioms dėsningumams

Pagrindinės problemos su kuriomis susiduria HMM yra įvertinimas (*angl.* Evaluation), iššifravimas (*angl.* Decoding) ir apmokymas (*angl.* Learning). Įvertinimas yra susijęs su modelio atpažinimu nustatant stebėjimų sekos tikimybę atsižvelgiant į duotąjį HMM, iššifravimas taikomas surasti labiausiai tikėtiną paslėptųjų būsenų seką, kuri būtų sugeneruota iš stebėjimų sekos. Apmokymo problema yra HMM parametrų generavimas atsižvelgiant į stebėjimų seką (Kavitha, Udhayakumar ir Nagarajan 2013).

### 2.1.1. Paslėptą Markovo modelio apibrėžimas

Paslėptąjį Markovo modelį (HMM) galima apibrėžti kaip tam tikrą dvigubą stochastinį procesą, turintį du aspektus (Kouemou 2011):

- Pirmasis stochastinis procesas yra baigtinis būsenų rinkinys, kur kiekviena iš būsenų paprastai yra susijusi su tikimybinu pasiskirstymu. Perėjimai tarp skirtingų būsenų statistiškai apibrėžiami pagal tikimybinį rinkinį, vadinamą perėjimo tikimybėmis.
- Antrame stochastiniame procese stebimi įvykiai, kurie pasirodo esant bet kurioje iš būsenų. Analizuojama tik tai, kas yra stebima, nežinant, kuriose būsenose įvykis pasirodė, nes būsenos yra „paslėptos“ stebėtojų, todėl modelis ir pavadintas „Paslėptuoju Markovo modeliu“.

HMM naudojamos charakteristikos ir jų žymėjimai pateikti 1 lentelėje.

**1 lentelė.** Pagrindinės HMM charakteristikos

Charakteristika	Žymėjimas
Stebėjimų skaičius	$T$
Būsenų skaičius	$N$
Būsenų simbolių skaičius	$M$
Stebėjimų seka	$O = o_1, o_2, \dots, o_T$
Būsenų seka	$X = x_1, x_2, \dots, x_T$
Galimų būsenų aibė	$S = \{s_1, s_2, \dots, s_N\}$
Galimi būsenų simboliai	$v_k = \{v_1, v_2, \dots, v_M\}$
Būsenų perėjimo tikimybių matrica	$A = a_{11} \dots a_{ij} \dots a_{NN}$
Simbolių pasirodymo tikimybės matrica	$B = b_j(k)$
Pradinės būsenos tikimybė	$\pi = \pi_1 \pi_2 \dots \pi_N$

Būsenų perėjimo tikimybių matricos  $A$  elementas  $a_{ij}$  apibrėžia tikimybę, kad laiko momentu  $t + 1$  būseną bus lygi  $S_j$ , kai laiko momentu  $t$  būseną yra lygi  $S_i$ :

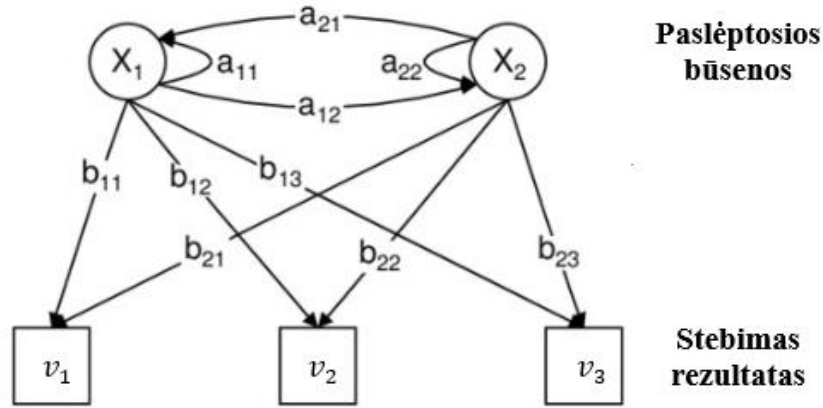
$$a_{ij} = P(x_{t+1} = S_j | x_t = S_i), \quad 1 \leq i, j \leq N. \quad (2.1)$$

Perėjimo tikimybės turėtų tenkinti įprastus stochastiniai apribojimus:  $a_{ij} \geq 0, \quad 1 \leq i, j \leq N$  ir  $\sum_{j=1}^N a_{ij} = 1, \quad 1 \leq i \leq N$ .

Stebėjimo simbolio išvesties tikimybės pasiskirstymas kiekvienoje būsenoje  $B = b_j(k)$ , kur  $b_j(k)$  yra tikimybė, kad simbolis  $v_k$  pasirodė būsenoje  $S_j$ .

$$b_j(k) = P(o_t = v_k | x_t = S_j), \quad 1 \leq j \leq N, \quad 1 \leq k \leq M. \quad (2.2)$$

Šie stochastiniai apribojimai taip pat turi būti tenkinami:  $b_j(k) \geq 0$ ,  $1 \leq j \leq N$ ,  $1 \leq k \leq M$  ir  $\sum_{k=1}^M b_j(k) = 1$ ,  $1 \leq j \leq N$ .



**1 pav.** Paslėptojo Markovo modelio dviejų būsenų perėjimo schema, kai turimi trys stebėjimų simboliai (Toledo ir Katz 2009)

Modelio pradinių būsenų pasiskirstymas  $\pi = \{\pi_i\}$ , kur  $\pi_i$  yra tikimybė, kad modelis yra būsenoje  $S_i$  pradiniu laiko momentu  $t = 0$ , lygi

$$\pi_i = P(x_1 = S_i), \quad 1 \leq i \leq N. \quad (2.3)$$

Būsenų perėjimo tikimybių matrica  $A$ , išvesties tikimybių matrica  $B$  ir pradinės būsenos vektorius  $\pi$  sudaro HMM modelį, kuris apibendrintai žymimas:

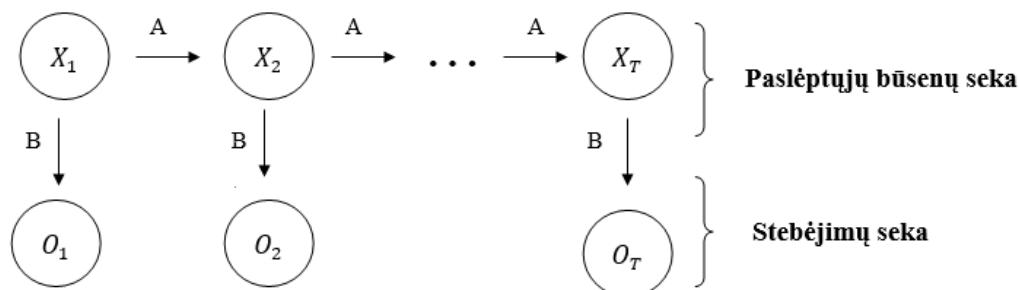
$$\lambda = \{A, B, \pi\}. \quad (2.4)$$

Jeigu yra daroma prielaida, kad stebėjimo pasirodymo tikimybės pasiskirsčiusios pagal Gauso skirstinį, tada turimas tolydusis HMM modelis, kur  $b_j(k) = b_i(o_t = v_k) = N(v_k, \mu_i, \sigma_i)$ , čia  $\mu_i$  ir  $\sigma_i$  yra pasiskirstymo vidurkis ir standartinis nuokrypis atitinkantis būseną  $S_i$ , o  $N$  – Gauso tankio funkcija. Perrašome  $b_i(o_t = v_k) = b_i(o_t)$ , tada HMM modelį apibrėžia tokie parametrai:

$$\lambda = \{A, \mu, \sigma, \pi\}. \quad (2.5)$$

HMM yra tikimybinis modelis, kuris įvertina dabartinę būseną remiantis prielaida, kad tik praėjusi būseną gali paveikti dabartinę būseną. Svarbu, kad modelyje kiekviena būseną seka Markovo grandines, tik tiek, kad jos yra paslėptos. Tai pavaizduota 2 paveikslėlyje, kur HMM yra modelis sudarytas iš dviejų elementų: paslėptųjų būsenų reikšmių ( $X$ ) sekos ir stebėjimų reikšmių ( $O$ ) sekos

(Kim, Jeong ir Lee 2019). Kiekviena būseną  $X_t$  priklauso tik nuo prieš tai buvusios būsenos  $X_{t-1}$ , o stebėjimas  $O_t$  priklauso tik nuo dabartinės būsenos  $X_t$ .



**2 pav.** Paslėptojo Markovo modelio bendroji struktūra (Kim, Jeong ir Lee 2019)

Pagrindinės prielaidos, kurias turi tenkinti modelis (Nguyen 2018):

1. Stebėjimas  $O_t$  sugeneruojamas pagal paslėptąją būseną (arba rinkos režimą) laiko momentu  $t$
2. Paslėptųjų būsenų skaičius yra baigtinis ir tenkina pirmos eilės Markovo sąlygą.
3. Perėjimo tarp būsenų tikimybių matrica yra pastovi.
4. HMM modelio stebėjimas laiko momentu  $t$  turi tam tikrą tikimybinį pasiskirstymą, kuris atitinka galimą paslėptą būseną.

Trys pagrindinės HMM problemos (Kim, Jeong ir Lee 2019; Nguyen 2017):

**Įvertinimo problema.** Turint stebėjimų seką  $O = o_1, o_2, \dots, o_T$  ir HMM modelio parametrus  $\lambda = \{A, B, \pi\}$ , nustatyti  $P(O|\lambda)$  tikimybę. Apskaičiuotos stebėjimo sekos tikimybės padeda įvertinti kaip gerai duotasis modelis atitinka stebėjimų seką. Stebėjimų tikimybėms rasti naudojami pirmyn (*angl.* forward) arba atgal (*angl.* backward) algoritmas.

**Iššifravimo problema.** Turint stebėjimų seką  $O = o_1, o_2, \dots, o_T$  ir HMM modelio parametrus  $\lambda = \{A, B, \pi\}$ , rasti būsenų seką  $S = s_1, s_2, \dots, s_T$  geriausiai tinkančią stebėjimų sekai. *Viterbi* algoritmo pagalba yra nustatoma optimali būsenų seka duotajai stebėjimų sekai.

**Apmokymo problema.** Turint stebėjimų seką  $O = o_1, o_2, \dots, o_T$ , sugeneruoti modelio parametrus  $\{A, B, \pi\}$ , kad būtų maksimizuota  $P(O|\lambda)$  tikimybė. Parametrus rasti naudojamas *Baum-Welch* algoritmas.



## 2.2. Pirmyn ir atgal algoritmai

Stebėjimo sekos tikimybes galima apskaičiuoti naudojant paprastus tikimybinius argumentus, tačiau jis nėra praktiškas, nes skaičiavimo sudėtingumas yra  $N^T$  eilės (Kouemou 2011). Dėl tos priežasties naudojamas kitas skaičiavimo metodas, kuriame naudojamas pagalbinis kintamasis:

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, x_t = i | \lambda), \quad (2.6)$$

$\alpha_t(i)$  yra vadinamas „pirmyn“ kintamasis, o  $o_1, o_2, \dots, o_t$  yra dalinė stebėjimų seka.

Šis kintamasis gali būti perrašytas rekurentiniu pavidalu:

$$\alpha_{t+1}(j) = b_j(o_{t+1}) \sum_{i=1}^N \alpha_t(i) a_{ij}, \quad 1 \leq j \leq N, \quad 1 \leq t \leq T-1, \quad (2.7)$$

kai  $\alpha_1(j) = \pi_j b_j(o_1)$ ,  $1 \leq j \leq N$ .

Remiantis (2.7) rekurentine formule galima apskaičiuoti  $\alpha_T(i)$ ,  $1 \leq j \leq N$ . Taigi, ieškoma stebėjimo sekos tikimybę galima užrašyti tokia forma:

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i). \quad (2.8)$$

Šis metodas yra dar žinomas kaip pirmyn algoritmas.

Kintamasis „atgal“ yra apibrėžiamas panašiai:

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \dots, o_T, x_t = i | \lambda). \quad (2.9)$$

Jeigu dabartinė būseną yra  $i$ , tai  $\beta_t(i)$  yra stebėjimų dalinės sekos  $o_{t+1}, o_{t+2}, \dots, o_T$  tikimybė.

$\beta_t(i)$  taip pat gali būti perskaičiuojamas rekurentiniu būdu:

$$\beta_t(i) = \sum_{j=1}^N \beta_{t+1}(j) a_{ij} b_j(o_{t+1}), \quad 1 \leq j \leq N, \quad 1 \leq t \leq T-1, \quad (2.10)$$

kur  $\beta_T(i) = 1$ ,  $1 \leq i \leq N$ .

Galiausiai išraiška užrašoma taip:

$$P(O|\lambda) = \sum_{i=1}^N \pi_j \beta_0(i). \quad (2.11)$$

Taigi, yra du būdai  $P(O|\lambda)$  tikimybei apskaičiuoti, panaudojant kintamąjį „pirmyn“ arba „atgal“.

### 2.3. Viterbi algoritmas

*Viterbi* algoritmas naudojamas, norint surasti visą būsenos seką su maksimalia tikimybe.

Apibrėžiamas pagalbiniis kintamasis, kuris parodo didžiausią tikimybę, kurią gali turėti stebėjimų dalinė seka ir būsenų seka iki laiko momento  $t$ , atsižvelgiant į dabartinę būseną  $i$ .

$$\delta_t(i) = \max_{x_1, x_2, \dots, x_{t-1}} P(x_1, x_2, \dots, x_{t-1}, x_t = i, o_1, o_2, \dots, o_t | \lambda). \quad (2.12)$$

Iš to seka, kad

$$\delta_{t+1}(j) = b_j(o_{t+1}) \left[ \max_{1 \leq i \leq N} \delta_t(i) a_{ij} \right], 1 \leq j \leq N, 1 \leq t \leq T - 1, \quad (2.13)$$

kur  $\delta_1(j) = \pi_j b_j(o_1)$ ,  $1 \leq j \leq N$ .

Taigi, norint apskaičiuoti labiausiai tikėtiną būsenų seką, pradedama skaičiuoti nuo  $\delta_T(j)$ ,  $1 \leq j \leq N$ . Ieškant maksimumo kiekvienoje operacijoje išimenuama „laiminčioji būseną“. Gaunamas rezultatas  $j^*$ , kur  $j^* = \arg \max_{1 \leq i \leq N} \delta_T(i)$ . Pradedant nuo šios būsenos atkurama būsenų seka pagal išimintas būsenų reikšmes. Taip gaunamas ieškomas būsenų rinkinys.

### 2.4. Baum-Welch algoritmas

*Baum–Welch* algoritmas yra išsamus įvertinimo metodas, nes modelio parametrai optimizuojami remiantis didžiausio tikėtimumo metodu. *Baum–Welch* algoritmas yra atskiras EM (*angl.* Expectation-Maximization) pavyzdys.

Turint stebėjimų seką  $O$ , *Baum–Welch* algoritmas iteracinių būdų apskaičiuoja HMM parametru reikšmes  $\lambda = \{A, B, \pi\}$ . Uždavinį galima performuluoti taip, kad algoritmas randa tokius parametrus, kurie maksimizuoja tikimybę  $\arg \max P(O|\lambda)$ .

Kad būtų surastas optimalus modelis, *Baum-Welch* metodas naudoja pirmyn ir atgal algoritmą, todėl šis metodas kartais dar vadinamas pirmyn-atgal algoritmu. Šis algoritmas naudojamas tam, kad sukurtų naują įverčių modelį  $\bar{\lambda} = \{\bar{A}, \bar{B}, \bar{\pi}\}$ . Vietoje  $\lambda$  naudojant įvertį  $\bar{\lambda}$  stebėjimų tikimybė gauta iš modelio gali padidėti iki lokalaus maksimumo. Matematiškai užrašoma taip:

$$\max_{\bar{\lambda}} [X(\lambda, \bar{\lambda})] = P(O|\bar{\lambda}) \geq P(O|\lambda), \quad (2.14)$$

kur tikėtumo funkcija artėja iki kritinio taško.

Algoritmas pradedamas nuo pradinio modelio  $\lambda$ , kuriam yra paduodama stebėjimų sekos imtis skirta apmokymui. Tada modelis įvertina kiekvieną modelio parametro įvertį ir pakeičia parametrus į tuos, kurie maksimaliai padidina apmokyme naudotos stebėjimo sekos tikimybių vertes. Kartojant šį procesą tikimasi susitarti optimalias modelio parametrų vertes (Li 2016).

## 2.5. Modelio patikimumo vertinimas

Skyrelyje pateikiami kriterijai, kurie naudoti modelio tikslumui ir patikimumui įvertinti.

### 2.5.1. Sumaišymo matrica

HMM modelio kainų krypties prognozės tikslumui ir korektiškumui įvertinti naudota sumaišymo matrica (*angl.* confusion matrix). Ši matrica parodo, kiek teisingų ir kiek klaidingų spėjimų atlikta bei galima matyti klaidų pasiskirstymą kiekvienai klasei.

		Tikrosios klasės	
		P	N
Prognozuotos klasės	P	TP	FP
	N	FN	TN

3 pav. Sumaišymo matrica

Sumaišymo matrica pavaizduota 3 paveiksle, kur P žymi teigiamą, N – neigiamą, TP – teisingą teigiamą, FP – klaidingą neigiamą, TN – teisingą neigiamą, o FN – klaidingą neigiamą prognozės klasę.

Be sumaišymo matricos naudojami ir klasifikatoriaus kokybės nustatymo matai, kurie yra paremti sumaišymo matrica (Stehman 1997).

Taiklumo (*angl.* accuracy) rodiklis parodo bendrą modelio tikslumą apskaičiuojant teisingai klasifikuotų duomenų ir visų duomenų santykį. Šis matas yra vengtinas, kai teigiamų ir neigiamų klasių imtys nėra apylygės.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}. \quad (2.15)$$

*Kappa* (*angl.* Cohen's Kappa) rodiklis yra klasifikatoriaus atlikto darbo matas. *Kappa* matuoja duomenų vertės procentinę dalį pagrindinėje lentelės įstrižainėje ir tada koreguoja šias vertes pagal susitarimo kiekį, kurio galima tikėtis vien dėl atsitiktinumo. Kitaip tariant, modelis turės aukštą *Kappa* balą, jei bus didelis skirtumas tarp tikslumo ir nulinio klaidų lygio.

$$Kappa = \frac{Accuracy - Accuracy_{expected}}{1 - Accuracy_{expected}}. \quad (2.16)$$

Jautrumo (*angl.* sensitivity) rodiklis arba dar gali būti vadinamas atkūrimo (*angl.* Recall) rodikliu. Jis taip pat yra skirtas metodo tikslumui išmatuoti, tačiau parodo tik teisingai identifikuotų tikrųjų teigiamų klasifikavimo reikšmių dalį.

$$Sensitivity = \frac{TP}{TP + FN}. \quad (2.17)$$

Specifiškumo (*angl.* specificity) rodiklis yra prieš tai aprašyto jautrumo rodiklio atvirkštinis dydis ir gaunamas apskaičiuojama, kiek tikrųjų neigiamų reikšmių buvo teisingai prognozuota.

$$Specivisity = \frac{TN}{TN + FP}. \quad (2.18)$$

Tikslumo (*angl.* precision) rodiklis parodo tikrųjų teigiamų reikšmių dalį iš visų prognozuotų teigiamų reikšmių.

$$Precision = \frac{TP}{TP + FP}. \quad (2.19)$$

F1 įvertis (*angl.* F1 score) skirtas apibendrinti tikslumo ir atkūrimo rodikliams. Apskaičiavimui naudojamas svertinis šių rodiklių vidurkis. Šis matas leidžia objektyviau palyginti kelis prognozės modelius tarpusavyje.

$$F1\ score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (2.20)$$

### 2.5.2. Pirmos eilės Markovo grandinės testas

Paslėptųjų būsenų seka turi tenkinti pirmos eilės Markovo grandinės savybę.

Diskrečių atsitiktinių dydžių seka  $\{X_t: t \in T\}$  vadinama pirmos eilės Markovo grandine, jeigu visiems  $t \in T$  galioja Markovo savybė:

$$P(X_{t+1}|X_t, X_{t-1}, \dots, X_1) = P(X_{t+1}|X_t), \quad (2.21)$$

čia  $\{X_t\}$  yra būsenų seka.

Jeigu visiems  $t \in T, k \geq 1$  galioja tokia Markovo savybė:

$$P(X_{t+1}|X_t, X_{t-1}, \dots, X_1) = P(X_{t+1}|X_t, X_{t-1}, \dots, X_{t-n+1}), \quad (2.22)$$

Tada sakoma, kad diskrečių atsitiktinių dydžių seka  $\{X_t: t \in T\}$  yra  $n$ -tosios eilės Markovo grandinė (Ky ir Tuyen 2018).

Markovo grandinė vadinama homogeninė, jei perėjimo tikimybės nepriklauso nuo laiko, t. y.,

$$a_{i,j}^{t,t+1} = a_{i,j} = P(X_{t+1} = j | X_t = i), \quad i, j \in N, \quad t \in T. \quad (2.23)$$

Kad patikrinti, ar Markovo grandinė yra pirmos eilės, reikia atlikti du testus. Seka turi būti nepriklausoma, o atsitiktinio dydžio reikšmių tikimybių pasiskirstymas turi būti tolygusis.

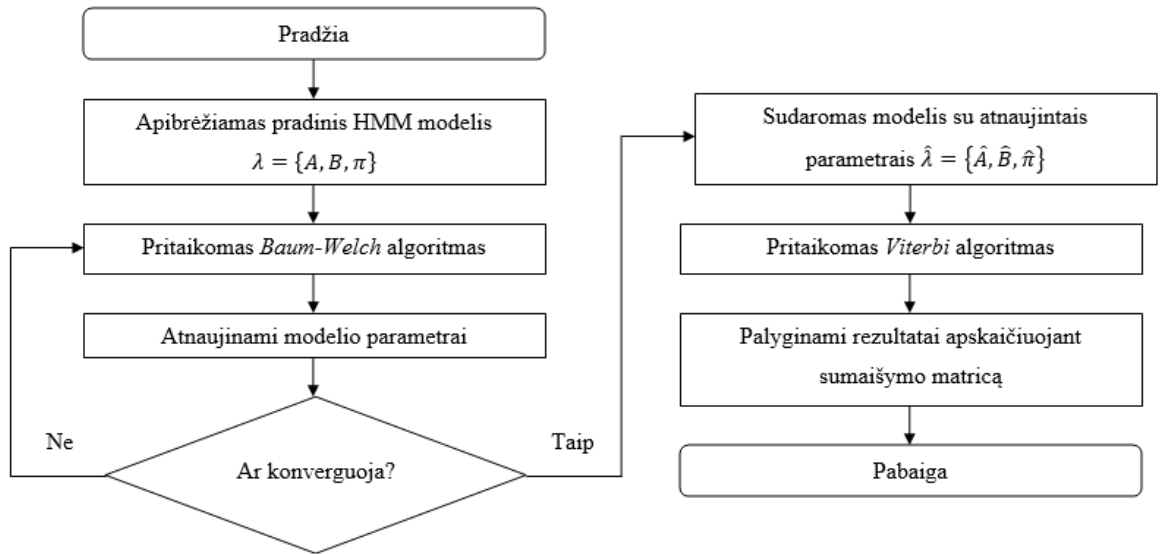
## 2.6. HMM pritaikymo kriptovaliutų kainoms tyrimo eiga

HMM pritaikymas duomenims apie kriptovaliutų kainas atliktas taikant tokią tyrimo eigą:

- atliekama statistinė analizė kriptovaliutų logaritminėms grąžoms;
- nustatomas optimalus paslėptųjų būsenų skaičius;
- pritaikomas 3 būsenų paslėptasis Markovo modelis režimų nustatymui;
- sudarytas paslėptasis Markovo modelis kriptovaliutų kainų grąžas suskirsčius į 5 stebėjimų intervalus ir darant prielaidą, kad egzistuoja 2 paslėptosios būsenos nusakančios kainos kryptį;
- pritaikomas *Baum-Welch* algoritmas modelio parametrų nustatymui;

- naudojant *Viterbi* algoritmą prognozuotos kainos kryptys;
- rezultatai palyginami su realiomis kainų kryptimis apskaičiuojant sumaišymo matricą
- modelio tikslumas palygintas su kitais klasterizavimui naudojamais algoritmais.

Kainos krypties prognozės procesas taip pat pavaizduotas schematiškai (žr. 4 pav.).



4 pav. HMM modeliavimo schema

Šiems uždaviniams atlikti pasirinkta skaičiavimus atlikti su R programa. Šį pasirinkimą nulėmė tai, kad programa yra laisvai prieinama bei turi gausybę naudingų paketų, kurie leidžia supaprastinti duomenų analizės procesą.

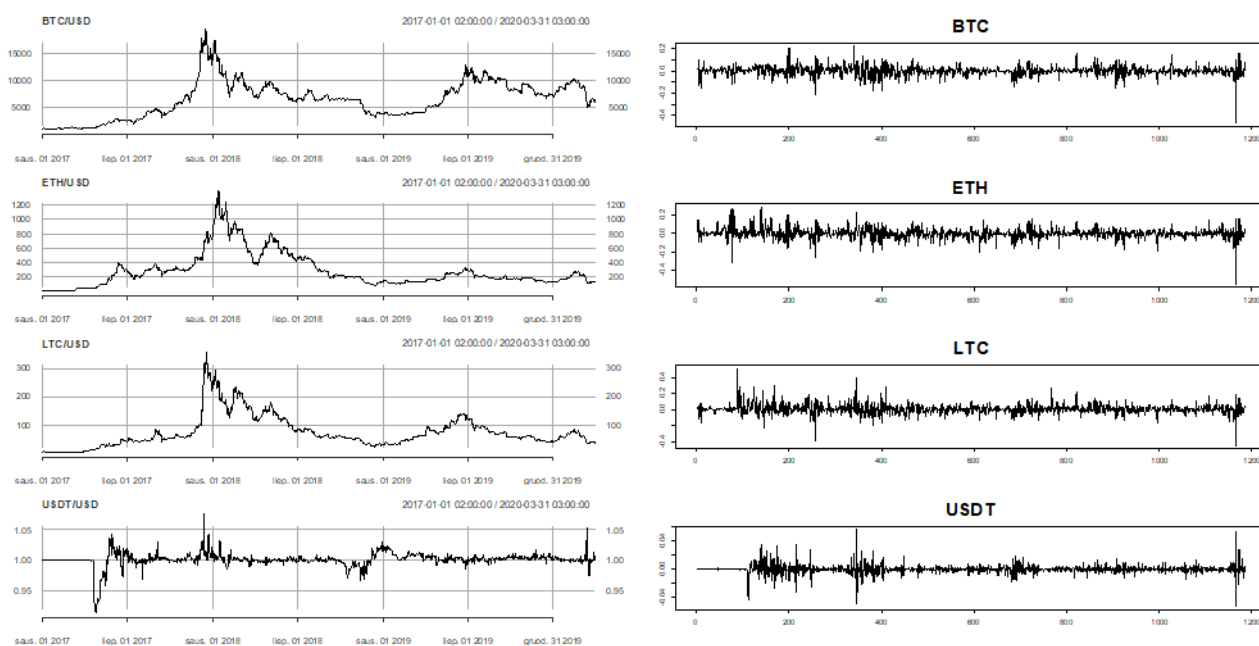
### 3. Tyrimo rezultatai ir jų apibendrinimas

Šiame skyrelyje pateikti atlikto tyrimo rezultatai.

#### 3.1. Duomenys ir aprašomoji statistika

Tyrimui atlikti duomenys paimti iš „Yahoo! Finance“ platformos, kurioje pateikiamos finansinės naujienos, duomenys ir komentarai, įskaitant akcijų, kriptovaliutų kainas, pranešimus spaudai, finansines ataskaitas ir originalų turinį.

Tyrimui pasirinkta analizuoti tris kintamos vertės kriptovaliutas - bitkoiną, eterį ir laitkoiną bei vieną stabiliają kriptovaliutą – teterį. Istoriniai duomenys apie kiekvienos dienos kriptovaliutų kainas imti laikotarpiu nuo 2017.01.01. iki 2020.03.31. Iš viso turima 1186 stebėjimų. Šio laikotarpio kriptovaliutų uždarymo kainos dinamikos bei gražų grafikai pateikti 5 paveikslėlyje. Kiekvienos dienos logaritmė graža buvo rasta paėmus dviejų iš eilės uždarymo kainų santykio natūrinį logaritmą.



5 pav. Bitkoino, eterio, laitkoino ir teterio kainos kitimo (kairėje) ir gražų grafikai

Iš kainos dinamikos grafikų galima pastebėti, kad trijų kintamos vertės kriptovaliutų kainos pokyčių kryptys sutampa, o skiriasi tik kainų pokyčių apimtys. Atsiradus kriptovaliutomis pirmaisiais metais kainos kilo lėtai, kadangi į atsiradusias naujoves žiūrima atsargiai. Tačiau vis daugiau žmonių ėmė domėtis apie kriptovaliutas ir investuoti siekiant gauti kuo didesnės naudos iš kainų vertės pokyčių. Kas nulėmė, kad 2017 metais prasidėjo staigus kainų augimas. 2018 m. matoma bitkoino, eterio ir

laitkoino kainos mažėjimo tendencija. 2019 m. rinka atsigavo ir kainos pradėjo kilti, tačiau nepasiekė rekordinių aukštumų ir nuo liepos mėnesio iki pat metų pabaigos kriptovaliutų kainos palaipsniui mažėjo. Šių metų kovo mėnesį Pasaulio sveikatos organizacijos paskelbtas pranešimas apie COVID-19 pandemiją sukretė ir kriptovaliutų rinką. Fiksuotas staigus kainų kritimas. Per parą bitkoino kainos nukrito 37 %, laitkoino – 36 %, o eterio – net 42 %.

Įprastai teterio kaina svyruoja apie fiksuotą vertę ir ilgalaikių kainos kilimo ar kritimo tendencijų neturi. Iš pradžių teterio kaina buvo stabili, tačiau kai 2017 m. balandžio mėnesį teteris paskelbė, kad jų tarptautiniai pavedimai užblokuoti, kaina staigiai nukrito (8,6 %). Po šio įvykio teterio kaina pradėjo svyruoti didesne amplitude.

Iš kriptovaliutų uždarymo kainų apskaičiuotos logaritminės gražos. Stebėjimų kiekis sumažėja vienetu, kadangi buvo neįtraukta pirmosios stebėjimo dienos kainos pokyčio. Negalima paskaičiuoti gražos dydžio, neturint prie tai buvusios dienos uždarymo kainos.

Gražų grafikuose (žr. 5 pav. dešinėje) bendras tendencijas išvelgti yra sunkiau, tačiau visų valiutų kainos vienoda kryptimi reaguoja į didesnius rinkos sukrėtimus. Palyginimui apskaičiuotos pagrindinės statistinės charakteristikos, kurios pateiktos 2 lentelėje.

**2 lentelė.** Bitkoino, eterio, laitkoino ir teterio gražų pagrindinės charakteristikos

	<b>BTC</b>	<b>ETH</b>	<b>LTC</b>	<b>USDT</b>
<b>Imties dydis</b>	1185	1185	1185	1185
<b>Nulinių gražų kiekis</b>	1	0	0	40
<b>Vidurkis</b>	0,00157	0,00236	0,00183	0,00000
<b>Standartinis nuokrypis</b>	0,04487	0,05956	0,06390	0,00673
<b>CV</b>	28,52623	25,25985	34,98449	-
<b>Mažiausia reikšmė</b>	-0,46473	-0,55073	-0,44906	-0,05257
<b>Mediana</b>	0,00204	0,00017	-0,00097	0,00000
<b>Didžiausia reikšmė</b>	0,22512	0,29014	0,51142	0,05661
<b>MAD</b>	0,02783	0,03522	0,03948	0,00319
<b>Asimetrijos koeficientas</b>	-0,91352	-0,43295	0,75422	0,33133
<b>Eksceso koeficientas</b>	12,17475	9,37904	10,27061	17,53895
<b>Pusiaus dispersija</b>	0,04751	0,05780	0,05726	0,00635

Iš karto galima pastebėti, kad per analizuojamą laikotarpį eteris ir laitkoinas neturėjo nulinių gražų, o bitkoinas turėjo tik vieną. Tai patvirtina, kad kriptovaliutoms būdingas didelis kintamumas. Net ir stabiliosios kriptovaliutos teterio nulinių gražų skaičius yra nedidelis. 95% nulinių gražų yra gauti 2017 m. (žr. 1 priedas) pačioje analizuojamo laikotarpio pradžioje (žr. 5 pav.).



Nors kriptovaliutos nėra būdingos nulinės gražos, bet kintamos vertės kriptovaliutos vidurkiai yra artimi nuliui, o stabiliosios kriptovaliutos - lygus nuliui. Teterio standartinis nuokrypis taip pat ženkliai mažesnis už likusių trijų kriptovaliutų. Didžiausia duomenų skaida apie vidurkį pasižymi laikoinas.

Visos kriptovaliutos pasižymi dideliu eksceso koeficientu, kas patvirtina, kad duomenų skaida yra mažesnė lyginant su normaliuoju skirstiniu. Be to asimetrijos koeficientas nėra artimas nuliui. Analizuojamu laikotarpiu bitkoinui ir eteriui būdinga kairioji asimetrija, o laikoinui ir teteriui dešinioji asimetrija. Šie rodikliai parodo, kad duomenys nėra pasiskirstę pagal Gauso skirstinį ir tokius atveju nebūtų tikslinga taikyti tradicinius finansinių laiko eilučių modelius.

Nesimetriniams duomenims vietoje standartinio nuokrypio rekomenduojama naudoti MAD (*angl.* Mean Absolute Deviation) rodiklį, kuris parodo vidutinį teigiamą nuokrypį tarp stebimo dydžio ir duomenų vidurkio. Nedidelės šio rodiklio reikšmės rodo, kad kriptovaliutos yra koncentruotos apie vidurkį.

Gražų esančių mažesnių už vidurkį rizikai įvertinti apskaičiuotas pusiau dispersijos matas. Didžiausią neigiamą pasiskirstymo riziką turi eteris ir laikoinas, šiek tiek mažesnę riziką turi bitkoinas, o mažiausią riziką – teteris.

Taigi, apskaičiavus statistines charakteristikas galima nesunkiai atskirti, kurios kriptovaliutos yra kintamos vertės, jų variacijos koeficientas priklausomai nuo valiutos yra tarp 25 – 35 %, Teteriui šis koeficientas yra neapibrėžtas, nes vidurkis yra lygus 0 ir ši valiuta yra laikoma stabilios vertės kriptovaliuta.

Apskaičiavus tos pačios rodiklius kiekvieniems metams atskirai (žr. 1 priedas), galima pastebėti, kad, pavyzdžiui, didelės eksceso koeficiento reikšmės nulėmė šių metų pirmojo ketvirčio kainų pokyčiai. Taip pat šiais metais yra fiksuoti ir mažiausios gražų reikšmės, tuo tarpu didžiausi teigiami pokyčiai fiksuoti 2017 m.

### **3.2. HMM modelio pritaikymas paslėptųjų būsenų (režimų) nustatymui kriptovaliutų rinkoje**

HMM metodas skirtas modeliuoti paslėptųjų būsenų perėjimus iš nuosekliai stebimų duomenų. Kadangi kainos priklauso nuo daugybės veiksnių, kurie investuotojui paprastai lieka nematomi (paslėpti kintamieji), o perėjimas tarp pagrindinių veiksnių vyksta atsižvelgiant į rinkos situaciją,

spekulantų sprendimus ir kitus pasaulinius įvykius, kurie daro įtaką kainai (stebimi duomenys), tai laikoma, kad šis modelis yra tinkamas finansinių duomenų kainų prognozavimo problemai spręsti.

Šio skyrelio tikslas - naudojant kriptovaliutų kainų duomenis nustatyti, ar egzistuoja paslėptieji rinkos režimai bei rasti optimalų būsenų skaičių.

Labai svarbu parinkti tinkamą paslėptųjų būsenų skaičių modeliuojant HMM. Modelio tinkamumui vertinti naudoti du standartiniai informaciniai kriterijai: AIC (*angl.* Akaike information criterion) ir BIC (*angl.* Bayesian information criterion), kad iširti HMM efektyvumą skirtingam būsenų kiekiui. Pasirinkti šie informaciniai kriterijai, kadangi modelio apmokymo algoritme (*Baum-Welch* algoritme) naudojamas EM metodas, kuris gražina maksimalią tikėtimumo funkcijos reikšmę. Ši reikšmė reikalinga, norint apskaičiuoti AIC ir BIC kriterijus:

$$AIC = -2 \ln(L) + 2k, \quad (3.1)$$

$$BIC = -2 \ln(L) + k \ln(n), \quad (3.2)$$

kur  $L$  yra maksimizuota modelio tikėtimumo funkcijos reikšmė,  $n$  – stebėjimų skaičius ir  $k$  – modelio parametrų skaičius (Nguyen 2017).

Remiantis ankstesniais tyrimais pateiktais literatūros apžvalgoje būsenų skaičiaus diapazonas pasirinktas nuo dviejų iki šešių. Modelius su didesniais būsenų skaičiais būtų sunku interpretuoti. HMM modelio apmokymui naudoti fiksuoto skaičiaus stebėjimų duomenys:

$$O = \{O_t^{(1)}, O_t^{(2)}, t = 1, 2, \dots, T\}, \quad (3.3)$$

kur  $O_t^{(1)}$  yra kriptovaliutos gražos dydis laiko momentu  $t$ , o  $O_t^{(2)}$  – ATR indikatorius laiko momentu  $t$ . Šis vidutinio tikrojo intervalo indikatorius (*angl.* Average True Range) išmatuoja rinkos kintamumą. ATR apskaičiuojamas slenkančio vidurkio metodu imant tikrojo intervalo reikšmes, kurios gaunamos atsižvelgiant į aukščiausios (*angl.* High), žemiausios (*angl.* Low) ir uždarymo (*angl.* Close) kainos duomenis (Investopedia 2020):

$$TR = \max\{High - Low; |High - Close|; |Low - Close|\}, \quad (3.4)$$

$$ATR = \frac{1}{n} \sum_{i=1}^n TR_i, \quad (3.5)$$

Darbe naudotas dažniausiai taikomas 14 dienų slenkantis vidurkis ATR indikatoriumi apskaičiuoti. ATR naudojamas kaip kriptovaliutų kintamumo rodiklis.

### 3.2.1. HMM modelio optimalaus būsenų skaičiaus radimas

Modelis tikrintas pasirenkant kelis skirtingus stebėjimų laiko ilgius  $T$ . Keičiant būsenų skaičių nuo dviejų iki šešių HMM modeliai pritaikyti, kai  $T = \{365, 730, 1095, 1185\}$ . Rezultatų lentelė su AIC, BIC ir tikėtinumo funkcijos reikšmėmis pateikta 2 priede. Tinkamiausias modelis renkamas atsižvelgiant į mažiausias AIC ir BIC reikšmes (žr. 3 lentelė).

3 lentelė. HMM modeliai pagal mažiausias AIC ir BIC informacinius kriterijus

Kripto valiuta	T	N	AIC	BIC	Tikėtinumo funkcijos reikšmė
<b>Modeliai turintys mažiausią AIC reikšmę</b>					
<b>BTC</b>	1172	4	-4416,32	-4299,79	2231,16
<b>ETH</b>	1172	6	-3747,19	-3509,06	1920,59
<b>LTC</b>	1172	5	-3626,32	-3454,06	1847,16
<b>USDT</b>	1172	5	-9873,36	-9701,10	4970,68
<b>Modeliai turintys mažiausią BIC reikšmę</b>					
<b>BTC</b>	1172	3	-4407,37	-4336,44	2217,69
<b>ETH</b>	1172	2	-3695,03	-3659,57	1854,52
<b>LTC</b>	1172	3	-3605,02	-3534,09	1816,51
<b>USDT</b>	1172	5	-9873,36	-9701,10	4970,68

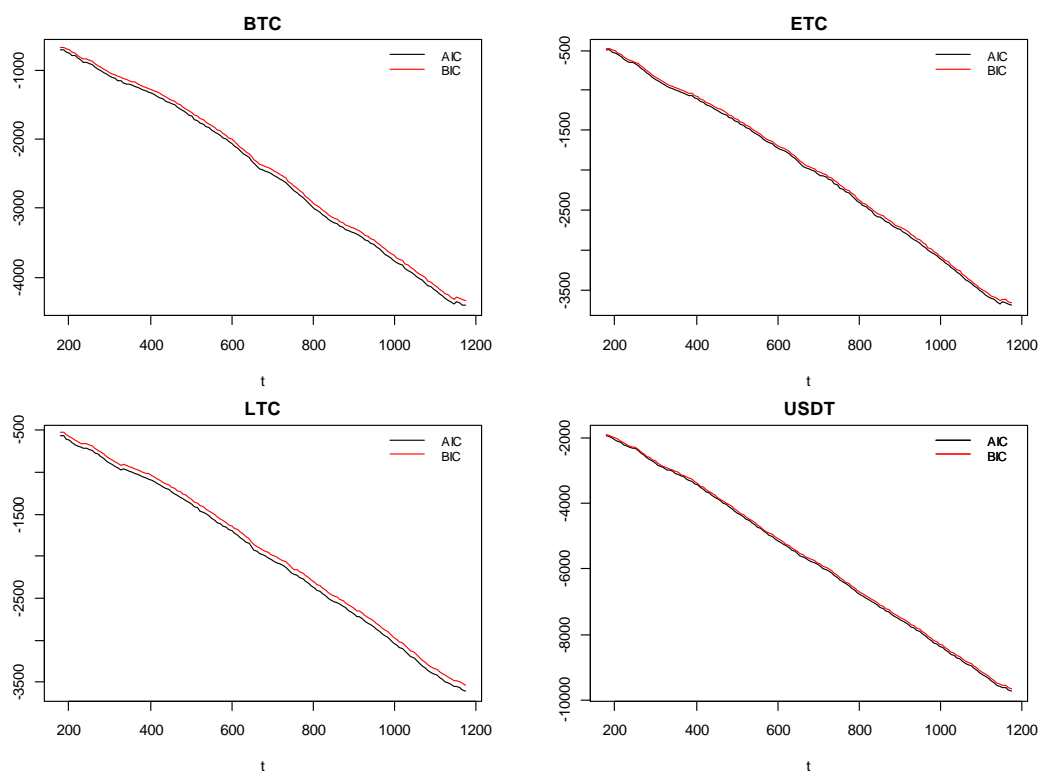
Iš lentelės galima matyti, kad išsiskiria modelio tikėtinumo funkcijos reikšmės kintamos vertės ir stabilios vertės valiutų atvejais. Teterio kriptovaliutų gražoms modelio tikėtinumo funkcijos reikšmės daugiau nei dvigubai didesnės už modelius pritaikytus kintamos vertės kriptovaliutoms. Lyginant šitas kriptovaliutas tarpusavyje, tai didžiausią standartinį nuokrypį turinčio laikotinio gražų HMM modeliai yra įvertinti prasčiausiai. Iš čia daroma išvada, kad kuo kriptovaliutų kainos yra nepastovesnės, tuo HMM modelio efektyvumas mažesnis.

Būsenų skaičiaus parinkimui naudojant tik AIC kriterijų, gauta, kad laikotiniui ir teteriui geriausias modelis sudarytas su 5, bitkoinui – su 4, o eteriui – su 6 būsenomis. Pagal BIC kriterijaus mažiausias reikšmes būsenų skaičius sutampa tik teteriui. Kitoms kriptovaliutoms įvertinant bauduojant kiekvieną papildomai įtrauktą būseną, gauta, kad bitkoinui ir laikotiniui optimaliausia naudoti 3 būsenas, o eteriui tik 2. Įvertinus tikėtinumo funkcijos reikšmes atsižvelgiant į abu kriterijus bendrai (žr. 3 priedą) kintamos vertės valiutoms tinkamiausias yra 3 būsenų modelis, kadangi BIC kriterijaus reikšmė pradeda blogėti didinant būsenų skaičių. Teterio atveju 3 ir 4 būsenų HMM modelių

efektyvumas yra panašus ir ryškesnis skirtumas matomas, kai imama nedidelė stebėjimų seka ( $T = 365$ ). Laikome, kad 3 būsenų HMM modelis yra tinkamas ir pakankamai efektyvus visoms kriptovaliutoms.

Geriausi modeliai, nepriklausomai nuo to pagal kurį kriterijų vertinama, yra gauti paėmus didžiausią galimą stebėjimo sekos ilgį ( $T = 1172$ ). Visoms kriptovaliutoms į modelį įtraukus daugiau stebėjimų gaunamos mažesnės AIC ir BIC reikšmės. Tačiau lyginant vienodų būsenų HMM modelius tarpusavyje, kai imamas 3,2 metų ( $T = 1172$ ) ir 3 metų ( $T = 1095$ ) stebėjimus, modelio AIC ir BIC reikšmės pasikeičia palyginus nedaug 5-7%.

Patikrinta, kaip AIC ir BIC reikšmės priklauso nuo stebėjimo sekos ilgio trijų būsenų HMM modelyje. Stebėjimai imti pradėdant nuo 180 ilgio sekos ir su kiekviena iteracija pridėta po 7 stebėjimus. Iteracijos vykdytos tol kol išnaudoti visi turimi stebėjimų duomenys.



**6 pav.** AIC ir BIC reikšmių priklausomybė nuo stebėjimų skaičiaus

Iš 6 paveikslo galima matyti, kad didinant stebėjimų skaičių modelyje abiejų informacinių kriterijų reikšmės mažėja. Yra labai minimalių lokalių pablogėjimų, tačiau bendra tendencija išlieka tokia pati, todėl, naudota pilna stebėjimų seka, kad nustatyti, ar egzistuoja rinkos režimai darantis įtaką kriptovaliutų kainoms.

### 3.2.2. Kriptovaliutų rinkos režimų nustatymas naudojant 3 būsenų HMM

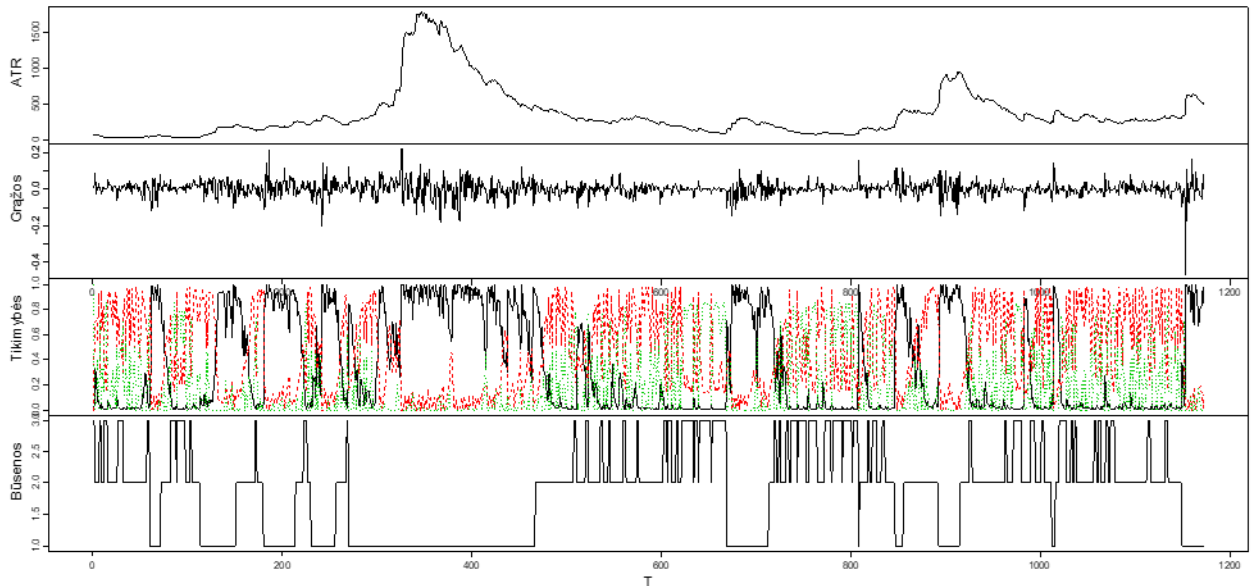
Finansinius duomenis veikia įvairūs rinkos režimai tokie kaip besikeičianti reguliavimo aplinka arba ekstremalaus nepastovumo laikotarpiai. Naudotas 3 būsenų HMM metodas, kad nustatyti, ar kriptovaliutų rinkai yra būdingas režimų pasikeitimas (*angl. regime switching*).

Kaip ir prieš tai, sudarytas 3 būsenų HMM modelis yra pritaikytas naudojant *Baum-Welch* algoritimą. Apmokinus duomenis sprendžiama iššifravimo problema naudojant *Viterbi* algoritimą, kad sugeneruotų labiausiai tikėtiną paslėptųjų būsenų seką.

Rezultatai gauti pritaikius modelius pateikti lentelėse (žr. 4 lentelė). O paveikslėliuose pavaizduoti ATR indikatorius, gražų kitimo, kiekvieno režimo posteriorinės tikimybės bei būsenų sekos grafikai. (žr. 5-8 pav.). Tikimybių grafike juoda spalva žymi tikimybes, kad būseną yra 1 režime, raudona – 2 režime, o žalia – 3 režime.

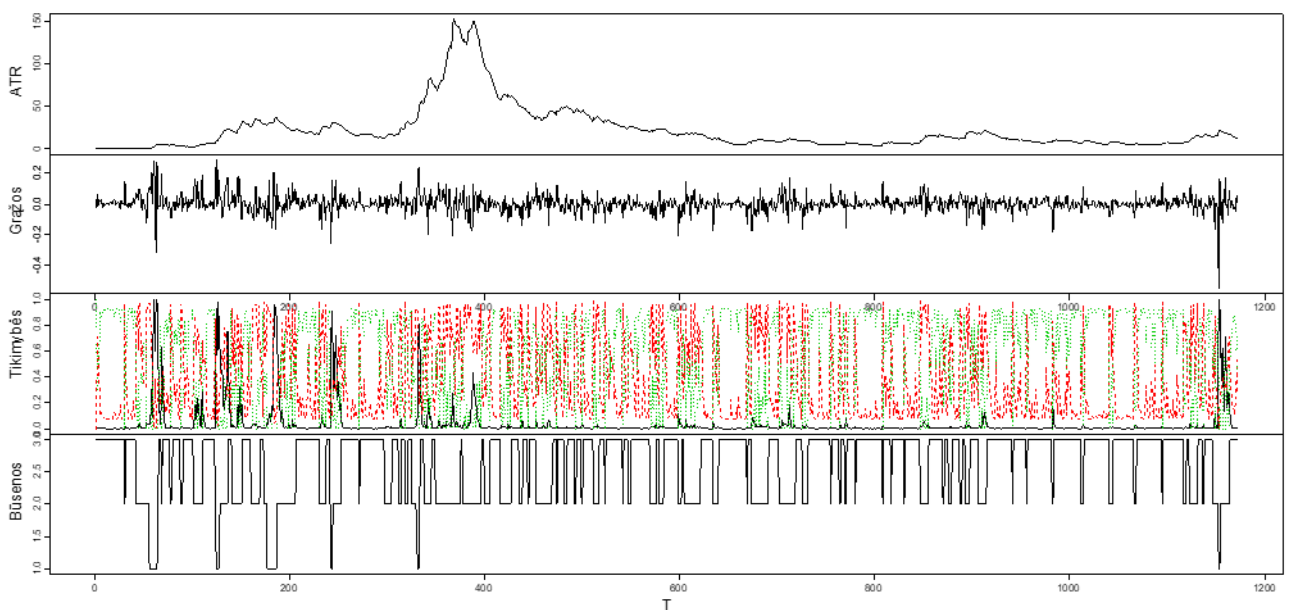
4 lentelė. 3 būsenų HMM parametrų reikšmės

	Bitkoinas			Eteris			Laitkoinas			Teteris		
<b>Pradinės tikimybės</b>												
	$\pi = (0,0,1)$			$\pi = (0,0,1)$			$\pi = (1,0,0)$			$\pi = (1,0,0)$		
<b>Perėjimo matricos</b>												
	toS1	toS2	toS3	toS1	toS2	toS3	toS1	toS2	toS3	toS1	toS2	toS3
<b>fromS1</b>	0,939	0,061	0,000	0,771	0,229	0,000	0,885	0,030	0,085	0,962	0,000	0,038
<b>fromS2</b>	0,027	0,792	0,181	0,025	0,712	0,263	0,000	0,463	0,537	0,000	0,850	0,150
<b>fromS3</b>	0,029	0,395	0,576	0,000	0,194	0,806	0,080	0,023	0,897	0,003	0,056	0,941
<b>Būsenų vidurkiai (<math>\mu</math>) ir standartiniai nuokrypiai (<math>\sigma</math>)</b>												
	$\mu$	$\sigma$		$\mu$	$\sigma$		$\mu$	$\sigma$		$\mu$	$\sigma$	
<b>St1</b>	0,000	0,069		0,022	0,160		-0,001	0,023		0	0,000	
<b>St2</b>	0,003	0,032		0,002	0,071		0,066	0,185		0	0,013	
<b>St3</b>	0,001	0,008		0,001	0,024		-0,001	0,061		0	0,003	



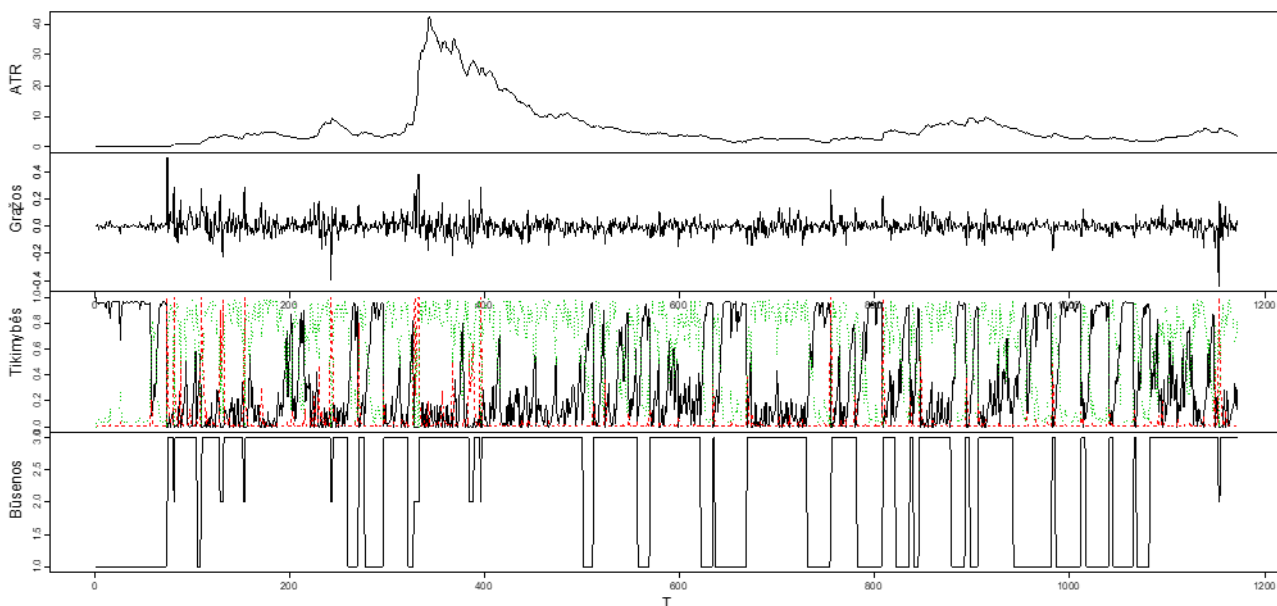
**7 pav.** Bitkoino 3 būsenų HMM modelio ATR, gražų, posteriorinių tikimybių ir būsenų grafikai

Bitkoino 3 būsenų HMM modelyje pradinė būsenų tikimybė siūlo, kad pradinė būseną yra 3 režime, prie kurio galima priskirti prie labai žemo kintamumo ( $\sigma = 0.008$ ) turinčias gražas. Iš grafiko (žr. 7 pav.) aiškiai matyti, kad į 1 režimą papuola didelio kintamumo gražos ( $\sigma = 0.069$ ). Įprasto kintamumo ( $\sigma = 0.032$ ) gražas apibudina 2 režimas. Posteriorinės tikimybės ir būsenų diagramose matome, kaip keičiasi režimai tam tikrais laiko momentais. Šie perėjimai susiję su perėjimo matricos vertėmis. Yra 94 % tikimybė likti 1 režime, 75 % – 2 režime ir 58 % – 3 režime. Režimo pasikeitimo tikimybės yra mažos arba net neįmanomos. Tikimybė, kad būseną pereis iš 1 režimo į 3 lygi 0.



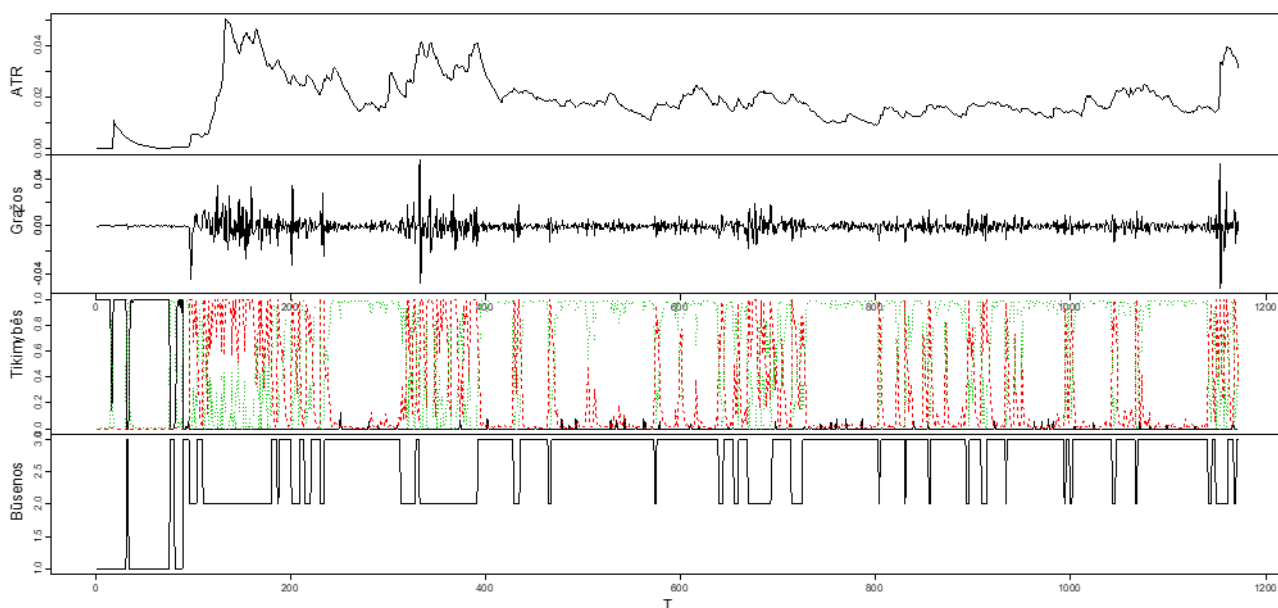
**8 pav.** Eterio 3 būsenų HMM modelio ATR, gražų, posteriorinių tikimybių ir būsenų grafikai

Kaip ir bitkoino, taip ir eterio 3 būsenų HMM modelyje pradinė būsenų tikimybė siūlo, kad pradinė būseną yra 3 režime. Nors šio režimo negalima pavadinti žemo kintamumo ( $\sigma = 0.024$ ), bet atsižvelgus, kad kitų režimų standartiniai nuokrypiai yra daug didesni, tai galima sakyti, kad tai žemiausio kintamumo režimas. Analogiškai 2 režimas apima vidutinio kintamumo grąžas, o tada 1 režimas – didžiausio kintamumo. 1 režimas išsiskiria ir tuo, kad grąžų vidurkis ( $\mu = 0.022$ ) yra ženkliai didesnis už kitų dviejų režimų, kurių vidurkiai artimi nuliui. Teigiama vidurkio reikšmė rodo, kad esant šiam režimui eterio kainos linkusios stipriai didėti, o iš grafiko (žr. 8 pav.) galima matyti, kad šis režimas pasitaiko itin retai. Perėjimo matricos vertės įstrižainėje yra didesnės nei 70%, todėl labiau tikėtina, kad būseną išliks tame pačiame režime nei pereis į kitus režimus. Režimo pasikeitimo tikimybės pereiti iš 1 režimo į 3 režimą ir atvirkščiai yra lygi 0.



**9 pav.** Laitkoino 3 būsenų HMM modelio ATR, grąžų, posteriorinių tikimybių ir būsenų grafikai

Laitkoino 3 būsenų modelio rezultatai iš pradžių gali pasirodyti visai kitoks negu prieš tai nagrinėti. Pagal pradinę tikimybę, būseną yra 1 režime. Įvertinus visų režimų standartinius nuokrypius, šiam režimui priklauso mažiausiai kintančios grąžos ( $\sigma = 0.023$ ). Vidutinio kintamumo ( $\sigma = 0.061$ ) grąžos laitkoino modelyje yra 3 režime, o labiausiai nepastovumas fiksuojamas 2 režime ( $\sigma = 0.185$ ), kuriame vidurkis taip pat nėra artimas nuliui ( $\mu = 0.066$ ). Kaip ir eterio atveju, šiame režime būseną atsiduria palyginus retai (žr. 9 pav.). O taip yra todėl, kad yra tik 46% tikimybė, kad išliks 2 režimas. Be to iš šio režimo negalima pereiti į žemiausio kintamumo režimą.



**10 pav.** Teterio 3 būsenų HMM modelio ATR, gražų, posteriorinių tikimybių ir būsenų grafikai

Kadangi teteris svyruoja apie fiksuotą vertę, tai HMM modelyje visų režimų vidurkių vertės gautos nulinės. Skiriasi tik standartiniai nuokrypiai priklausomai nuo režimo. 1 režime ne tik vidurkis, bet ir standartinis nuokrypis yra lygus 0. Šis režimas neturi jokio poveikio teterio kainai ir ji išlieka stabili bei nekinta. Žiūrint į grafiką (žr. 10 pav.) šiame režime būseną buvo tik pačioje stebėjimų pradžioje. Yra du gali atvejai, kai būseną yra 1 režime, tai kad išliks tas pats režimas arba bus pereita į 3 režimą, kuris turi nedidelio kintamumo svyravimus. Perėjus į šį režimą mažai tikėtina, kad būseną sugrįš atgal, o duodama net 94% tikimybė, kad režimas išliks tas pats. Didesni teterio kainų pokyčiai įvyko kai būseną buvo 2 režime. Šiame režime pasirodžiusių gražų standartinis nuokrypis lygus 0.013.

Taigi, paslėptasis Markovo modelis leido identifikuoti paslėptuosius režimus, kurie turi skirtingą poveikį kriptovaliutų kainų kintamumui. Pagal 4 lentelėje esančius standartinius nuokrypius matyti, kad režimai suskirstyti į žemo kintamumo (pažymėta žalia spalva), vidutinio (pažymėta geltona spalva) ir aukšto kintamumo (pažymėta raudona spalva) režimus atitinkamai kiekvienai kriptovaliutai.

### 3.3. Kainos krypties prognozė dviejų būsenų paslėptuoju Markovo modeliu

Kainos krypties prognozė atlikta kintamos vertės kriptovaliutoms - bitkoinui, eteriui ir laitkoinui.

Apmokymo duomenims pasirinkti kriptovaliutų gražų duomenys nuo 2017 m. sausio 1 d. iki 2019 m. gruodžio 31d. Testavimui naudota šių metų I ketvirčio kriptovaliutų gražos.



Sudarytas HMM modelis, kurio tikslas nustatyti, kaip gerai HMM modelis gali prognozuoti paslėptąsias būsenas, kai stebėjimai priklauso kategoriniams kintamiesiems.

Visų pirma atliktas duomenų diskretizavimas. Iš kriptovaliutų gražų sekos sudarytos dvi sekos: stebėjimų seka ir paslėptųjų būsenų seka skirta patikrinti modelio prognozės tikslumui. Pastaroji seka vaizduoja realias kriptovaliutų kainos kryptis analizuojamu laikotarpiu ir ji bus naudojama įvertinant modelio atlikto prognozavimo tikslumą. Kainų gražos, kurios yra mažesnės arba lygios 0 pažymimos, kad yra būsenoje „Mažėja“, o kurios yra didesnės už 0 – būsenoje „Didėja“. Taigi, galimų būsenų aibė:

$$S = \{Mazeja, Dideja\}. \quad (3.6)$$

Stebėjimų sekai sudaryti naudojami 5 simboliai „EK“, „K“, „Z“, „A“, „EA“ atitinkamai žymintys kainos ekstremalų kritimą, kritimą, žemą kintamumą, augimą ir ekstremalų augimą. Simboliai priskirti kainų gražoms tokiais intervalais:

$$F(x) = \begin{cases} EK & , kai x \in (-1; -0.05] \\ K & , kai x \in (-0.05; -0.01] \\ Z & , kai x \in (-0.01; 0.01] \\ A & , kai x \in (0.01; 0.05] \\ EA & , kai x \in (0.05; 1] \end{cases} \quad (3.7)$$

Daroma prielaida, kad pradinės būsenos tikimybė  $\pi$  yra vienodai pasiskirsčiusi. Pagal (2.1) ir (2.2) formules kiekvienai kriptovaliutai apskaičiuojama perėjimo matrica  $A$  ir emisijos matrica  $B$ . Apibrėžiamas pradinis HMM modelis:

$$\lambda = \{A, B, \pi\}. \quad (3.8)$$

Pagrindinis tyrimas atliktas imant kiekvienos dienos kriptovaliutų gražas. Vėliau taip pat patikrinta, kaip modelio tikslumas kinta agreguotiems duomenims. Duomenys agreguoti apskaičiuojant logaritmines gražas kriptovaliutų 3 dienų, 5 dienų ir savaitiniams kainų vidurkiams.

### 3.3.1. Pirmos eilės Markovo grandinių testas

Patikrinimas, ar Markovo grandinės modelyje yra pirmos eilės. Atliekamas testas, kuris tikrina simbolių sekos tinkamumą pirmos eilės Markovo grandinėms. Testas tikrinama hipotezė:

$$H_0: \text{seka yra pirmos eilės Markovo grandinė}$$

$H_a$ : seka nėra pirmos eilės Markovo grandinė

Tikrinamas duotosios sekos nepriklausomumas naudojant *Ljung-Box* testą, o pasiskirstymo tolydumas intervale remiasi Kolmogorovo-Smirnovo testu.

Testas yra supaprastintas taip, kad atliekama kelių bandymų pataisa, kuri suteikia vieną pakoreguotą  $p$  vertę. Jei ši  $p$  vertė yra mažesnė už reikšmingumo lygį, tai nulinė hipotezė, kad yra pirmos eilės Markovo grandinė atmetama. Priešingu atveju nulinė hipotezė priimama.

**5 lentelė.** Pirmos eilės Markovo grandinių testo  $p$ -reikšmių rezultatai

	Box-Ljung	Kolmogorov-Smirnov	Markov test
<b>Kasdieniniai duomenys</b>			
<b>BTC</b>	0,928410	3,0920e-09	3,092e-09
<b>ETH</b>	0,081843	0,081843	0,081843
<b>LTC</b>	0,574621	0,022853	0,022853
<b>3 dienų duomenys</b>			
<b>BTC</b>	0,017076	1,8681e-06	1,8681e-06
<b>ETH</b>	0,000663	0,005194	0,000663
<b>LTC</b>	0,000509	0,026611	0,000509
<b>5 dienų duomenys</b>			
<b>BTC</b>	0,841915	0,000307	0,000307
<b>ETH</b>	0,507930	0,179240	0,179240
<b>LTC</b>	0,021763	0,050112	0,021763
<b>Savaitiniai duomenys</b>			
<b>BTC</b>	0,481087	0,025699	0,025699
<b>ETH</b>	0,170372	0,009344	0,009344
<b>LTC</b>	0,017224	0,006850	0,006850

Sudarytoms paslėptųjų būsenų sekoms pritaikytas Markovo testas ir jo rezultatai pateikti 5 lentelėje.

Tik eterio kasdieninių kainų kryptių seka atitinka pirmos eilės Markovo grandinių savybes. Kitos dvi kriptovaliutos netenkina Kolmogorovo-Smirnovo testo, nes apskaičiuota  $p$ -reikšmė yra mažesnė už reikšmingumo lygį  $\alpha = 0,05$ . Dėl tos priežasties yra atmetama hipotezė, kad sekos yra pirmos eilės Markovo grandinės. Tačiau pagal *Box-Ljung* testą gauta, kad tiek bitkoino, tiek laitkoino kainų kryptių sekos yra nepriklausomos.

Tikrinant agreguotų duomenų Markovo pirmos eilės testą gauta, kad eterio 5 dienų duomenys tenkina abu testus reikalingus priimti nulinei hipotezei, jog Markovo grandinė yra pirmos eilės. Galima taip pat pastebėti, kad *Box-Ljung* testo  $p$ -reikšmės yra didesnės už  $\alpha$  reikšmingumo lygį, kai turimi 5 dienų ir savaitiniai duomenys apie bitkoino kainos kryptis.

Įvertinant rezultatus bendrai HMM modeliavimui tinkamiausia yra imti kasdieninius duomenis. Prieš tai dar atlikti papildomi testai, kurie tikrina, ar Markovo grandinės yra ergodiškos, ar tenkinama Markovo savybė, taip pat atliekamas pakartotinis Markovo eilės testas bei įvertinamas stacionarumas. Rezultatai pateikti 6 lentelėje.

6 lentelė. Markovo savybių testų rezultatai

	<b>Ergodiškumo testas</b>	<b>Markovo savybės testas</b>	<b>Markovo eilės testas</b>	<b>Stacionarumo testas</b>
	True/False	p-value	p-value	p-value
<b>Kasdieniniai duomenys</b>				
<b>BTC</b>	True	0,93428	0,26640	1
<b>ETH</b>	True	0,99948	0,77381	1
<b>LTC</b>	True	0,99962	0,75395	1
<b>3 dienų duomenys</b>				
<b>BTC</b>	True	0,999999	0,98217	1
<b>ETH</b>	True	0,999977	0,92236	1
<b>LTC</b>	True	0,999977	0,92236	1
<b>5 dienų duomenys</b>				
<b>BTC</b>	True	0,88621	0,24761	0,999985
<b>ETH</b>	True	0,99933	0,81574	0,999997
<b>LTC</b>	True	0,99933	0,81574	0,999997
<b>Savaitiniai duomenys</b>				
<b>BTC</b>	True	0,81603	0,14023	0,999574
<b>ETH</b>	True	0,99998	0,96864	0,999498
<b>LTC</b>	True	0,99998	0,96864	0,999499

Ergodiškumo testas yra teigiamas visoms būsenų sekoms. Taip pat yra tenkinamas Markovo savybės testas. Markovo eilės testo  $p$ -reikšmė yra didesnė už reikšmingumo lygį  $\alpha = 0,05$ , tai reiškia, kad negalime atmesti hipotezės, kad Markovo grandinės seka yra pirmos eilės. Taip pat negalime atmesti hipotezės, kad seka yra stacionari. Šie testai parodo, kad sudarytas sekas galima laikyti Markovo grandinėmis ir HMM modelis gali būti taikomas.

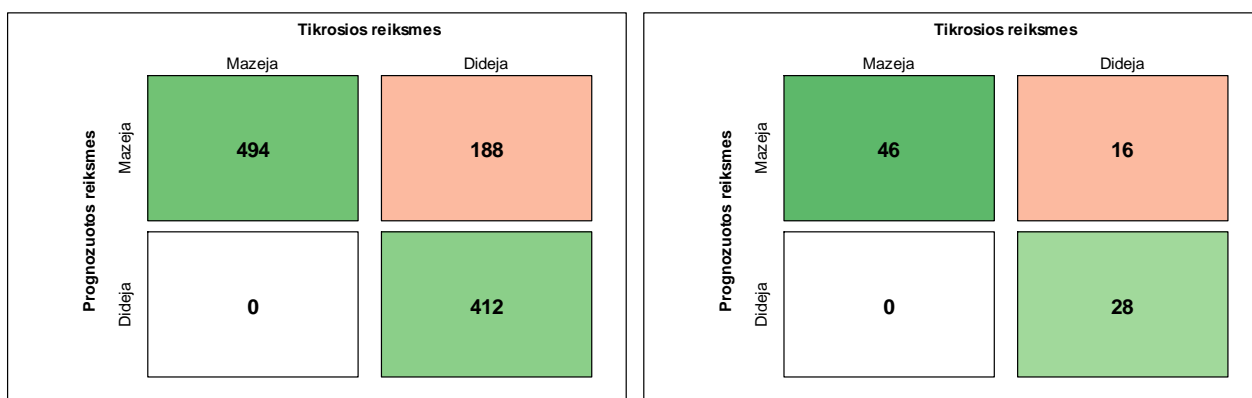
### 3.3.2. Sudaryto modelio pritaikymo eiga

Pradinis HMM modeliui pritaikomas *Baum-Welch* algoritmas, kuris skirtas modelio apmokymui ir pagal stebėjimų seką perskaičiuoja perėjimo ir emisijos tikimybių matricas. Maksimalus algoritmo iteracijų skaičius yra 100 iteracijų. Taip pat tikrinama papildoma nutraukimo sąlyga, jei perėjimo ir emisijos matricos konverguoja anksčiau negu įvykdomos visos iteracijos. Perėjimo ir emisijos parametrų skirtumas turi būti mažesnis nei  $1 \times 10^{-9}$ , kad būtų galima nutraukti algoritmą.

Apskaičiavus naujas perėjimo ir emisijos tikimybes *Viterbi* algoritmo pagalba atliekama būsenų prognozė. Gautos suprognuozuotos kainų kryptys palyginamos su tikrosiomis reikšmėmis apskaičiuojant sumaišymo matricą. Prognozės tikslumo rezultatai pateikti sekančiame skyrelyje.

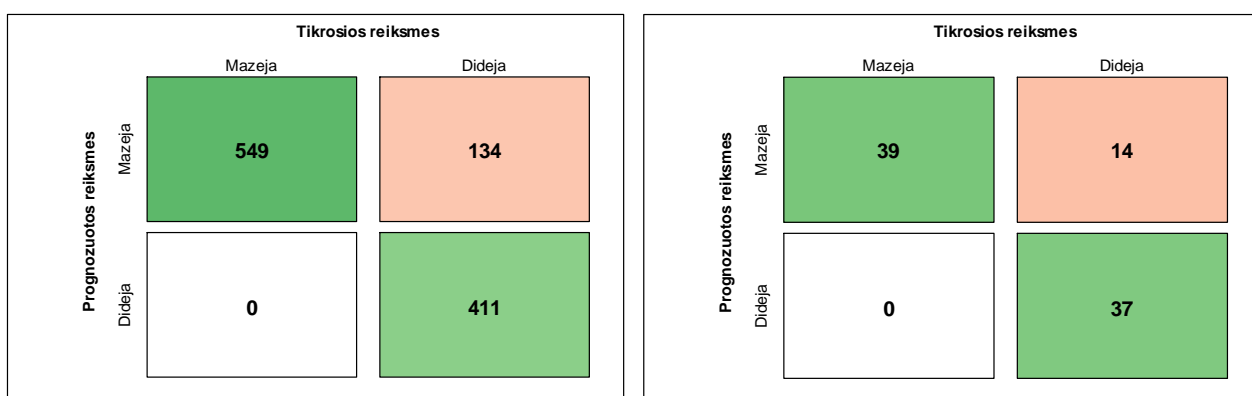
### 3.3.3. Kriptovaliutų kainos krypties prognozės HMM modeliu įvertinimas

Ankstesniame skyrelyje aprašytas HMM modelis pritaikytas bitkoino, eterio ir laitkoino apmokymo duomenims. Gautas modelis su naujais parametru įverčiais išbandytas imant testavimo duomenis. Sudaryto apmokyto modelio patikimumui įvertinti, panaudota testavimui skirti stebėjimai. Pagal stebėjimų seką atlikus paslėptųjų būsenų kainų krypties prognozė ir palyginus su realiomis reikšmėmis sudarant sumaišymo matricos (žr. 11 – 13 pav.).



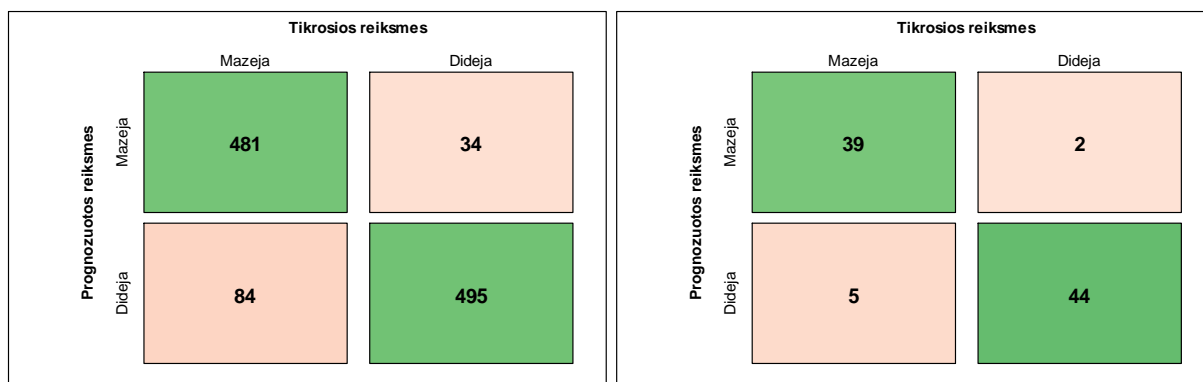
11 pav. Bitkoino apmokymo duomenų (kairėje) ir testavimo duomenų sumaišymo matrica

Bitkoino kainos krypties prognozės modelis apmokymo imtyje 906 kartus nustatė kryptį teisingai, o 188 kartus buvo suklysta prognozuojant, kad kainos mažės, tačiau kainų kryptis buvo didėjanti. Modelio prognozė testavimo imčiai analogiška. Neatitikimas pasitaikė manant, kad kainos turėtų kristi, tačiau jų kryptis iš tikrųjų buvo teigiama. Tokią klaidą modelis padarė 16 kartų.



**12 pav.** Eterio apmokymo duomenų (kairėje) ir testavimo duomenų sumaišymo matrica

HMM modelis tokia pačia struktūra suveikė ir eterio kriptovaliutai. Palyginus sumaišymo matricas pastebimas mažesnis klaidų skaičius. Apmokymo imčiai buvo suklysta 134 kartus, o testavimo imčiai – 14. Kad kainos mažėja apmokymo modelyje teisingai nustatyta 549 kartus, o kad kainos didės 411.



**13 pav.** Laitkoino apmokymo duomenų (kairėje) ir testavimo duomenų sumaišymo matrica

Prognozė laitkoino kainoms skiriasi nuo prieš tai buvusių valiutų prognozės rezultatu. Didesnė dalis prognozuotų kryptių sutapo su realiomis. Iš viso neatitiko 118 prognozuotų kryptių apmokymo duomenims, tačiau pasitaikė abiejų tipo klaidų. Dažniau klydo prognozuojant, kad kainos didės, nors iš tikrųjų kainos mažėja. Testavimo imties klaidų skaičius labai nedidelis lyginant su bitkoino ar eterio kainų krypties prognozę.

Remiantis sumaišymo matricomis apskaičiuoti klasifikatoriaus kokybės matai prognozės tikslumui įvertinti. Modelio apmokymo ir testavimo imtis gautų rodiklių reikšmės pateiktos 7 lentelėje.

**7 lentelė.** HMM prognozės metodo tikslumo rodiklių reikšmės

Rodikliai	Apmokymas				Testavimas			
	BTC	ETH	LTC	Vidutinis	BTC	ETH	LTC	Vidutinis
<b>Taiklumas</b>	0,828	0,878	0,892	<b>0,866</b>	0,822	0,844	0,922	<b>0,863</b>
<b>Kappa</b>	0,664	0,755	0,785	<b>0,735</b>	0,641	0,696	0,844	<b>0,727</b>
<b>Jautrumas</b>	1,000	1,000	0,851	<b>0,950</b>	1,000	1,000	0,886	<b>0,962</b>
<b>Specifiškumas</b>	0,687	0,754	0,936	<b>0,792</b>	0,636	0,725	0,957	<b>0,773</b>
<b>Tikslumas</b>	0,724	0,804	0,934	<b>0,821</b>	0,742	0,736	0,951	<b>0,810</b>
<b>F1 įvertis</b>	0,840	0,891	0,891	<b>0,874</b>	0,852	0,848	0,918	<b>0,873</b>

HMM modeliu atliktos kainos prognozės taiklumo rodiklis apmokymo duomenų rinkiniams yra didesnis nei 82 %. Apmokymo imčiai bitkoino kainos kryptį taikliai siekia 82,8 %, eterio 87,8%, o laitkoino 89,2 %.

*Kappa* rodiklis mažiausias gautas bitkoinui ir lygus 0,64, eteriui ir laitkoinui gautos didesnės šio rodiklio reikšmės, atitinkamai lygios 0,755 ir 0,785.

Jautrumo reikšmės tiek bitkoinui, tiek eteriui yra lygios vienetui. Skiriasi tik laitkoinui, kuriam gauta, kad modelio jautrumas lygus 0,851. Vidutiniškai 95 % visų kainos mažėjimų buvo prognozuota teisingai.

Sekančio rodiklio rezultatai skirtingi lyginant visas tris kriptovaliutas. Bitkoino atveju specifiškumo vertė gauta mažiausia ir lygi 0,687, o laitkoino atveju – didžiausia (0,936). Vidutiniškai 79,2 % visų kainos didėjimų buvo prognozuota teisingai.

Tikslumas pagal kriptovaliutas rikiuojasi taip: blogiausiai įvertintas bitkoino modelis, kurio tikslumas yra 0,724, eterio tikslumas siekia 0,804, o laitkoino – 0,934. Vidutiniškai 82,1 % visų prognozuotų kainos mažėjimų buvo teisingai nustatyti.

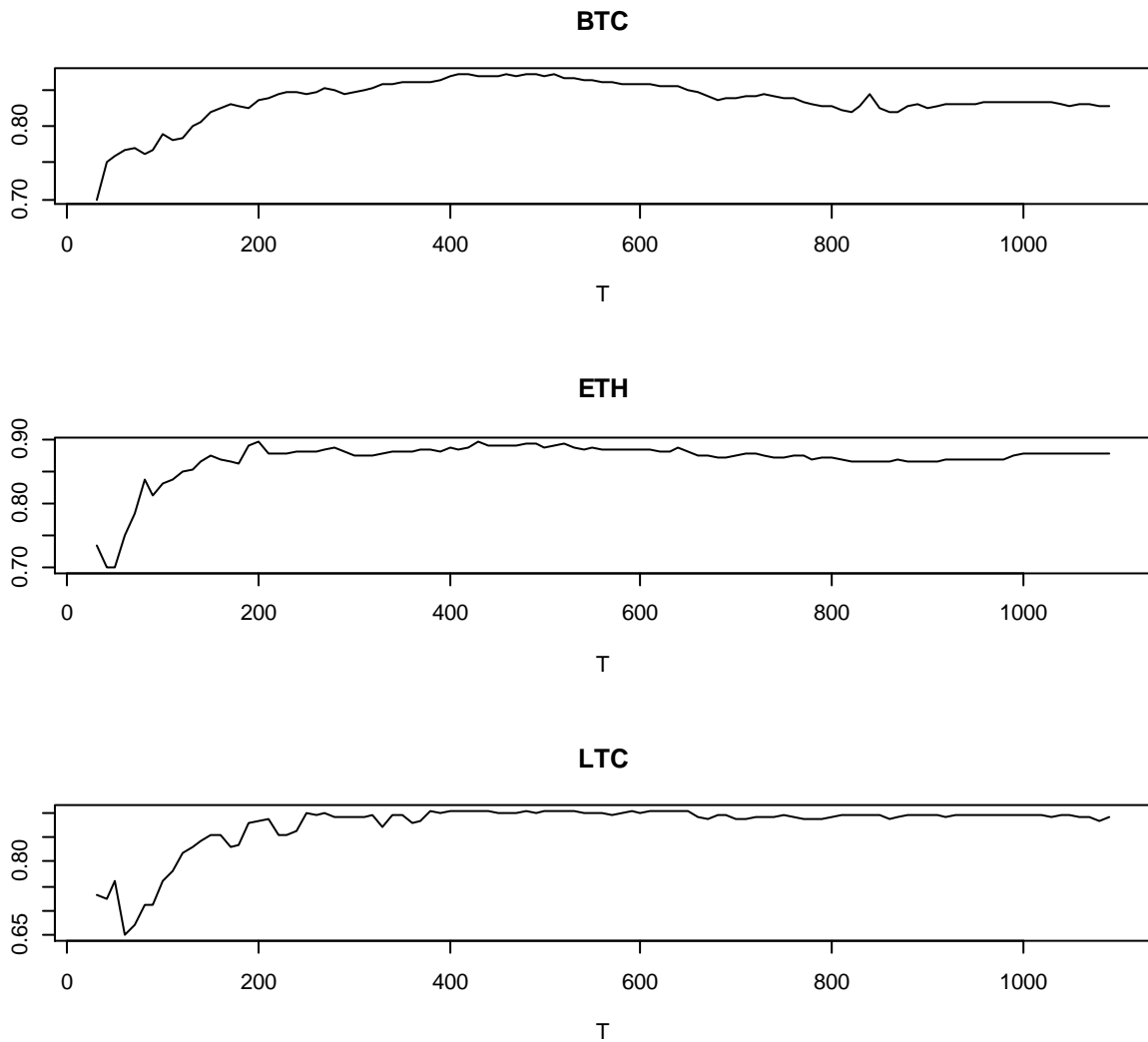
Apylygis visoms kriptovaliutoms yra F1 įvertis. Eterio ir laitkoino atveju jis sutampa. Mažiausia vertė fiksuojama bitkoinui.

Palyginus apmokyto HMM modelio tikslumo rodiklių reikšmes su rodikliais gautais testavimo prognozės metu ryškių skirtumų tarp modelio kokybės matų verčių nėra. Vidutinis apmokyto modelio taiklumas yra 86,6 %, o atliktos prognozės – 86,3 %. Vidutinis jautrumas testavimo duomenims lygus 0,962 ir yra šiek tiek didesnis už apmokymo imties (0,95). Modelis apmokytas vidutiniškai 0,821 tikslumu, prognozuotas – 0,810.

### **3.3.4. HMM modelio tikslumo priklausomybė nuo stebėjimų sekos dydžio**

Įvertinti, ar modelis nebuvo permokytas (*angl.* overfitted) dėl per didelės duomenų imties, tirta HMM modelio tikslumo priklausomybė nuo stebėjimo sekos dydžio.

Modelis pritaikytas pradėdant nuo 30 stebėjimo sekos narių. Stebėjimų skaičius kiekvieną kartą padidinamas 10 stebėjimų ir iš naujo pritaikomas modelis. Nubraižytas modelio taiklumo mato priklausomybės grafikas kiekvienai kriptovaliutai (žr. 14 pav.)



**14 pav.** Modelio tikslumo (*angl. accuracy*) priklausomybė nuo stebėjimo sekos skaičiaus

Modelio taiklumo rodiklio tikslios reikšmės pateiktos 5 priede. Visoms kriptovaliutomis su maža stebėjimu imtini taiklumo rodiklis turėjo žemas reikšmes. Palaipsniui didėjant stebėjimų skaičiui modelio taiklumo vertė taip pat didėjo. Kai modeliui buvo paduota imtis turinti nuo 400 iki maždaug 600 stebėjimų taiklumas buvo pasiekęs itin dideles reikšmes. Toliau tendencijos šiek tiek išsiskiria priklausomai nuo kriptovaliutos. Laikotarpio taiklumas išlieka panašus toliau didinant stebėjimų kiekį, tuo tarpu bitkoino ir eterio šio rodiklio reikšmės pradeda mažėti. Tiesa spartesniu rodiklio kritimu pasižymi bitkoinas.

### 3.3.5. Agregavimo lygio įtaka rezultatams

Sudarytas modelis pritaikytas ir agreguotiems duomenims. Dieniniai duomenys pakeisti į 3 dienų, 5 dienų ir savaitinius duomenis skaičiuojant atitinkamo laikotarpio vidurkius. Paslėptųjų būsenų

prognozė kainų kryptims atitinkamai gaunama nurodanti trijų dienų, penkių dienų ir savaitės pokyčius. Kiekvienos kriptovaliutos pritaikyto modelio rezultatai pateikti 6 priede.

**8 lentelė.** Modelio tikslumas agreguotiems duomenims

	<b>kasdieniniai</b>	<b>3 dienų</b>	<b>5 dienų</b>	<b>savaitės</b>
<b>Apmokymas</b>				
<b>Taiklumas</b>	0,866	0,908	0,934	0,946
<b>Kappa</b>	0,735	0,816	0,867	0,893
<b>Jautrumas</b>	0,950	0,996	0,921	0,990
<b>Specifiškumas</b>	0,792	0,832	0,948	0,909
<b>Tikslumas</b>	0,821	0,835	0,945	0,904
<b>F1 įvertis</b>	0,874	0,908	0,929	0,945
<b>Testavimas</b>				
<b>Taiklumas</b>	0,863	0,931	0,921	0,939
<b>Kappa</b>	0,727	0,860	0,841	0,880
<b>Jautrumas</b>	0,962	1,000	0,833	1,000
<b>Specifiškumas</b>	0,773	0,884	1,000	0,889
<b>Tikslumas</b>	0,810	0,853	1,000	0,889
<b>F1 įvertis</b>	0,873	0,921	0,905	0,939

Vidutiniai agreguotų duomenų modelio tikslumo rezultatai pateikti 8 lentelėje. Galima pastebėti, kad vidutinis apmokyto modelio taiklumas didėja keičiant kasdieninius duomenis į trijų dienų, penkių dienų ir savaitinius duomenis. Atliekant testavimą ir prognozuojant kainos kitimo kryptį modelio taiklumas didžiausiais taikant savaitinius duomenis ir yra lygus 94 %. Be to, didžiausia F1 įverčio reikšmė taip pat pasiekta imant stebėjimų duomenis kas savaitę.

Didžiausiais jautrumas gautas naudojant 3 dienų ir savaitinius duomenis, kuris testuotiems duomenims yra lygus 1, tai reiškia, kad visi kainų mažėjimai buvo teisingai prognozuoti.

Pagal specifiškumo ir tikslumo rodiklius didžiausios vertės fiksuojamos 5 dienų duomenims.

Apibendrinant, išskirti vieno modelio, kuris būtų tinkamiausias prognozei nebūtų teisinga, kadangi vidutiniai prognozavimo tikslumo rodikliai tiek 3 dienų, tiek savaitinių duomenų yra labai panašūs. Todėl kitame skyrelyje prognozės tikslumo rezultatai palyginami su kitais mašininio mokymosi metodais.



### 3.3.6. HMM ir kitų mašininio mokymosi metodų tikslumo palyginimas

HMM modelio gerumas įvertinamas palyginant prognozės tikslumą su kitais mašininio mokymosi metodais. Palyginimui pasirinkti atsitiktinių miškų (*angl.* random forest – RF) ir dirbtinių neuroninių tinklų (*angl.* neural network – NN) algoritmai bei naivus Bajeso klasifikatorius (*angl.* naive Bayes classifier – NB). Visų keturių metodų rezultatai kiekvienos kriptovaliutos testavimo duomenų imčiai pateikti 4 priede, o kiekvieno algoritmo vidutinės tikslumo rodiklio reikšmės pateiktos 9 lentelėje.

9 lentelė. Skirtingų metodų vidutiniai prognozės tikslumo rodikliai

	HMM	RF	ANN	NB	HMM	RF	NN	NB
	kasdieniniai				3 dienų			
<b>Taiklumas</b>	0,863	0,844	0,844	0,759	0,931	0,966	0,966	0,759
<b>Kappa</b>	0,727	0,693	0,693	0,510	0,860	0,929	0,929	0,448
<b>Jautrumas</b>	0,962	0,862	0,862	0,500	1,000	0,923	0,923	0,406
<b>Specifiškumas</b>	0,773	0,843	0,843	1,000	0,884	1,000	1,000	1,000
<b>Tikslumas</b>	0,810	0,855	0,855	1,000	0,853	1,000	1,000	1,000
<b>F1 įvertis</b>	0,873	0,832	0,832	0,664	0,921	0,957	0,957	0,574
	5 dienų				savaitės			
<b>Taiklumas</b>	0,921	0,921	0,921	0,765	0,939	0,939	0,939	0,757
<b>Kappa</b>	0,841	0,844	0,844	0,528	0,880	0,880	0,880	0,501
<b>Jautrumas</b>	0,833	0,926	0,926	0,602	1,000	1,000	1,000	0,600
<b>Specifiškumas</b>	1,000	0,926	0,926	0,926	0,889	0,889	0,889	0,889
<b>Tikslumas</b>	1,000	0,926	0,926	0,889	0,889	0,889	0,889	0,833
<b>F1 įvertis</b>	0,905	0,919	0,919	0,714	0,939	0,939	0,939	0,695

Vertinant skirtingus metodus tarpusavyje pastebima, kad atsitiktinių miškų ir dirbtinių neuroninių tinklų metodai turi identišką rodiklių reikšmės. Naivus Bajeso klasifikatoriaus taiklumo, Kappa, jautrumo ir F1 reikšmės yra ženkliai mažesnės palyginus su kitais algoritmais. Geriausiai Bajeso klasifikatorius prognozavo 5 dienų duomenims, nors bendras modelio tikslumas tesiekia 76,5 %.

Lyginant visus modelius tarpusavyje HMM modelis vidutiniškai yra taikliausias (86,3 %) kai turimi kasdieniniai stebėjimai. Vis tik, trijų dienų duomenims geresnis prognozės tikslumas gautas panaudojus atsitiktinių miškų ir neuroninių tinklų algoritmus, kurių bendras modelio tikslumas siekia 96,6 %. O 5 dienų ir savaitiniams duomenims visi šie trys metodai turi tokį patį taiklumą atitinkamai lygų 92,1 % ir 93,9 %.

### 3.4. Tyrimo apibendrinimas ir rekomendacijos

Tyrimo metu buvo pritaikytas paslėptasis Markovo modelis kriptovaliutų kainų gražoms bei sudarytas diskretusis modelis kainos krypties prognozei atlikti ir modelio tikslumui įvertinti.

Statistinės analizės pagalba galima nesunkiai atskirti kuriam tipui priklauso konkreti kriptovaliuta. Kintamos vertės kriptovaliutos pasižymi didesniu standartiniu nuokrypiu (0,045 – 0,064) lyginant su teterio standartiniu nuokrypiu (0,0067), todėl kintamos vertės valiutų gražos yra labiau išsibarsčiusios aplink vidurkį. Variacijos koeficientas kintamos vertės valiutoms siekia net iki 35 % visu analizuojamu laikotarpiu, o imant mažesnius intervalus šis rodiklis gali būti dar didesnis. Tai tik patvirtina, kad tokio tipo kriptovaliutos pasižymi dideliu kintamumu. Stabilios kriptovaliutos vidurkis yra lygus 0, kadangi kainos svyruoja apie fiksuotą vertę.

Apskaičiuotoms gražoms pritaikius paslėptąjį Markovo modelį nustatyta, kad optimalu rinktis 3 būsenų modelį norint įvertinti kriptovaliutų kainas veikiančius režimus. Visų valiutų pradinė būsena pateko į režimą, kurio metu kainos svyruoja minimaliai arba yra stabilios. Taip pat išskirtas režimas, kuriame papuola ekstremaliai didelio kintamumo gražos. Bei tarpinis režimas su įprastu gražų svyravimu. Režimai labai dažnai keičiasi, kas taip pat gali indikuoti, kad kriptovaliutų kainos yra labai nepastovios. Norint nustatyti režimus kriptovaliutų kainoms stebėjimų skaičius turėtų būti imamas kuo didesnis, tada yra gaunami geresni rezultatai atsižvelgiant į AIC ir BIC kriterijus.

Diskretizavus kriptovaliutų gražas į 5 stebėjimų intervalus ir atlikus kainos krypties prognozę gauti pakankamai tikslūs modeliai. Vidutiniškai modelio taiklumas lygus 86,3 %. Atliktas duomenų agregavimas pagerino modelio tikslumą. Vertėtų rinktis 3 dienų arba savaitinius duomenis naudojant HMM modeliavimą kainų krypties prognozei. Jeigu norima atlikti trumpesnio laikotarpio krypties prognozę, mano manymu, tinkamiausia būtų imti 3 dienų duomenis, o jeigu planuojama atlikti ilgesnio laikotarpio – tuomet rinktis savaitinius duomenis.

Ištyrus HMM kainos krypties prognozės tikslumą imant skirtingo ilgio stebėjimo sekas, pastebėta, kad imant labai mažą stebėjimų imtį prognoze nėra pakankamai tiksli. Vis tik, šiame tyrime buvo galima gauti didesnę modelio tikslumą, jeigu modelio apmokymui būtų buvusi paimta trumpesnė stebėjimų seka.

Palyginant su kitais mašininio mokymosi algoritmais HMM modelis pademonstravo šiek tiek geresnius rezultatus prognozuojant kainų kryptį turint kiekvienos dienos stebėjimus. Atsitiktinių miškų ir neuroninių tinklų metodai davė tikslesnę prognozę 3 dienų duomenims. Su dar stambesniais stebėjimų intervalais metodų tikslumas gautas vienodas.

Tolimesnės rekomendacijos tyrimas susijusiais su kriptovaliutų kainos krypties prognozę HMM metodu:

1. Imti mažesnę stebėjimų kiekį apmokymo imčiai atliekant kriptovaliutų kainos krypties prognozę HMM metodu, taip bus gautas tikslesnis prognozės modelis.
2. Atlikti tyrimą kainos krypties prognozei sudarant HMM hibridinius metodus su kitais mašininio mokymosi algoritmais, kad nustatyti, ar galima pagerinti prognozės tikslumą.
3. Atlikti tyrimą, kuriame būtų įvertinamas kriptovaliutų kainos kitimo procesas su atmintimi.
4. Sudaryti kainos krypties prognozavimo modelį imant ne su kaina susijusius stebėjimus.
5. Atlikti tyrimą sudarant HMM skirtingoms investavimo strategijoms ir pažiūrėti, kaip jas paveikia režimų pasikeitimai.

### 3.5. Programinė realizacija

Tyrime visi skaičiavimai buvo atlikti atviro kodo R programa. Pasirinkimą nulėmė tai, kad ši programa yra laisvai prieinama ir turi didelę statistinių paketų įvairovę. Programos kodas pateiktas priede 7, o žemiau išvardinti programinės įrangos paketai, kurie naudoti atliekant tyrimą:

- *quantmod* – šiuo paketu iš *Yahoo!Finance* platformos gauti duomenys apie kriptovaliutų kiekvienos dienos kainą;
- *Summarytools*, *PerformanceAnalytics* ir *Pastecs* – naudoti pagrindinėms statistikoms apskaičiuoti;
- *depmixS4* paketu atliktas paslėptojo Markovo modelio pritaikymas kriptovaliutų kainų gražoms, kad nustatyti optimalų būsenų skaičių bei nustatyti paslėptuosius rinkos režimus;
- *markovchain* ir *sps* – skirti Markovo savybių tikrinimui. *Markovchain* pagalba apskaičiuotos perėjimo tikimybių matricos;
- *HMM* – paketas, kuris naudotas diskretaus paslėptojo Markovo modelio apmokymui ir prognozei atikti;
- *Caret* – panaudotas apskaičiuoti sumaišymo matricai ir kitiems tikslumo rodikliams

## Išvados

- Taikant tolydų paslėptųjų Markovo modelį kriptovaliutų rinkos režimų nustatymui, gauta, kad atsižvelgiant į AIC ir BIC informacinių kriterijų reikšmes yra optimalu naudoti 3 būsenų metodą kriptovaliutų kainų grąžų duomenims. Paslėptieji režimai skirstomi pagal kriptovaliutų kainų kintamumą į žemo, vidutinio ir aukšto kintamumo režimus. O praktikoje tai reiškia, kad egzistuoja tokie rinkos režimai, kurie turi įtakos kriptovaliutų kainos svyravimams.
- Diskretusis dviejų būsenų paslėptasis Markovo modelis vidutiniškai 86,3 % taiklumu atlieka kriptovaliutų kainų krypties prognozę, kai turimi kiekvienos dienos stebėjimai. Tai rodo, kad modelis gali prognozuoti kainos kryptis pakankamai gerai.
- Apskaičiavus sumaišymo matricą prognozės tikslumui įvertinti nustatyta, kad dažniausiai pasitaikanti prognozės klaida buvo tokia, kai prognozuota, jog kaina mažėja, nors iš tikrųjų kaina didėjo.
- Taikant palyginamąjį metodų tikslumo analizę nustatyta, kad paslėptasis Markovo modelis (su 86,3 % taiklumu) yra tinkamesnis atlikti kasdienių duomenų kainos krypties prognozę nei kiti mašininio mokymosi metodai: atsitiktinių miškų metodas, neuroninių tinklų metodas ar naivus Bajeso klasifikatorius.
- Atlikus modelio tikslumo priklausomybę nuo stebėjimų sekos dydžio nustatyta, kad didžiausiais vidutinis tikslumas (89 %) gautas, kai stebėjimų seka turėjo nuo 400 iki 530 narių. Tai reiškia, kad apmokymui reikalinga turėti daugiau nei vienerių metų stebėjimus apie kiekvienos dienos kriptovaliutų kainą.
- Taikant duomenų agregavimą nustatyta, kad modelis su 3 dienų ir savaitiniais duomenimis atlieka kainos krypties prognozę vidutiniškai geriau nei imant kasdieninius ar 5 dienų duomenis. 3 dienų modelio taiklumo rodiklio reikmė vidutiniškai yra 93,1 %, o savaitinių duomenų – 93,9 %. Praktikoje tai reiškia, kad kiekvienos dienos duomenų stebėjimai pasižymi didesniu kintamumu, todėl tokią prognozę yra daug sunkiau atlikti.

## Literatūros sąrašas

1. ANG Andrew ir Allan TIMMERMANN. Regime Changes and Financial Markets. Journal. *Annual Review of Financial Economics, Annual Reviews*, 2012, vol. 4(1), pages 313-337. Ang, Andrew and Timmermann, Allan, Regime Changes and Financial Markets (October 2012). Annual Review of Financial Economics, Vol. 4, pp. 313-337, 2012. Prieiga per: <http://dx.doi.org/10.1146/annurev-financial-110311-101808>
2. Audito, apskaitos, turto vertinimo ir nemokumo valdymo tarnyba: Kripto valiutos ir žetonų apskaitos rekomendacijos [interaktyvus]. 2018 [žiūrėta 2020-04-18] Prieiga per: <http://www.avnt.lt/assets/Veiklos-sritys/Apskaita/VAS/Euras-ir-kripto-valiuta/Kripto-valiutos-ir-eton-apskaitos-rekomendacijosskelbimui06-12.pdf>
3. Barauskaitė, Ieva. Vartotojo identifikavimo naudojant balso atpažinimą metodų tyrimas. *Magistrinis darbas* [interaktyvus]. Kauno technologijos universitetas, 2017 [žiūrėta 2020-03-22]. Prieiga per: <https://epubl.ktu.edu/object/elaba:22733722/>
4. Beigel, O., Who Accepts Bitcoin as Payment? [interaktyvus]. 2020 [žiūrėta 2020-04-16] Prieiga per: <https://99bitcoins.com/bitcoin/who-accepts/>
5. CHAPPELL, Daniel, Regime Heteroskedasticity in Bitcoin: A Comparison of Markov Switching Models [interaktyvus]. 2018 [žiūrėta 2020-05-03]. Available at <http://dx.doi.org/10.2139/ssrn.3290603>
6. CHOI, H., D. FERMIN, A. I. NESVIZHSHKII, D. GHOSH ir Z. S. QIN, Sparsely correlated hidden Markov models with application to genome-wide location studies. *Bioinformatics (Oxford, England)* [interaktyvus]. 2013, 29(5), 533–541 [žiūrėta 2020-04-01]. Prieiga per: <https://doi.org/10.1093/bioinformatics/btt012>
7. CHOI, Sangyup and Shin, JUNHYEOK, Brave New World? Bitcoin is not the New Gold: Understanding Cryptocurrency Price Dynamics [interaktyvus]. 2020 [žiūrėta 2020-04-19]. Prieiga per: <http://dx.doi.org/10.2139/ssrn.3555599>
8. DIPPLE, Stephen, Abhishek CHOUDHARY, James FLAMINO, Boleslaw SZYMANSKI ir G. KORNISS. Using correlated stochastic differential equations to forecast cryptocurrency rates and social media activities. *Applied Network Science* [interaktyvus]. 2020 [žiūrėta 2020-04-11]. Prieiga per: <https://doi.org/10.1007/s41109-020-00259-1>
9. ERLWEIN Christina, Gautam MITRA ir Diana ROMAN. HMM based scenario generation for an investment optimisation problem, *Annals of Operations Research, Springer*, [interaktyvus], 2012 vol. 193(1), pages 173-192 [žiūrėta 2020-03-26]. Prieiga per: <https://doi.org/10.1007/s10479-011-0865-8>
10. FANG, Fan, Carmine VENTRE, Michail BASIOS, Hoilong KONG, Leslie KANTHAN, Lingbo LI, David MARTINEZ-REGOBAND ir Fan WU. Cryptocurrency Trading: A Comprehensive Survey [interaktyvus]. 2020 [žiūrėta 2020-04-15]. Prieiga per: [https://www.researchgate.net/publication/340172901\\_Cryptocurrency\\_Trading\\_A\\_Comprehensive\\_Survey](https://www.researchgate.net/publication/340172901_Cryptocurrency_Trading_A_Comprehensive_Survey)
11. Finblog: Bitkoino kaina 2020 metais – ko tikėtis? Ekspertų nuomonės [interaktyvus]. 2020 [žiūrėta 2020-04-18]. Prieiga per: <https://www.finblog.lt/2020/01/03/bitcoino-kaina-2020-metais-ko-tiketis-ekspertu-nuomones/>
12. GYAMERAH, Samuel Asante. On forecasting the intraday Bitcoin price using ensemble of variational mode decomposition and generalized additive model. *Journal of King Saud University - Computer and Information Science*. ISSN 1319-1578, Prieiga per: <https://doi.org/10.1016/j.jksuci.2020.01.006>
13. GUPTA, A. ir B. DHINGRA. Stock market prediction using Hidden Markov Models. *2012 Students Conference on Engineering and Systems, Allahabad, Uttar Pradesh*, 2012, pp. 1-4.
14. HASSAN, M R. ir B. NATH. Stock Market Forecasting Using Hidden Markov Model: A New Approach. *Proceedings - 5th International Conference on Intelligent Systems Design and Applications*, 2005, ISDA '05. 2005. 192 - 196. Prieiga per: doi: 10.1109/ISDA.2005.85.

15. Yahoo! Finance: Top 10 cryptocurrencies by market capitalisation [interaktyvus]. 2020 [žiūrėta 2020-04-15] Prieiga per: <https://finance.yahoo.com/news/top-10-cryptocurrencies-market-capitalisation-160046487.html>
16. Investopedia: Average True Range - ATR Definition [interaktyvus]. 2019 [žiūrėta 2020-05-10] Prieiga per internetą: <https://www.investopedia.com/terms/a/atr.asp>
17. YUAN, Yuan ir Gautam MITRA. Market Regime Identification Using Hidden Markov Models [interaktyvus]. 2016 [žiūrėta 2020-05-10] Prieiga per: <http://dx.doi.org/10.2139/ssrn.3406068>
18. JAKŠTAS, Artūras. Vairuotojo elgsenos modeliavimas Panaudojant dirbtinį intelektą. *Magistrinis darbas* [interaktyvus]. Vytauto Didžiojo Universitetas, 2008 [žiūrėta 2019-12-22]. Prieiga per: <https://vb.vdu.lt/object/elaba:1765833/1765833.pdf>
19. KAVITHA, G., U. ANNAMALAI ir N. DEIVANAYAGAMPILLAI. Stock Market Trend Analysis Using Hidden Markov Models [interaktyvus]. 2013 [žiūrėta 2019-12-21]. Prieiga per: [https://www.researchgate.net/publication/258631733\\_Stock\\_Market\\_Trend\\_Analysis\\_Using\\_Hidden\\_Markov\\_Models](https://www.researchgate.net/publication/258631733_Stock_Market_Trend_Analysis_Using_Hidden_Markov_Models)
20. KY, Dao Xuan ir Luc Tri TUYEN. A Higher order Markov model for time series forecasting. *International Journal of Applied Mathematics and Statistics*, 2018, Vol. 57, pp.1-18, ISSN 0973-7545
21. KIM, E.-C., H.-W. Jeong, N.-Y. Lee. Global Asset Allocation Strategy Using a Hidden Markov Model. *J. Risk Financial Manag* [interaktyvus]. 2019 [žiūrėta 2019-12-21]. 12(4), 168. Prieiga per: <https://doi.org/10.3390/jrfm12040168>
22. KOUEMOU, G., L. History and Theoretical Basics of Hidden Markov Models, Hidden Markov Models. *Theory and Applications, Przemyslaw Dymarski, IntechOpen* [interaktyvus]. 2011 [žiūrėta 2020-03-22]. Prieiga per: doi: 10.5772/15205.
23. KRAAIJEVELD, Olivier ir Johannes De SMEDT. The predictive power of public Twitter sentiment for forecasting cryptocurrency prices. *Journal of International Financial Markets, Institutions and Money*, Volume 65, 2020, 101188, ISSN 1042-4431, Prieiga per: <https://doi.org/10.1016/j.intfin.2020.101188>.
24. LI, N. Hidden Markov Model and Financial Application. *Master of Science in Statistics* [interaktyvus]. 2016 [žiūrėta 2020-04-10], The University of Texas at Austin. Prieiga per: <https://repositories.lib.utexas.edu/handle/2152/47043>
25. Lietuvos bankas: LBChain [interaktyvus]. 2020 [žiūrėta 2020-04-15]. Prieiga per internetą: <https://www.lb.lt/lt/lbchain>
26. Lietuvos bankas: Skaitmeninė kolekcinė moneta (LBCoin) [interaktyvus]. 2020 [žiūrėta 2020-04-15]. Prieiga per internetą: <https://www.lb.lt/lt/skaitmenine-kolekcinė-moneta-lbcoin>
27. MAITI, Moinak, Yaroslav VYKLYUK ir Darko VUKOVIC. Cryptocurrencies Chaotic Co-movement Forecasting with Neural Networks. *Internet Technology Letters* [interaktyvus]. 2020 [žiūrėta 2020-04-11] Prieiga per: doi: 10.1002/itl2.157
28. Master The Crypto: Crypto Volatility: Why Volatility Is Important In The Cryptocurrency Market [interaktyvus]. 2020 [žiūrėta 2020-04-18]. Prieiga per internetą: <https://masterthecrypto.com/crypto-volatility-important-cryptocurrency-market/>
29. NGUYEN, Nguyet. An Analysis and Implementation of the Hidden Markov Model to Technology Stock Prediction. *Risks* [interaktyvus]. 2017 [žiūrėta 2020-03-17] 5(4):62. Prieiga per: doi: 10.3390/risks5040062.
30. NGUYEN, Nguyet. Hidden Markov Model for Stock Trading. *International Journal of Financial Studies* [interaktyvus]. 2018 [žiūrėta 2020-05-03] 6(2):36. Prieiga per: <http://dx.doi.org/10.3390/ijfs6020036>
31. PHILLIPS, R. C. Ir D. GORSE. Predicting cryptocurrency price bubbles using social media data and epidemic modelling. *2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI* [interaktyvus]. 2017 [žiūrėta 2020-03-19], pp. 1-7. Prieiga per: doi: 10.1109/SSCI.2017.8280809.

32. SASIKUMAR, R. ir S. ABDULLAH. Forecasting The Stock Market Values Using Hidden Markov Model. *International Journal of Business Analytics and Intelligence* [ininteraktyvus]. 2016 [žiūrėta 2019-12-21] Volume 4 Issue 1. Prieiga per: doi: 10.21863/ijbai/2016.4.1.018.
33. STEHMAN, Stephen. Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*. 1997, Volume 62, Issue 1, Pages 77-89, ISSN 0034-4257
34. SUDA, D. ir L. SPITERI. Analysis and Comparison of Bitcoin and S and P 500 Market Features Using HMMs and HSMMs. *Information* [interaktyvus]. 2019 [žiūrėta 2020-03-22] 10(10):322. Prieiga per: doi: 1-30. 10.3390/info10100322.
35. Tether.to: About Us [interaktyvus]. 2020 [žiūrėta 2020-04-18]. Prieiga per: <https://tether.to/about-us/>
36. TOLEDO, Tomer ir Romina KATZ. State Dependence in Lane-Changing Models. *Transportation Research Record* [interaktyvus]. 2009 [žiūrėta 2020-05-10]. Vol. 2124, no. 1, pp. 81–88, Prieiga per: doi:10.3141/2124-08.
37. VAIČIULYTĖ, Jūratė ir Gražvydas FELINSKAS. Paslėptųjų Markovo modelių metodo tyrimas ir taikymas balso įrašams stenografuoti. *Jaunųjų Mokslininkų Darbai*, 2016, T. 1, nr. 45, pp. 71-78, Prieiga per: doi:10.21277/jmd.v1i45.40.
38. VAIČIULYTĖ, Jūratė ir Leonidas Sakalauskas. Rekurentinis paslėptųjų Markovo modelių parametrų vertinimo algoritmas. *Computational Science and Techniques* [interaktyvus]. 2017 [žiūrėta 2020-03-13]. Vol. 5, no 1. Prieiga per: doi: 10.15181/csat.v5i1.1561.
39. VAIČIULYTĖ, Jūratė. Automatinio šnekos atpažinimo metodų tyrimas ir taikymai balso įrašams stenografuoti, *Magistrinis darbas* [interaktyvus]. 2015 [žiūrėta 2020-03-10]. Šiaulių universitetas. Prieiga per: <http://gs.elaba.lt/object/elaba:8674571/>

Priedai

1 priedas

Bitkoino, eterio, laitkoino ir teterio charakteristikos pagal metus

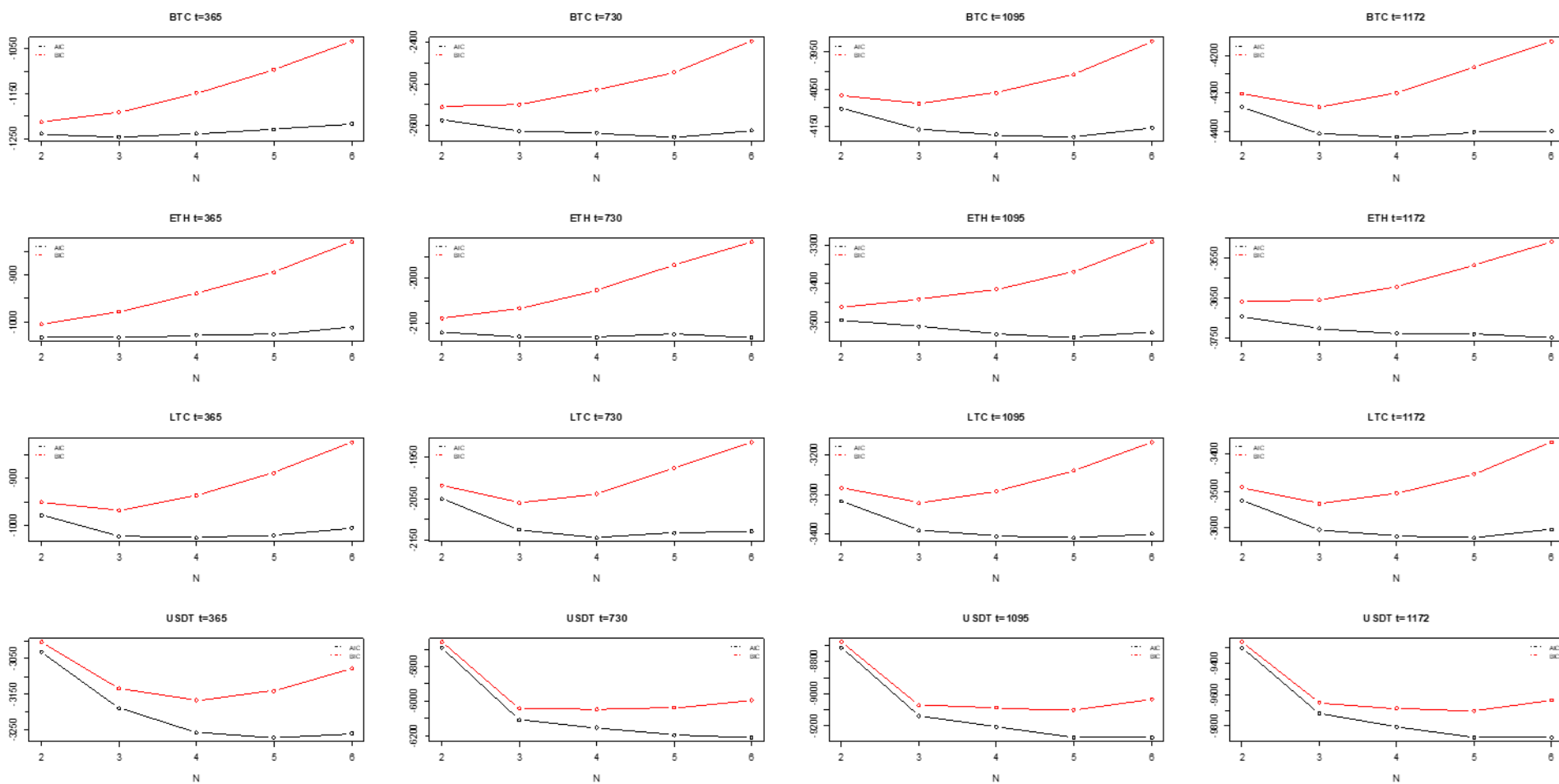
	Imties dydis	Nulinių gražų kiekis	Vidurkis	Standartinis nuokrypis	CV	Mažiausia reikšmė	Mediana	Didžiausia reikšmė	MAD	Asimetrijos koeficientas	Eksceso koeficientas	Pusiau dispersija
<b>Bitkoinas</b>												
2017	364	1	0,00729	0,04935	6,77363	-0,20753	0,00878	0,22512	0,03468	0,04185	3,20068	0,05110
2018	365	0	-0,00364	0,04290	-11,76886	-0,18458	0,00079	0,12413	0,02725	-0,44585	1,97980	0,04846
2019	365	0	0,00179	0,03533	19,73644	-0,15182	0,00121	0,16004	0,02070	0,22466	4,35155	0,03362
2020 I ketv.	91	0	-0,00122	0,06265	-51,42315	-0,46473	-0,00176	0,16710	0,03092	-4,12943	30,89933	0,07475
2017 - 2020 I ketv.	1185	1	0,00157	0,04487	28,52623	-0,46473	0,00204	0,22512	0,02783	-0,91352	12,17475	0,04751
<b>Eteris</b>												
2017	364	0	0,01244	0,07042	5,66095	-0,31542	0,00368	0,29014	0,03780	0,47735	3,65459	0,05883
2018	365	0	-0,00476	0,05657	-11,89500	-0,20686	-0,00257	0,16615	0,04117	-0,29568	1,27970	0,06058
2019	365	0	-0,00008	0,04139	-	-0,18326	-0,00076	0,14500	0,02704	-0,44658	3,56086	0,04202
2020 I ketv.	91	0	0,00033	0,07839	235,68985	-0,55073	0,00189	0,17345	0,03774	-3,64775	24,95946	0,09890
2017 - 2020 I ketv.	1185	0	0,00236	0,05956	25,25985	-0,55073	0,00017	0,29014	0,03522	-0,43295	9,37904	0,05780
<b>Laitkoinas</b>												
2017	364	0	0,01083	0,08046	7,43257	-0,39503	0,00121	0,51142	0,04105	1,23817	8,26893	0,06356
2018	365	0	-0,00556	0,05561	-9,99742	-0,21187	-0,00535	0,29059	0,04353	0,33126	2,89360	0,05470
2019	365	0	0,00084	0,04766	57,01352	-0,18028	-0,00016	0,26872	0,03615	0,67349	4,81408	0,04373
2020 I ketv.	91	0	-0,00056	0,07317	-131,51182	-0,44906	0,00244	0,19093	0,04333	-2,29062	14,24199	0,08452
2017 - 2020 I ketv.	1185	0	0,00183	0,06390	34,98449	-0,44906	-0,00097	0,51142	0,03948	0,75422	10,27061	0,05726
<b>Teteris</b>												
2017	364	38	0,00003	0,00893	-	-0,04820	0,00000	0,05661	0,00287	0,29024	10,19211	0,00791
2018	365	2	0,00000	0,00544	-	-0,01915	0,00001	0,02698	0,00356	0,25767	3,70280	0,00535
2019	365	0	-0,00004	0,00396	-	-0,01426	-0,00024	0,01503	0,00308	0,18036	1,90705	0,00372
2020 I ketv.	91	0	0,00004	0,00956	-	-0,05257	-0,00011	0,05339	0,00304	0,28189	19,15257	0,00918
2017 - 2020 I ketv.	1185	40	0,00000	0,00673	-	-0,05257	0,00000	0,05661	0,00319	0,33133	17,53895	0,00635



## HMM modelių AIC, BIC ir tikėtinumo funkcijos reikšmės

T	N	BTC			ETH			LTC			USDT		
		AIC	BIC	logLik	AIC	BIC	logLik	AIC	BIC	logLik	AIC	BIC	logLik
365	2	-1239,51	-1212,21	626,75	-1032,31	-1005,01	523,16	-977,90	-950,60	495,95	-3030,65	-3003,35	1522,33
365	3	-1246,42	-1191,82	637,21	-1032,44	-977,84	530,22	-1022,67	-968,07	525,34	-3188,51	-3133,92	1608,26
365	4	-1238,93	-1149,23	642,46	-1028,13	-938,43	537,06	-1025,64	-935,94	535,82	-3257,05	-3167,35	1651,52
365	5	-1229,61	-1097,01	648,80	-1026,20	-893,60	547,10	-1020,63	-888,03	544,31	-3272,14	-3139,54	1670,07
365	6	-1217,30	-1034,01	655,65	-1011,06	-827,76	552,53	-1005,78	-822,49	549,89	-3260,79	-3077,49	1677,39
730	2	-2587,49	-2555,34	1300,75	-2121,70	-2089,55	1067,85	-2049,29	-2017,14	1031,64	-5692,81	-5660,66	2853,41
730	3	-2614,95	-2550,64	1321,47	-2131,81	-2067,51	1079,90	-2123,39	-2059,08	1075,69	-6105,31	-6041,01	3066,66
730	4	-2619,43	-2513,79	1332,72	-2131,82	-2026,18	1088,91	-2142,58	-2036,94	1094,29	-6152,44	-6046,80	3099,22
730	5	-2628,73	-2472,56	1348,36	-2125,59	-1969,43	1096,80	-2130,79	-1974,63	1099,39	-6193,01	-6036,85	3130,50
730	6	-2613,33	-2397,46	1353,67	-2132,33	-1916,45	1113,16	-2128,13	-1912,26	1111,06	-6209,15	-5993,28	3151,58
1095	2	-4101,74	-4066,75	2057,87	-3495,39	-3460,40	1754,70	-3318,39	-3283,40	1666,19	-8716,95	-8681,96	4365,47
1095	3	-4159,09	-4089,11	2093,55	-3510,67	-3440,69	1769,34	-3391,35	-3321,37	1709,68	-9138,44	-9068,46	4583,22
1095	4	-4173,74	-4058,77	2109,87	-3530,41	-3415,44	1788,21	-3406,85	-3291,89	1726,43	-9202,07	-9087,11	4624,04
1095	5	-4179,61	-4009,66	2123,81	-3539,03	-3369,08	1803,52	-3410,39	-3240,44	1739,20	-9271,42	-9101,47	4669,71
1095	6	-4154,67	-3919,74	2124,33	-3526,52	-3291,59	1810,26	-3401,35	-3166,42	1747,68	-9270,20	-9035,27	4682,10
1172	2	-4337,53	-4302,07	2175,77	-3695,03	-3659,57	1854,52	-3525,39	-3489,92	1769,69	-9297,65	-9262,18	4655,82
1172	3	-4407,37	-4336,44	2217,69	-3725,76	-3654,82	1876,88	-3605,02	-3534,09	1816,51	-9719,80	-9648,87	4873,90
1172	4	-4416,32	-4299,79	2231,16	-3737,78	-3621,25	1891,89	-3622,63	-3506,10	1834,32	-9803,38	-9686,85	4924,69
1172	5	-4403,33	-4231,07	2235,67	-3739,28	-3567,02	1903,64	-3626,32	-3454,06	1847,16	-9873,36	-9701,10	4970,68
1172	6	-4401,42	-4163,30	2247,71	-3747,19	-3509,06	1920,59	-3604,36	-3366,24	1849,18	-9873,23	-9635,11	4983,62

## HMM modelių AIC ir BIC (raudona kreivė) reikšmių grafikai



Modelio tikslumo rodiklio reikšmių lentelė skirtingų metodų palyginimui

	HMM				RF				ANN				NB			
	BTC	ETH	LTC	Vidutinis	BTC	ETH	LTC	Vidutinis	BTC	ETH	LTC	Vidutinis	BTC	ETH	LTC	Vidutinis
<b>Taiklumas</b>	0,822	0,844	0,922	<b>0,863</b>	0,789	0,844	0,900	<b>0,844</b>	0,789	0,844	0,900	<b>0,844</b>	0,700	0,811	0,767	<b>0,7593</b>
<b>Kappa</b>	0,641	0,696	0,844	<b>0,727</b>	0,581	0,696	0,801	<b>0,693</b>	0,581	0,696	0,801	<b>0,693</b>	0,408	0,595	0,528	<b>0,5103</b>
<b>Jautrumas</b>	1,000	1,000	0,886	<b>0,962</b>	0,587	1,000	1,000	<b>0,862</b>	0,587	1,000	1,000	<b>0,862</b>	0,413	0,564	0,523	<b>0,5000</b>
<b>Specifiškumas</b>	0,636	0,725	0,957	<b>0,773</b>	1,000	0,725	0,804	<b>0,843</b>	1,000	0,725	0,804	<b>0,843</b>	1,000	1,000	1,000	<b>1,0000</b>
<b>Tikslumas</b>	0,742	0,736	0,951	<b>0,810</b>	1,000	0,736	0,830	<b>0,855</b>	1,000	0,736	0,830	<b>0,855</b>	1,000	1,000	1,000	<b>1,0000</b>
<b>F1 įvertis</b>	0,852	0,848	0,918	<b>0,873</b>	0,740	0,848	0,907	<b>0,832</b>	0,740	0,848	0,907	<b>0,832</b>	0,585	0,721	0,687	<b>0,6643</b>

## Modelio tikslumo rodiklio reikšmių lentelė skirtingam stebėjimų sekos narių skaičiui

<b>T</b>	<b>BTC</b>	<b>ETH</b>	<b>LTC</b>	<b>T</b>	<b>BTC</b>	<b>ETH</b>	<b>LTC</b>
<b>30</b>	0,7000	0,7333	0,7333	<b>450</b>	0,8689	0,8911	0,9000
<b>40</b>	0,7500	0,7000	0,7250	<b>460</b>	0,8717	0,8913	0,9000
<b>50</b>	0,7600	0,7000	0,7600	<b>470</b>	0,8702	0,8894	0,9000
<b>60</b>	0,7667	0,7500	0,6500	<b>480</b>	0,8708	0,8917	0,9021
<b>70</b>	0,7714	0,7857	0,6714	<b>490</b>	0,8714	0,8939	0,9000
<b>80</b>	0,7625	0,8375	0,7125	<b>500</b>	0,8700	0,8880	0,9020
<b>90</b>	0,7667	0,8111	0,7111	<b>510</b>	0,8706	0,8902	0,9039
<b>100</b>	0,7900	0,8300	0,7600	<b>520</b>	0,8673	0,8923	0,9038
<b>110</b>	0,7818	0,8364	0,7818	<b>530</b>	0,8660	0,8868	0,9019
<b>120</b>	0,7833	0,8500	0,8167	<b>540</b>	0,8630	0,8833	0,9000
<b>130</b>	0,8000	0,8538	0,8308	<b>550</b>	0,8636	0,8855	0,9000
<b>140</b>	0,8071	0,8643	0,8429	<b>560</b>	0,8607	0,8839	0,9000
<b>150</b>	0,8200	0,8733	0,8533	<b>570</b>	0,8596	0,8842	0,8947
<b>160</b>	0,8250	0,8688	0,8562	<b>580</b>	0,8586	0,8845	0,9000
<b>170</b>	0,8294	0,8647	0,8294	<b>590</b>	0,8576	0,8831	0,9017
<b>180</b>	0,8278	0,8611	0,8333	<b>600</b>	0,8583	0,8850	0,8983
<b>190</b>	0,8263	0,8895	0,8789	<b>610</b>	0,8574	0,8836	0,9033
<b>200</b>	0,8350	0,8950	0,8850	<b>620</b>	0,8565	0,8806	0,9048
<b>210</b>	0,8381	0,8762	0,8857	<b>630</b>	0,8556	0,8810	0,9016
<b>220</b>	0,8455	0,8773	0,8545	<b>640</b>	0,8547	0,8859	0,9031
<b>230</b>	0,8478	0,8783	0,8565	<b>650</b>	0,8492	0,8800	0,9015
<b>240</b>	0,8458	0,8792	0,8625	<b>660</b>	0,8470	0,8758	0,8894
<b>250</b>	0,8440	0,8800	0,9000	<b>670</b>	0,8403	0,8731	0,8866
<b>260</b>	0,8462	0,8808	0,8962	<b>680</b>	0,8353	0,8721	0,8956
<b>270</b>	0,8519	0,8852	0,9000	<b>690</b>	0,8377	0,8710	0,8942
<b>280</b>	0,8500	0,8857	0,8929	<b>700</b>	0,8386	0,8743	0,8871
<b>290</b>	0,8448	0,8793	0,8931	<b>710</b>	0,8408	0,8761	0,8887
<b>300</b>	0,8467	0,8733	0,8900	<b>720</b>	0,8417	0,8778	0,8903
<b>310</b>	0,8484	0,8742	0,8903	<b>730</b>	0,8438	0,8753	0,8904
<b>320</b>	0,8531	0,8750	0,8938	<b>740</b>	0,8405	0,8716	0,8919
<b>330</b>	0,8576	0,8788	0,8697	<b>750</b>	0,8400	0,8720	0,8960
<b>340</b>	0,8588	0,8794	0,8941	<b>760</b>	0,8382	0,8750	0,8895
<b>350</b>	0,8600	0,8800	0,8971	<b>770</b>	0,8338	0,8740	0,8883
<b>360</b>	0,8611	0,8806	0,8778	<b>780</b>	0,8308	0,8692	0,8885
<b>370</b>	0,8595	0,8838	0,8811	<b>790</b>	0,8291	0,8709	0,8886
<b>380</b>	0,8605	0,8842	0,9026	<b>800</b>	0,8275	0,8700	0,8900
<b>390</b>	0,8641	0,8821	0,9000	<b>810</b>	0,8235	0,8679	0,8963
<b>400</b>	0,8675	0,8875	0,9025	<b>820</b>	0,8207	0,8659	0,8963
<b>410</b>	0,8707	0,8829	0,9049	<b>830</b>	0,8265	0,8639	0,8952
<b>420</b>	0,8714	0,8881	0,9048	<b>840</b>	0,8429	0,8643	0,8952
<b>430</b>	0,8698	0,8953	0,9047	<b>850</b>	0,8259	0,8659	0,8941
<b>440</b>	0,8682	0,8886	0,9045	<b>860</b>	0,8186	0,8663	0,8884

**Modelio tikslumo rodiklio reikšmių lentelė skirtingam stebėjimų sekos narių skaičiui (tęsinys)**

<b>T</b>	<b>BTC</b>	<b>ETH</b>	<b>LTC</b>	<b>T</b>	<b>BTC</b>	<b>ETH</b>	<b>LTC</b>
<b>870</b>	0,8207	0,8678	0,8897	<b>990</b>	0,8343	0,8758	0,8949
<b>880</b>	0,8284	0,8659	0,8943	<b>1000</b>	0,8340	0,8770	0,8940
<b>890</b>	0,8303	0,8663	0,8955	<b>1010</b>	0,8337	0,8772	0,8941
<b>900</b>	0,8256	0,8656	0,8944	<b>1020</b>	0,8324	0,8775	0,8941
<b>910</b>	0,8275	0,8659	0,8945	<b>1030</b>	0,8320	0,8786	0,8913
<b>920</b>	0,8293	0,8674	0,8924	<b>1040</b>	0,8308	0,8788	0,8952
<b>930</b>	0,8312	0,8688	0,8957	<b>1050</b>	0,8286	0,8762	0,8943
<b>940</b>	0,8309	0,8670	0,8947	<b>1060</b>	0,8302	0,8774	0,8925
<b>950</b>	0,8316	0,8674	0,8937	<b>1070</b>	0,8308	0,8785	0,8925
<b>960</b>	0,8323	0,8677	0,8948	<b>1080</b>	0,8287	0,8787	0,8815
<b>970</b>	0,8340	0,8691	0,8959	<b>1090</b>	0,8284	0,8780	0,8917
<b>980</b>	0,8337	0,8694	0,8949				

Modelio tikslumo rodiklio reikšmių lentelė skirtingų metodų palyginimui

	kasdieniniai				3 dienų				5 dienų				savaitės			
	BTC	ETH	LTC	Vidutinis	BTC	ETH	LTC	Vidutinis	BTC	ETH	LTC	Vidutinis	BTC	ETH	LTC	Vidutinis
<b>Apmokymo</b>																
Taiklumas	0,828	0,878	0,892	<b>0,866</b>	0,893	0,915	0,915	<b>0,908</b>	0,912	0,945	0,945	<b>0,934</b>	0,903	0,968	0,968	<b>0,946</b>
Kappa	0,664	0,755	0,785	<b>0,735</b>	0,787	0,830	0,830	<b>0,816</b>	0,826	0,888	0,888	<b>0,867</b>	0,807	0,936	0,936	<b>0,893</b>
Jautrumas	1,000	1,000	0,851	<b>0,950</b>	1,000	0,994	0,994	<b>0,996</b>	1,000	0,882	0,882	<b>0,921</b>	0,971	1,000	1,000	<b>0,990</b>
Specifiškumas	0,687	0,754	0,936	<b>0,792</b>	0,810	0,843	0,843	<b>0,832</b>	0,843	1,000	1,000	<b>0,948</b>	0,851	0,938	0,938	<b>0,909</b>
Tikslumas	0,724	0,804	0,934	<b>0,821</b>	0,802	0,851	0,851	<b>0,835</b>	0,835	1,000	1,000	<b>0,945</b>	0,835	0,938	0,938	<b>0,904</b>
F1 įvertis	0,840	0,891	0,891	<b>0,874</b>	0,890	0,917	0,917	<b>0,908</b>	0,910	0,938	0,938	<b>0,929</b>	0,898	0,968	0,968	<b>0,945</b>
<b>Testavimo</b>																
Taiklumas	0,822	0,844	0,922	<b>0,863</b>	0,931	0,931	0,931	<b>0,931</b>	1,000	0,882	0,882	<b>0,921</b>	1,000	0,909	0,909	<b>0,939</b>
Kappa	0,641	0,696	0,844	<b>0,727</b>	0,863	0,859	0,859	<b>0,860</b>	1,000	0,761	0,761	<b>0,841</b>	1,000	0,820	0,820	<b>0,880</b>
Jautrumas	1,000	1,000	0,886	<b>0,962</b>	1,000	1,000	1,000	<b>1,000</b>	1,000	0,750	0,750	<b>0,833</b>	1,000	1,000	1,000	<b>1,000</b>
Specifiškumas	0,636	0,725	0,957	<b>0,773</b>	0,875	0,889	0,889	<b>0,884</b>	1,000	1,000	1,000	<b>1,000</b>	1,000	0,833	0,833	<b>0,889</b>
Tikslumas	0,742	0,736	0,951	<b>0,810</b>	0,867	0,846	0,846	<b>0,853</b>	1,000	1,000	1,000	<b>1,000</b>	1,000	0,833	0,833	<b>0,889</b>
F1 įvertis	0,852	0,848	0,918	<b>0,873</b>	0,929	0,917	0,917	<b>0,921</b>	1,000	0,857	0,857	<b>0,905</b>	1,000	0,909	0,909	<b>0,939</b>

## Programos kodas

### Statistical Analysis.R

```

library(quantmod)

#Issitraukiami duomenys is Yahoo!Finance apie kriptovaliutu uzdarymo kaina
portfolio = c("BTC-USD","ETH-USD","LTC-USD", "USDT-USD") #,"XRP-USD","ADA-USD")
crypto <- getSymbols(portfolio, src="yahoo", from="2017-01-01")

data <- xts()
for(i in seq_along(crypto)) {
  symbol <- crypto[i]
  data <- merge(data, get(symbol)[,paste(symbol, "Close", sep=".")])
}

train.data <- data[1:1186,] #nuo 2017-01-01 iki 2020-03-31
test.data <- data[1187:1200,] #nuo 2020-04-01 iki 2020-04-14

#Funkcija apskaiciuojanti logaritmine gra?a
cal_log_returns <- function(d, d_set_name){
  r <- matrix(, nrow = dim(d)[1]-1, ncol = dim(d)[2])
  colnames(r) <- d_set_name
  for (j in 1:dim(d)[2]){
    for (i in 1:(dim(d)[1]-1)){
      r[i,j] <- log(as.numeric(d[i+1,j])/as.numeric(d[i,j]),exp(1))
    }
  }
  return(r)
}

returns <- cal_log_returns(train.data, portfolio)
currencies <- c("BTC", "ETH", "LTC", "USDT")
colnames(returns) <- currencies

#Braizomi kainos dinamikos ir gra?u grafikai
par(mfrow=c(4,2))
plot(data[, 'BTC.USD.Close'][1:1186,], type="l", main="BTC/USD", font=2)
plot(returns[, 'BTC'], type="l", xlab="", ylab="", main="BTC", cex.main=2.5)
plot(data[, 'ETH.USD.Close'][1:1186,], type="l", main="ETH/USD", cex.main=2.5)
plot(returns[, 'ETH'], type="l", xlab="", ylab="", main="ETH", cex.main=2.5)
plot(data[, 'LTC.USD.Close'][1:1186,], type="l", main="LTC/USD", cex.main=2.5)
plot(returns[, 'LTC'], type="l", xlab="", ylab="", main="LTC", cex.main=2.5)
plot(data[, 'USDT.USD.Close'][1:1186,], type="l", main="USDT/USD", cex.main=2.5)
plot(returns[, 'USDT'], type="l", xlab="", ylab="", main="USDT", cex.main=2.5)

```

```
#par(mfrow=c(2,2))
#plot(hist(returns[, 'BTC'], ylab="", main="BTC", col="lightblue")
#plot(density(returns[, 'ETH']), ylab="", main="ETH")
#plot(density(returns[, 'LTC']), ylab="", main="LTC")
#plot(density(returns[, 'USDT']), ylab="", main="USDT")
```

```
library(summarytools)
st_options(round.digits = 5)
stat <- summarytools::descr(returns)
view(stat)
```

```
library(PerformanceAnalytics)
round(SemiVariance(returns),5) #Pusiau dispersija
```

```
library(pastecs)
round(stat.desc(returns)[2,],0) #Nulines grazos
```

```
#Isskaidomi duomenys pagal metus
data17 <- data['2017/2017']
data18 <- data['2018/2018']
data19 <- data['2019/2019']
data20 <- data['2020/202003']
```

```
metai <- c("2017", "2018", "2019", "2020", "2017-2020")
returns17 <- returns[1:364,]
returns18 <- returns[365:729,]
returns19 <- returns[730:1094,]
returns20 <- returns[1095:1185,]
```

```
stat17 <- summarytools::descr(returns17)
stat18 <- summarytools::descr(returns18)
stat19 <- summarytools::descr(returns19)
stat20 <- summarytools::descr(returns20)
```

```
view(stat17)
round(SemiVariance(returns17),5) #Pusiau dispersija
round(stat.desc(returns17)[2,],0) #Nulines grazos
view(stat18)
round(SemiVariance(returns18),5) #Pusiau dispersija
round(stat.desc(returns18)[2,],0) #Nulines grazos
view(stat19)
round(SemiVariance(returns19),5) #Pusiau dispersija
round(stat.desc(returns19)[2,],0) #Nulines grazos
view(stat20)
round(SemiVariance(returns20),5) #Pusiau dispersija
```

## **Regime Switching.R**

```
library(quantmod)
library(depmixS4)
```



```

crypto <- new.env()
portfolio = c("BTC-USD", "ETH-USD", "LTC-USD", "USDT-USD") #, "XRP-USD", "ADA-USD")
getSymbols(portfolio, env=crypto, src="yahoo", from="2017-01-01")

BTC <- cbind(Hi(crypto$'BTC-USD'), Lo(crypto$'BTC-USD'), Cl(crypto$'BTC-USD'))[1:1186,]
ETH <- cbind(Hi(crypto$'ETH-USD'), Lo(crypto$'ETH-USD'), Cl(crypto$'ETH-USD'))[1:1186,]
LTC <- cbind(Hi(crypto$'LTC-USD'), Lo(crypto$'LTC-USD'), Cl(crypto$'LTC-USD'))[1:1186,]
USDT <- cbind(Hi(crypto$'USDT-USD'), Lo(crypto$'USDT-USD'), Cl(crypto$'USDT-USD'))[1:1186,]

create_ModelData <- function(cryptocurrency) {
ATRindicator<-ATR(cryptocurrency[,1:3],n=14) #calculate the indicator
ATR<-ATRindicator[,2] #grab just the ATR

LogReturns <- log(cryptocurrency[,3]) - lag(log(cryptocurrency[,3])) #calculate the logarithmic returns

ModelData<-data.frame(LogReturns,ATR) #create the data frame for our HMM model
ModelData<-ModelData[-c(1:14),] #remove the data where the indicators are being calculated
colnames(ModelData)<-c("LogReturns", "ATR") #name our columns

return(ModelData)
}

fit_hmm <- function (data_to_fit){
  aic <- vector(mode="numeric", length=5)
  bic <- vector(mode="numeric", length=5)
  logLik <- vector(mode="numeric", length=5)
  t <- rep(length(data_to_fit$LogReturns),5)
  n <- 2:6
  for (i in n) {
    hmm <- depmix(LogReturns~ 1, family = gaussian(), nstates = i, data=data.frame(data_to_fit));
    hmmfit <- fit(hmm,em.control=em.control(maxit=5000, tol=1e-10), verbose = FALSE)
    aic[i-1]<- AIC(hmmfit)
    bic[i-1]<- BIC(hmmfit)
    logLik[i-1]<- logLik(hmmfit)
    #fit <- cbind(aic,bic,logLik)
  }
  fit <- cbind(t,n,aic,bic,logLik)
  #print(paste("best model with number of hidden states", min_index+1))
  return(data.frame(fit))
}

validation_table<- function(observations){
T <- c(730,1095,length(observations[,1]))
N <- fit_hmm(observations[1:365,])
for (i in T){
  N2 <-fit_hmm(observations[1:i,])
  N<- rbind(N,N2)
}
return(data.frame(N))
}

```

```

set.seed(124)
BTC_Model <- create_ModelData(BTC)
tableBTC <- validation_table(BTC_Model)
print(round(tableBTC,2))

ETH_Model <- create_ModelData(ETH)
tableETH <- validation_table(ETH_Model)
print(round(tableETH,2))

LTC_Model <- create_ModelData(LTC)
tableLTC <- validation_table(LTC_Model)
print(round(tableLTC,2))

USDT_Model <- create_ModelData(USDT)
tableUSDT <- validation_table(USDT_Model)
print(round(tableUSDT,2))

par(mfrow=c(4,4))
plot(tableBTC[which(tableBTC$t==365),]$n,tableBTC[which(tableBTC$t==365),]$aic,
ylim=c(min(tableBTC[which(tableBTC$t==365),]$aic,tableBTC[which(tableBTC$t==365),]$bic),max(tableBTC[which(tableBTC$t==365),]$aic,tableBTC[which(tableBTC$t==365),]$bic)),
ylab="", ty="b", xlab="N",
main=paste("BTC", "t=365"))
lines(tableBTC[which(tableBTC$t==365),]$n,tableBTC[which(tableBTC$t==365),]$bic, ty="b", col="red")
legend("topleft",legend=c("AIC", "BIC"), col=c("black", "red"), lty=4, cex=0.8, bty='n')
plot(tableBTC[which(tableBTC$t==730),]$n,tableBTC[which(tableBTC$t==730),]$aic,
ylim=c(min(tableBTC[which(tableBTC$t==730),]$aic,tableBTC[which(tableBTC$t==730),]$bic),max(tableBTC[which(tableBTC$t==730),]$aic,tableBTC[which(tableBTC$t==730),]$bic)),
ylab="", ty="b", xlab="N",
main=paste("BTC", "t=730"))
lines(tableBTC[which(tableBTC$t==730),]$n,tableBTC[which(tableBTC$t==730),]$bic, ty="b", col="red")
legend("topleft",legend=c("AIC", "BIC"), col=c("black", "red"), lty=4, cex=0.8, bty='n')
plot(tableBTC[which(tableBTC$t==1095),]$n,tableBTC[which(tableBTC$t==1095),]$aic,
ylim=c(min(tableBTC[which(tableBTC$t==1095),]$aic,tableBTC[which(tableBTC$t==1095),]$bic),max(tableBTC[which(tableBTC$t==1095),]$aic,tableBTC[which(tableBTC$t==1095),]$bic)),
ty="b", ylab="", xlab="N",
main=paste("BTC", "t=1095"))
lines(tableBTC[which(tableBTC$t==1095),]$n,tableBTC[which(tableBTC$t==1095),]$bic, ty="b", col="red")
legend("topleft",legend=c("AIC", "BIC"), col=c("black", "red"), lty=4, cex=0.8, bty='n')
plot(tableBTC[which(tableBTC$t==1172),]$n,tableBTC[which(tableBTC$t==1172),]$aic,
ylim=c(min(tableBTC[which(tableBTC$t==1172),]$aic,tableBTC[which(tableBTC$t==1172),]$bic),max(tableBTC[which(tableBTC$t==1172),]$aic,tableBTC[which(tableBTC$t==1172),]$bic)),
ylab="", xlab="N",
main=paste("BTC", "t=1172"))
lines(tableBTC[which(tableBTC$t==1172),]$n,tableBTC[which(tableBTC$t==1172),]$bic, ty="b", col="red")
legend("topleft",legend=c("AIC", "BIC"), col=c("black", "red"), lty=4, cex=0.8, bty='n')

plot(tableETH[which(tableETH$t==365),]$n,tableETH[which(tableETH$t==365),]$aic,
ylim=c(min(tableETH[which(tableETH$t==365),]$aic,tableETH[which(tableETH$t==365),]$bic),max(tableETH[which(tableETH$t==365),]$aic,tableETH[which(tableETH$t==365),]$bic)),
ylab="", ty="b", xlab="N",
main=paste("ETH", "t=365"))
lines(tableETH[which(tableETH$t==365),]$n,tableETH[which(tableETH$t==365),]$bic, ty="b", col="red")
legend("topleft",legend=c("AIC", "BIC"), col=c("black", "red"), lty=4, cex=0.8, bty='n')
plot(tableETH[which(tableETH$t==730),]$n,tableETH[which(tableETH$t==730),]$aic,
ylim=c(min(tableETH[which(tableETH$t==730),]$aic,tableETH[which(tableETH$t==730),]$bic),max(tableETH[which(tableETH$t==730),]$aic,tableETH[which(tableETH$t==730),]$bic)),

```

```

bleETH$t==730),]$aic,tableETH[which(tableETH$t==730),]$bic)),          ylab="",          ty="b",          xlab="N",
main=paste("ETH", "t=730"))
lines(tableETH[which(tableETH$t==730),]$n,tableETH[which(tableETH$t==730),]$bic, ty="b", col="red")
legend("topleft",legend=c("AIC", "BIC"), col=c("black", "red"), lty=4, cex=0.8, bty='n')
plot(tableETH[which(tableETH$t==1095),]$n,tableETH[which(tableETH$t==1095),]$aic,
ylim=c(min(tableETH[which(tableETH$t==1095),]$aic,tableETH[which(tableETH$t==1095),]$bic),max(tableETH[which(
tableETH$t==1095),]$aic,tableETH[which(tableETH$t==1095),]$bic)),          ty="b",          ylab="",          xlab="N",
main=paste("ETH", "t=1095"))
lines(tableETH[which(tableETH$t==1095),]$n,tableETH[which(tableETH$t==1095),]$bic, ty="b", col="red")
legend("topleft",legend=c("AIC", "BIC"), col=c("black", "red"), lty=4, cex=0.8, bty='n')
plot(tableETH[which(tableETH$t==1172),]$n,tableETH[which(tableETH$t==1172),]$aic,          ty="b",
ylim=c(min(tableETH[which(tableETH$t==1172),]$aic,tableETH[which(tableETH$t==1172),]$bic),max(tableETH[which(
tableETH$t==1172),]$aic,tableETH[which(tableETH$t==1172),]$bic)),          ylab="",          xlab="N",
main=paste("ETH", "t=1172"))
lines(tableETH[which(tableETH$t==1172),]$n,tableETH[which(tableETH$t==1172),]$bic, ty="b", col="red")
legend("topleft",legend=c("AIC", "BIC"), col=c("black", "red"), lty=4, cex=0.8, bty='n')

plot(tableLTC[which(tableLTC$t==365),]$n,tableLTC[which(tableLTC$t==365),]$aic,
ylim=c(min(tableLTC[which(tableLTC$t==365),]$aic,tableLTC[which(tableLTC$t==365),]$bic),max(tableLTC[which(ta
bleLTC$t==365),]$aic,tableLTC[which(tableLTC$t==365),]$bic)),          ylab="",          ty="b",          xlab="N",
main=paste("LTC", "t=365"))
lines(tableLTC[which(tableLTC$t==365),]$n,tableLTC[which(tableLTC$t==365),]$bic, ty="b", col="red")
legend("topleft",legend=c("AIC", "BIC"), col=c("black", "red"), lty=4, cex=0.8, bty='n')
plot(tableLTC[which(tableLTC$t==730),]$n,tableLTC[which(tableLTC$t==730),]$aic,
ylim=c(min(tableLTC[which(tableLTC$t==730),]$aic,tableLTC[which(tableLTC$t==730),]$bic),max(tableLTC[which(ta
bleLTC$t==730),]$aic,tableLTC[which(tableLTC$t==730),]$bic)),          ylab="",          ty="b",          xlab="N",
main=paste("LTC", "t=730"))
lines(tableLTC[which(tableLTC$t==730),]$n,tableLTC[which(tableLTC$t==730),]$bic, ty="b", col="red")
legend("topleft",legend=c("AIC", "BIC"), col=c("black", "red"), lty=4, cex=0.8, bty='n')
plot(tableLTC[which(tableLTC$t==1095),]$n,tableLTC[which(tableLTC$t==1095),]$aic,
ylim=c(min(tableLTC[which(tableLTC$t==1095),]$aic,tableLTC[which(tableLTC$t==1095),]$bic),max(tableLTC[which(
tableLTC$t==1095),]$aic,tableLTC[which(tableLTC$t==1095),]$bic)),          ty="b",          ylab="",          xlab="N",
main=paste("LTC", "t=1095"))
lines(tableLTC[which(tableLTC$t==1095),]$n,tableLTC[which(tableLTC$t==1095),]$bic, ty="b", col="red")
legend("topleft",legend=c("AIC", "BIC"), col=c("black", "red"), lty=4, cex=0.8, bty='n')
plot(tableLTC[which(tableLTC$t==1172),]$n,tableLTC[which(tableLTC$t==1172),]$aic,          ty="b",
ylim=c(min(tableLTC[which(tableLTC$t==1172),]$aic,tableLTC[which(tableLTC$t==1172),]$bic),max(tableLTC[which(
tableLTC$t==1172),]$aic,tableLTC[which(tableLTC$t==1172),]$bic)),          ylab="",          xlab="N",
main=paste("LTC", "t=1172"))
lines(tableLTC[which(tableLTC$t==1172),]$n,tableLTC[which(tableLTC$t==1172),]$bic, ty="b", col="red")
legend("topleft",legend=c("AIC", "BIC"), col=c("black", "red"), lty=4, cex=0.8, bty='n')

plot(tableUSDT[which(tableUSDT$t==365),]$n,tableUSDT[which(tableUSDT$t==365),]$aic,
ylim=c(min(tableUSDT[which(tableUSDT$t==365),]$aic,tableUSDT[which(tableUSDT$t==365),]$bic),max(tableUSDT[whi
ch(tableUSDT$t==365),]$aic,tableUSDT[which(tableUSDT$t==365),]$bic)),          ylab="",          ty="b",          xlab="N",
main=paste("USDT", "t=365"))
lines(tableUSDT[which(tableUSDT$t==365),]$n,tableUSDT[which(tableUSDT$t==365),]$bic, ty="b", col="red")
legend("topright",legend=c("AIC", "BIC"), col=c("black", "red"), lty=4, cex=0.8, bty='n')
plot(tableUSDT[which(tableUSDT$t==730),]$n,tableUSDT[which(tableUSDT$t==730),]$aic,
ylim=c(min(tableUSDT[which(tableUSDT$t==730),]$aic,tableUSDT[which(tableUSDT$t==730),]$bic),max(tableUSDT[whi
ch(tableUSDT$t==730),]$aic,tableUSDT[which(tableUSDT$t==730),]$bic)),          ylab="",          ty="b",          xlab="N",
main=paste("USDT", "t=730"))

```

```

lines(tableUSDT[which(tableUSDT$t==730),]$n,tableUSDT[which(tableUSDT$t==730),]$bic, ty="b", col="red")
legend("topright",legend=c("AIC", "BIC"), col=c("black", "red"), lty=4, cex=0.8, bty='n')
plot(tableUSDT[which(tableUSDT$t==1095),]$n,tableUSDT[which(tableUSDT$t==1095),]$aic,
ylim=c(min(tableUSDT[which(tableUSDT$t==1095),]$aic,tableUSDT[which(tableUSDT$t==1095),]$bic),max(tableUSDT[w
hich(tableUSDT$t==1095),]$aic,tableUSDT[which(tableUSDT$t==1095),]$bic)),ty="b",      ylab="",      xlab="N",
main=paste("USDT", "t=1095"))
lines(tableUSDT[which(tableUSDT$t==1095),]$n,tableUSDT[which(tableUSDT$t==1095),]$bic, ty="b", col="red")
legend("topright",legend=c("AIC", "BIC"), col=c("black", "red"), lty=4, cex=0.8, bty='n')
plot(tableUSDT[which(tableUSDT$t==1172),]$n,tableUSDT[which(tableUSDT$t==1172),]$aic,      ty="b",
ylim=c(min(tableUSDT[which(tableUSDT$t==1172),]$aic,tableUSDT[which(tableUSDT$t==1172),]$bic),max(tableUSDT[w
hich(tableUSDT$t==1172),]$aic,tableUSDT[which(tableUSDT$t==1172),]$bic)),      ylab="",      xlab="N",
main=paste("USDT", "t=1172"))
lines(tableUSDT[which(tableUSDT$t==1172),]$n,tableUSDT[which(tableUSDT$t==1172),]$bic, ty="b", col="red")
legend("topright",legend=c("AIC", "BIC"), col=c("black", "red"), lty=4, cex=0.8, bty='n')

best<-NULL
min_index<-which.min(tableBTC$aic)
best <- cbind(Currency="BTC",round(tableBTC[min_index,],2))
min_index<-which.min(tableETH$aic)
best <- rbind(best,cbind(Currency="ETH", round(tableETH[min_index,],2)))
min_index<-which.min(tableLTC$aic)
best <- rbind(best,cbind(Currency="LTC", round(tableLTC[min_index,],2)))
min_index<-which.min(tableUSDT$aic)
best <- rbind(best,cbind(Currency="USDT", round(tableUSDT[min_index,],2)))
best

bestBIC<-NULL
min_index<-which.min(tableBTC$bic)
bestBIC <- cbind(Currency="BTC",round(tableBTC[min_index,],2))
min_index<-which.min(tableETH$bic)
bestBIC <- rbind(bestBIC,cbind(Currency="ETH", round(tableETH[min_index,],2)))
min_index<-which.min(tableLTC$bic)
bestBIC<- rbind(bestBIC,cbind(Currency="LTC", round(tableLTC[min_index,],2)))
min_index<-which.min(tableUSDT$bic)
bestBIC<- rbind(bestBIC,cbind(Currency="USDT", round(tableUSDT[min_index,],2)))
bestBIC

find_t <- function (data_to_fit,n){

  aic <- vector(mode="numeric")
  bic <- vector(mode="numeric")
  logLik <- vector(mode="numeric")
  T<- vector(mode="numeric")
  T[1] <- 180
  e<-9999
  hmm <- depmix(LogReturns ~ 1, family = gaussian(), nstates = n, data=data.frame(data_to_fit[1:T[1],]));
  hmmfit <- fit(hmm, verbose = FALSE)
  aic[1]<- AIC(hmmfit)
  bic[1]<- BIC(hmmfit)
  logLik[1]<- logLik(hmmfit)
  i <- 1
  while (T[i] <= length(data_to_fit[,1])) {

```

```

    i <- i+1
    T[i] <- T[i-1]+7
    hmm <- depmix(LogReturns ~ 1, family = gaussian(), nstates = n, data=data.frame(data_to_fit[1:(T[i]),]));
    hmmfit <- fit(hmm, verbose = FALSE)
    aic[i]<- AIC(hmmfit)
    bic[i]<- BIC(hmmfit)
    logLik[i]<- logLik(hmmfit)
    e <- (logLik[i] - logLik[i-1])/logLik[i-1]
  }
  fit <- cbind(T,aic,bic,logLik)
  return(data.frame(fit))
}

tBTC <- find_t(BTC_Model,3)
tETH <- find_t(ETH_Model,2)
tLTC <- find_t(LTC_Model,3)
tUSDT <- find_t(USDT_Model,3)

par(mfrow=c(2,2), mar = c(4, 4, 2, 2))
plot(tBTC$T,tBTC$aic, type="l", ylab="", xlab="t", main="BTC")
lines(tBTC$T,tBTC$bic, type="l", col=2)
legend("topright",legend=c("AIC", "BIC"), col=c("black", "red"), lty=1, cex=0.8, bty='n')
plot(tETH$T,tETH$aic, type="l", ylab="", xlab="t", main="ETC")
lines(tETH$T,tETH$bic, type="l", col=2)
legend("topright",legend=c("AIC", "BIC"), col=c("black", "red"), lty=1, cex=0.8, bty='n')
plot(tLTC$T,tLTC$aic, type="l", ylab="", xlab="t", main="LTC")
lines(tLTC$T,tLTC$bic, type="l", col=2)
legend("topright",legend=c("AIC", "BIC"), col=c("black", "red"), lty=1, cex=0.8, bty='n')
plot(tUSDT$T,tUSDT$aic, type="l", ylab="", xlab="t", main="USDT")
legend("topright",legend=c("AIC", "BIC"), col=c("black", "red"), lty=1, cex=0.8, bty='n')
lines(tUSDT$T,tUSDT$bic, type="l", col=2)
legend("topright",legend=c("AIC", "BIC"), col=c("black", "red"), lty=1, cex=0.8, bty='n')

regime_switcing <- function (model, n) {

  set.seed(124)
  HMM<-depmix(LogReturns~1,data=model[1:1172,],nstates=n,family=gaussian())
  HMMfit<-fit(HMM, verbose = FALSE) #fit our model to the data set

  print(HMMfit) #we can compare the log Likelihood as well as the AIC and BIC values to help choose our model
  summary(HMMfit)
  HMMpost<-posterior(HMMfit) #find the posterior odds for each state over our data set
  head(HMMpost) #we can see that we now have the probability for each state for everyday as well as the highest
  probability class.

  par(mfcol = c(4, 1), mar = numeric(4), oma = c(4, 4, .5, .5),
      mgp = c(0, .6, 0))
  plot(model$ATR, type="l", ylab = "ATR", xlab="", axes = FALSE)
  axis(2L)
  box()
  plot(model$LogReturns, type='l', lwd=1, xlab='', ylab='Gražos')
  axis(2L)

```

```

box()
matplot(HMMpost[,-1], type='l', ylab='Tikimybės', axes = FALSE)
axis(2L)
box()
plot(HMMpost$state, t="1", ylab = "Būsenos", xlab="T", axes = FALSE)
axis(2L)
axis(1L)
box()
mtext("T", side = 1, outer = TRUE, line = 2.2)
mtext("Būsenos", side = 2, outer = TRUE, line = 2.2)
mtext("Tikimybės", side = 2, outer = TRUE, line = 2.2)
mtext("Grąžos", side = 2, outer = TRUE, line = 2.2)
mtext("ATR", side = 2, outer = TRUE, line = 2.2)

print(attr(HMMfit,"response"))
}
n <- 3
regime_switching(BTC_Model, n)
regime_switching(ETH_Model, n)
regime_switching(LTC_Model, n)
regime_switching(USDT_Model, n)

```

## **Price Direction Prediction.R**

```

library(quantmod)
library(HMM)
library(markovchain)
library(caret)
library(dplyr)
set.seed(124)

crypto <- new.env()
portfolio = c("BTC-USD", "ETH-USD", "LTC-USD", "USDT-USD")
getSymbols(portfolio, env=crypto, src="yahoo", from="2017-01-01")

BTC <- Cl(crypto$'BTC-USD')[1:1095]
ETH <- Cl(crypto$'ETH-USD')[1:1095]
LTC <- Cl(crypto$'LTC-USD')[1:1095]

#-----
cuts <- c(-1, -0.05, -0.01, 0.01, 0.05, 1)
Symbols <- c("EK", "K", "Z", "A", "EA")

States <- c("Mazeja", "Dideja")
State_cuts <- c(-1, 0, 1)

#-----
new_ModelData <- function(cryptocurrency) {

LogReturns <- log(cryptocurrency[,1]) - lag(log(cryptocurrency[,1])) #calculate the logarithmic returns
Observations <- cut(LogReturns, cuts, labels=Symbols)
States <- cut(LogReturns, State_cuts, label=States)
}

```

```

ModelData<-data.frame(States,Observations) #create the data frame for our HMM model
ModelData<-ModelData[-c(1:1),]
colnames(ModelData)<-c("States","Observations") #name our columns

return (ModelData)
}

#-----

BTC_mod <- new_ModelData(BTC)
ETH_mod <- new_ModelData(ETH)
LTC_mod <- new_ModelData(LTC)

#-----

#Markovo grandiniu testavimas
library(spgs)
mt.BTC <- markov.test(BTC_mod$States)
mt.ETH <- markov.test(ETH_mod$States)
mt.LTC <- markov.test(LTC_mod$States)
mt.BTC
mt.ETH
mt.LTC

mcBTC<-markovchainFit(BTC_mod$States)
mcETH <- markovchainFit(ETH_mod$States)
mcLTC <- markovchainFit(LTC_mod$States)

transBTC <- markovchainFit(BTC_mod$States)$estimate
transETH <- markovchainFit(ETH_mod$States)$estimate
transLTC <- markovchainFit(LTC_mod$States)$estimate
is.irreducible(transBTC)
is.irreducible(transETH)
is.irreducible(transLTC)

verifyMarkovProperty(as.vector(BTC_mod$States))
assessOrder(as.vector(BTC_mod$States))
assessStationarity(BTC_mod$States, 10)

verifyMarkovProperty(as.vector(ETH_mod$States))
assessOrder(as.vector(ETH_mod$States))
assessStationarity(ETH_mod$States, 10)

verifyMarkovProperty(as.vector(LTC_mod$States))
assessOrder(as.vector(LTC_mod$States))
assessStationarity(LTC_mod$States, 10)

#-----

calculate_emission <- function (data, b, v) {
  em <- markovchainFit(data)$estimate
  new <- as.data.frame(em[b])
  return(round(new %>% select(v),5))
}

```

```

#-----

train_hmm <- function(train.data, States, Symbols, cuts) {

  tm <- markovchainFit(train.data$States)$estimate
  em <- calculate_emission(train.data, States, Symbols)
  hmm_model <- initWithMM(States, Symbols, transProbs=tm, emissionProbs=as.matrix(em))

  bw <- baumWelch(hmm_model,train.data$Observations, maxIterations=100, delta=1E-9)
  ep <- bw$hmm$emissionProbs
  tp <- bw$hmm$transProbs
  pi<-bw$hmm$startProbs

  hmm_model <- initWithMM(States, Symbols, startProbs=pi, transProbs=tp, emissionProbs=ep)

  vi <- viterbi(hmm_model, train.data$Observations)

  # show the result
  prog <- as.factor(vi)
  actual <- as.factor(train.data$States)
  result <- confusionMatrix(prog, actual)
  return(round(result$overall[1],4))
}

#-----

model_hmm <- function(train.data, States, Symbols, cuts) {

  tm <- markovchainFit(train.data$States)$estimate
  em <- calculate_emission(train.data, States, Symbols)
  hmm_model <- initWithMM(States, Symbols, transProbs=tm, emissionProbs=as.matrix(em))
  bw <- baumWelch(hmm_model,train.data$Observations, maxIterations=100, delta=1E-9)
  ep <- bw$hmm$emissionProbs
  tp <- bw$hmm$transProbs
  pi<-bw$hmm$startProbs

  hmm_model <- initWithMM(States, Symbols, startProbs=pi, transProbs=tp, emissionProbs=ep)

  vi <- viterbi(hmm_model, train.data$Observations)

  # show the result
  prog <- as.factor(vi)
  actual <- as.factor(train.data$States)
  result <- confusionMatrix(prog, actual)
  print(result)
  return(hmm_model)
}

#-----

test_hmm <- function (model, test.data, cuts, Symbols){

  stateViterbi <- viterbi(model, test.data$Observations)

```



```

    prog.t <- as.factor(stateViterbi)
    states.t <- as.factor(test.data$States)
    result <- confusionMatrix(prog.t, states.t)
    return(result)
}

#-----
#https://stackoverflow.com/questions/23891140/r-how-to-visualize-confusion-matrix-using-the-caret-package
draw_confusion_matrix <- function(cm) {

  total <- sum(cm$table)
  res <- as.numeric(cm$table)

  # Generate color gradients. Palettes come from RColorBrewer.
  greenPalette <- c("#F7FCF5", "#E5F5E0", "#C7E9C0", "#A1D99B", "#74C476", "#41AB5D", "#238B45", "#006D2C", "#00441B")
  redPalette <- c("#FFF5F0", "#FEE0D2", "#FCBBA1", "#FC9272", "#FB6A4A", "#EF3B2C", "#CB181D", "#A50F15", "#67000D")
  getColor <- function (greenOrRed = "green", amount = 0) {
    if (amount == 0)
      return("#FFFFFF")
    palette <- greenPalette
    if (greenOrRed == "red")
      palette <- redPalette
    colorRampPalette(palette)(100)[10 + ceiling(90 * amount / total)]
  }

  # set the basic layout
  layout(matrix(c(1,1,2)))
  par(mar=c(2,2,2,2))
  plot(c(100, 345), c(300, 450), type = "n", xlab="", ylab="", xaxt='n', yaxt='n')
  #title('Painiavos Matrica', cex.main=2)

  # create the matrix
  classes = colnames(cm$table)
  rect(150, 430, 240, 370, col=getColor("green", res[1]))
  text(195, 435, classes[1], cex=1.2)
  rect(250, 430, 340, 370, col=getColor("red", res[3]))
  text(295, 435, classes[2], cex=1.2)
  text(125, 370, 'Prognozuotos reikšmes', cex=1.3, srt=90, font=2)
  text(245, 450, 'Tikrosios reikšmes', cex=1.3, font=2)
  rect(150, 305, 240, 365, col=getColor("red", res[2]))
  rect(250, 305, 340, 365, col=getColor("green", res[4]))
  text(140, 400, classes[1], cex=1.2, srt=90)
  text(140, 335, classes[2], cex=1.2, srt=90)

  # add in the cm results
  text(195, 400, res[1], cex=1.6, font=2, col='black')
  text(195, 335, res[2], cex=1.6, font=2, col='black')
  text(295, 400, res[3], cex=1.6, font=2, col='black')
  text(295, 335, res[4], cex=1.6, font=2, col='black')

  # add in the specifics
  plot(c(100, 0), c(100, 0), type = "n", xlab="", ylab="", main = "", xaxt='n', yaxt='n')

```

```

text(10, 85, names(cm$byClass[1]), cex=1.2, font=2)
text(10, 70, round(as.numeric(cm$byClass[1]), 3), cex=1.2)
text(30, 85, names(cm$byClass[2]), cex=1.2, font=2)
text(30, 70, round(as.numeric(cm$byClass[2]), 3), cex=1.2)
text(50, 85, names(cm$byClass[5]), cex=1.2, font=2)
text(50, 70, round(as.numeric(cm$byClass[5]), 3), cex=1.2)
text(70, 85, names(cm$byClass[6]), cex=1.2, font=2)
text(70, 70, round(as.numeric(cm$byClass[6]), 3), cex=1.2)
text(90, 85, names(cm$byClass[7]), cex=1.2, font=2)
text(90, 70, round(as.numeric(cm$byClass[7]), 3), cex=1.2)

# add in the accuracy information
text(30, 35, names(cm$overall[1]), cex=1.5, font=2)
text(30, 20, round(as.numeric(cm$overall[1]), 3), cex=1.4)
text(70, 35, names(cm$overall[2]), cex=1.5, font=2)
text(70, 20, round(as.numeric(cm$overall[2]), 3), cex=1.4)
}

#-----

BTCt <- Cl(crypto$'BTC-USD')[1096:1186]
ETHt <- Cl(crypto$'ETH-USD')[1096:1186]
LTCt <- Cl(crypto$'LTC-USD')[1096:1186]

#-----

model <- model_hmm(BTC_mod, States, Symbols, cuts)
test.BTC <- new_ModelData(BTCt)
res.BTC <- test_hmm (model, test.BTC, cuts, Symbols)
tes.BTC <- test_hmm (model, BTC_mod, cuts, Symbols)

draw_confusion_matrix(tes.BTC)
draw_confusion_matrix(res.BTC)

#-----

model <- model_hmm(ETH_mod, States, Symbols, cuts)
test.ETH <- new_ModelData(ETHt)
res.ETH <- test_hmm (model, test.ETH, cuts, Symbols)
tes.ETH <- test_hmm (model, ETH_mod, cuts, Symbols)

draw_confusion_matrix(tes.ETH)
draw_confusion_matrix(res.ETH)

#-----

model <- model_hmm(LTC_mod, States, Symbols, cuts)
test.LTC <- new_ModelData(LTCt)
res.LTC <- test_hmm (model, test.LTC , cuts, Symbols)
tes.LTC <- test_hmm (model, LTC_mod, cuts, Symbols)

draw_confusion_matrix(tes.LTC)
draw_confusion_matrix(res.LTC)

```

```

#-----

sBTC <- round(as.numeric(tes.BTC$byClass), 3)
sETH <- round(as.numeric(tes.ETH$byClass), 3)
sLTC <- round(as.numeric(tes.LTC$byClass), 3)

sens <- cbind(sBTC, sETH, sLTC)
colnames(sens) <- c("BTC", "ETH", "LTC")
rownames(sens) <- names(tes.BTC$byClass)

aBTC <- round(as.numeric(tes.BTC$overall), 3)
aETH <- round(as.numeric(tes.ETH$overall), 3)
aLTC <- round(as.numeric(tes.LTC$overall), 3)

ac <- cbind(aBTC, aETH, aLTC)
colnames(ac) <- c("BTC", "ETH", "LTC")
rownames(ac) <- names(tes.BTC$overall)

all <- rbind(ac, sens)
all

sBTC <- round(as.numeric(res.BTC$byClass), 3)
sETH <- round(as.numeric(res.ETH$byClass), 3)
sLTC <- round(as.numeric(res.LTC$byClass), 3)

sens <- cbind(sBTC, sETH, sLTC)
colnames(sens) <- c("BTC", "ETH", "LTC")
rownames(sens) <- names(res.BTC$byClass)

aBTC <- round(as.numeric(res.BTC$overall), 3)
aETH <- round(as.numeric(res.ETH$overall), 3)
aLTC <- round(as.numeric(res.LTC$overall), 3)

ac <- cbind(aBTC, aETH, aLTC)
colnames(ac ) <- c("BTC", "ETH", "LTC")
rownames(ac ) <- names(res.BTC$overall)

all <- rbind(ac, sens)
all

-----

https://machinelearningmastery.com/machine-learning-in-r-step-by-step/
# Run algorithms using 10-fold cross validation
control <- trainControl(method="cv", number=10)
metric <- "Accuracy"

# Random Forest
set.seed(124)
fit.rf.BTC <- train(States~., data=BTC_mod, method="rf", metric=metric, trControl=control)
fit.rf.ETH <- train(States~., data=ETH_mod, method="rf", metric=metric, trControl=control)
fit.rf.LTC <- train(States~., data=LTC_mod, method="rf", metric=metric, trControl=control)

```

```

predictions.rf.BTC<- predict(fit.rf.BTC, test.BTC)
predictions.rf.ETH <- predict(fit.rf.ETH , test.ETH)
predictions.rf.LTC <- predict(fit.rf.LTC, test.LTC)

r.rf.BTC <- confusionMatrix(predictions.rf.BTC, test.BTC$States)
r.rf.ETH <- confusionMatrix(predictions.rf.ETH , test.ETH$States)
r.rf.LTC<- confusionMatrix(predictions.rf.LTC, test.LTC$States)

sens.rf.BTC <- round(as.numeric(r.rf.BTC$byClass), 3)
sens.rf.ETH <- round(as.numeric(r.rf.ETH$byClass), 3)
sens.rf.LTC <- round(as.numeric(r.rf.LTC$byClass), 3)

sens.rf<- cbind(sens.rf.BTC, sens.rf.ETH, sens.rf.LTC)
colnames(sens.rf) <- c("BTC", "ETH", "LTC")
rownames(sens.rf) <- names(res.BTC$byClass)

acc.rf.BTC <- round(as.numeric(r.rf.BTC$overall), 3)
acc.rf.ETH <- round(as.numeric(r.rf.ETH$overall), 3)
acc.rf.LTC <- round(as.numeric(r.rf.LTC$overall), 3)

ac.rf <- cbind(acc.rf.BTC, acc.rf.ETH, acc.rf.LTC)
colnames(ac.rf) <- c("BTC", "ETH", "LTC")
rownames(ac.rf) <- names(res.BTC$overall)

all.rf <- rbind(ac.rf, sens.rf)
all.rf

# Naive Baiesan
set.seed(124)
fit.nb.BTC <- train(States~., data=BTC_mod, method="nb", metric=metric, trControl=control)
fit.nb.ETH <- train(States~., data=ETH_mod, method="nb", metric=metric, trControl=control)
fit.nb.LTC <- train(States~., data=LTC_mod, method="nb", metric=metric, trControl=control)

predictions.nb.BTC <- predict(fit.nb.BTC , test.BTC)
predictions.nb.ETH<- predict(fit.nb.ETH, test.ETH)
predictions.nb.LTC<- predict(fit.nb.LTC, test.LTC)

r.nb.BTC <- confusionMatrix(predictions.nb.BTC, test.BTC$States)
r.nb.ETH <- confusionMatrix(predictions.nb.ETH , test.ETH $States)
r.nb.LTC <- confusionMatrix(predictions.nb.LTC, test.LTC$States)

sens.nb.BTC <- round(as.numeric(r.nb.BTC$byClass), 3)
sens.nb.ETH <- round(as.numeric(r.nb.ETH$byClass), 3)
sens.nb.LTC <- round(as.numeric(r.nb.LTC$byClass), 3)

sens.nb<- cbind(sens.nb.BTC, sens.nb.ETH, sens.nb.LTC)
colnames(sens.nb) <- c("BTC", "ETH", "LTC")
rownames(sens.nb) <- names(res.BTC$byClass)

```

```

acc.nb.BTC <- round(as.numeric(r.nb.BTC$overall), 3)
acc.nb.ETH <- round(as.numeric(r.nb.ETH$overall), 3)
acc.nb.LTC <- round(as.numeric(r.nb.LTC$overall), 3)

ac.nb <- cbind(acc.nb.BTC, acc.nb.ETH, acc.nb.LTC)
colnames(ac.nb) <- c("BTC", "ETH", "LTC")
rownames(ac.nb) <- names(res.BTC$overall)

all.nb <- rbind(ac.nb, sens.nb)
all.nb

# Neutral Network
# nnet
set.seed(124)
fit.nnet.BTC <- train(States~., data=BTC_mod, method="nnet", metric=metric, trControl=control)
fit.nnet.ETH <- train(States~., data=ETH_mod, method="nnet", metric=metric, trControl=control)
fit.nnet.LTC <- train(States~., data=LTC_mod, method="nnet", metric=metric, trControl=control)

predictions.nnet.BTC <- predict(fit.nnet.BTC , test.BTC)
predictions.nnet.ETH<- predict(fit.nnet.ETH, test.ETH)
predictions.nnet.LTC<- predict(fit.nnet.LTC, test.LTC)

r.nnet.BTC <- confusionMatrix(predictions.nnet.BTC, test.BTC$States)
r.nnet.ETH <- confusionMatrix(predictions.nnet.ETH , test.ETH $States)
r.nnet.LTC <- confusionMatrix(predictions.nnet.LTC, test.LTC$States)

sens.nnet.BTC <- round(as.numeric(r.nnet.BTC$byClass), 3)
sens.nnet.ETH <- round(as.numeric(r.nnet.ETH$byClass), 3)
sens.nnet.LTC <- round(as.numeric(r.nnet.LTC$byClass), 3)

sens.nnet<- cbind(sens.nnet.BTC, sens.nnet.ETH, sens.nnet.LTC)
colnames(sens.nnet) <- c("BTC", "ETH", "LTC")
rownames(sens.nnet) <- names(res.BTC$byClass)

acc.nnet.BTC <- round(as.numeric(r.nnet.BTC$overall), 3)
acc.nnet.ETH <- round(as.numeric(r.nnet.ETH$overall), 3)
acc.nnet.LTC <- round(as.numeric(r.nnet.LTC$overall), 3)

ac.nnet <- cbind(acc.nnet.BTC, acc.nnet.ETH, acc.nnet.LTC)
colnames(ac.nnet) <- c("BTC", "ETH", "LTC")
rownames(ac.nnet) <- names(res.BTC$overall)

all.nnet <- rbind(ac.nnet, sens.nnet)
all.nnet

#-----
T <- seq(30,1095, by=10)
Fitting <- NULL
FittingETH <- NULL
FittingLTC <- NULL

```

```

for (i in T) {
  b <- train_hmm(BTC_mod[1:i,], States, Symbols, cuts)

  e <- train_hmm(ETH_mod[1:i,], States, Symbols, cuts)

  l <- train_hmm(LTC_mod[1:i,], States, Symbols, cuts)

  Fitting <- rbind(Fitting, b)
  FittingETH <- rbind(FittingETH, e)
  FittingLTC <- rbind(FittingLTC, l)
}

par(mfrow=c(3,1))
plot(T, as.vector(Fitting[,1]), type = "l", ylab = "Accuracy", main= "BTC") #, ylim=c(0.4,1))

plot(T, as.vector(FittingETH[,1]), type = "l", ylab = "Accuracy", main= "ETH")

plot(T, as.vector(FittingLTC[,1]), type = "l", ylab = "Accuracy", main= "LTC")

Acc <- as.data.frame(cbind(Fitting, FittingETH, FittingLTC))
colnames(Acc) <- c("BTC", "ETH", "LTC")
rownames(Acc) <- T
Acc

```

## **Robustness.R**

```

library(quantmod)
library(HMM)
library(markovchain)

portfolio = c("BTC-USD","ETH-USD","LTC-USD")
crypto <- getSymbols(portfolio, src="yahoo", from="2017-01-01")
data <- xts()
for(i in seq_along(crypto)) {
  symbol <- crypto[i]
  data <- merge(data, get(symbol)[,paste(symbol, "Close", sep=".")])
}
data <- data[1:1186,]
head(data)
tail(data)
train.data <- data[1:1095,] #nuo 2017-01-01 iki 2020-03-31
test.data <- data[1096:1186,] #nuo 2020-04-01 iki 2020-04-14

week.btc<-sapply(split(train.data[,1], (seq_along(train.data[,1]) - 1) %% 7), mean)
days5.btc<-sapply(split(train.data[,1], (seq_along(train.data[,1]) - 1) %% 5), mean)
days3.btc<-sapply(split(train.data[,1], (seq_along(train.data[,1]) - 1) %% 3), mean)

week.eth<-sapply(split(train.data[,2], (seq_along(train.data[,2]) - 1) %% 7), mean)
days5.eth<-sapply(split(train.data[,2], (seq_along(train.data[,2]) - 1) %% 5), mean)
days3.eth<-sapply(split(train.data[,2], (seq_along(train.data[,2]) - 1) %% 3), mean)

```

```

week.ltc<-sapply(split(train.data[,2], (seq_along(train.data[,2]) - 1) %% 7), mean)
days5.ltc<-sapply(split(train.data[,2], (seq_along(train.data[,2]) - 1) %% 5), mean)
days3.ltc<-sapply(split(train.data[,2], (seq_along(train.data[,2]) - 1) %% 3), mean)

train.3d <- cbind(days3.btc,days3.eth,days3.ltc)
train.5d <- cbind(days5.btc,days5.eth,days5.ltc)
train.w <- cbind(week.btc,week.eth,week.ltc)

t.week.btc<-sapply(split(test.data[,1], (seq_along(test.data[,1]) - 1) %% 7), mean)
t.days5.btc<-sapply(split(test.data[,1], (seq_along(test.data[,1]) - 1) %% 5), mean)
t.days3.btc<-sapply(split(test.data[,1], (seq_along(test.data[,1]) - 1) %% 3), mean)

t.week.eth<-sapply(split(test.data[,2], (seq_along(test.data[,2]) - 1) %% 7), mean)
t.days5.eth<-sapply(split(test.data[,2], (seq_along(test.data[,2]) - 1) %% 5), mean)
t.days3.eth<-sapply(split(test.data[,2], (seq_along(test.data[,2]) - 1) %% 3), mean)

t.week.ltc<-sapply(split(test.data[,2], (seq_along(test.data[,2]) - 1) %% 7), mean)
t.days5.ltc<-sapply(split(test.data[,2], (seq_along(test.data[,2]) - 1) %% 5), mean)
t.days3.ltc<-sapply(split(test.data[,2], (seq_along(test.data[,2]) - 1) %% 3), mean)

test.3d <- cbind(t.days3.btc, t.days3.eth, t.days3.ltc)
test.5d <- cbind(t.days5.btc, t.days5.eth, t.days5.ltc)
test.w <- cbind(t.week.btc, t.week.eth, t.week.ltc)

#Funkcija apskaiciuojanti logaritmine grafa
cal_log_returns <- function(d, d_set_name){
  r <- matrix(, nrow = dim(d)[1]-1, ncol = dim(d)[2])
  colnames(r) <- d_set_name
  for (j in 1:dim(d)[2]){
    for (i in 1:(dim(d)[1]-1)){
      r[i,j] <- log(as.numeric(d[i+1,j])/as.numeric(d[i,j]),exp(1))
    }
  }
  return(r)
}

currencias <- c("BTC", "ETH", "LTC")
returns3d <- cal_log_returns(train.3d, portfolio)
colnames(returns3d) <- currencias
returns5d <- cal_log_returns(train.5d, portfolio)
colnames(returns5d) <- currencias
returnsweek <- cal_log_returns(train.w, portfolio)
colnames(returnsweek) <- currencias

t.returns3d <- cal_log_returns(test.3d, portfolio)
colnames(t.returns3d) <- currencias
t.returns5d <- cal_log_returns(test.5d, portfolio)
colnames(t.returns5d) <- currencias
t.returnsweek <- cal_log_returns(test.w, portfolio)
colnames(t.returnsweek) <- currencias

```

```

cuts <- c(-1, -0.05, -0.01, 0.01, 0.05, 1)
Symbols <- c("EK", "K", "Z", "A", "EA")

States <- c("Mazeja", "Dideja")
State_cuts <- c(-1, 0, 1)

new_ModelData <- function(cryptocurrency) {

Observations <- cut(cryptocurrency, cuts, labels=Symbols)
States <- cut(cryptocurrency, State_cuts ,label=States)

ModelData<-data.frame(States,Observations) #create the data frame for our HMM model
ModelData<-ModelData[-c(1:1),]
colnames(ModelData)<-c("States","Observations") #name our columns

return (ModelData)
}

#-----
BTC3d.model <- new_ModelData (returns3d[,1])
ETH3d.model <- new_ModelData (returns3d[,2])
LTC3d.model <- new_ModelData (returns3d[,3])

BTC5d.model <- new_ModelData (returns5d[,1])
ETH5d.model <- new_ModelData (returns5d[,2])
LTC5d.model <- new_ModelData (returns5d[,3])

BTCw.model <- new_ModelData (returnsweek[,1])
ETHw.model <- new_ModelData (returnsweek[,2])
LTCw.model <- new_ModelData (returnsweek[,3])

t.BTC3d.model <- new_ModelData (t.returns3d[,1])
t.ETH3d.model <- new_ModelData (t.returns3d[,2])
t.LTC3d.model <- new_ModelData (t.returns3d[,3])

t.BTC5d.model <- new_ModelData (t.returns5d[,1])
t.ETH5d.model <- new_ModelData (t.returns5d[,2])
t.LTC5d.model <- new_ModelData (t.returns5d[,3])

t.BTCw.model <- new_ModelData (t.returnsweek[,1])
t.ETHw.model <- new_ModelData (t.returnsweek[,2])
t.LTCw.model <- new_ModelData (t.returnsweek[,3])

#-----
#Markovo grandiniu testavimas

MarkovTest <- function(BTC_mod,ETH_mod,LTC_mod){
library(spgs)
mt.BTC <- markov.test(BTC_mod$States)
mt.ETH <- markov.test(ETH_mod$States)

```



```

mt.LTC <- markov.test(LTC_mod$States)

transBTC <- markovchainFit(BTC_mod$States)$estimate
transETH <- markovchainFit(ETH_mod$States)$estimate
transLTC <- markovchainFit(LTC_mod$States)$estimate

print("BTC")
print("-----")
print(mt.BTC)
print("-----")
print(is.irreducible(transBTC))
print("-----")
verifyMarkovProperty(as.vector(BTC_mod$States))
print("-----")
assessOrder(as.vector(BTC_mod$States))
print("-----")
assessStationarity(BTC_mod$States, 10)
print("-----")

print("ETH")
print("-----")
print(mt.ETH)
print("-----")
print(is.irreducible(transETH))
print("-----")
verifyMarkovProperty(as.vector(ETH_mod$States))
print("-----")
assessOrder(as.vector(ETH_mod$States))
print("-----")
assessStationarity(ETH_mod$States, 10)
print("-----")

print("LTC")
print("-----")
print(mt.LTC)
print("-----")
print(is.irreducible(transLTC))
print("-----")
verifyMarkovProperty(as.vector(LTC_mod$States))
print("-----")
assessOrder(as.vector(LTC_mod$States))
print("-----")
assessStationarity(LTC_mod$States, 10)
print("-----")
}

MarkovTest(BTC3d.model, ETH3d.model, LTC3d.model)
MarkovTest(BTC5d.model, ETH5d.model, LTC5d.model)
MarkovTest(BTCw.model, ETHw.model, LTCw.model)

#-----

```

```

accuracy_table <- function(c1,c2,c3){
  sBTC <- round(as.numeric(c1$byClass), 3)
  sETH <- round(as.numeric(c2$byClass), 3)
  sLTC <- round(as.numeric(c3$byClass), 3)

  sens <- cbind(sBTC, sETH, sLTC)
  colnames(sens) <- c("BTC", "ETH", "LTC")
  rownames(sens) <- names(c1$byClass)

  aBTC <- round(as.numeric(c1$overall), 3)
  aETH <- round(as.numeric(c2$overall), 3)
  aLTC <- round(as.numeric(c3$overall), 3)

  ac <- cbind(aBTC, aETH, aLTC)
  colnames(ac) <- c("BTC", "ETH", "LTC")
  rownames(ac) <- names(c1$overall)

  all <- rbind(ac, sens)
  all

  sBTC <- round(as.numeric(c1$byClass), 3)
  sETH <- round(as.numeric(c2$byClass), 3)
  sLTC <- round(as.numeric(c3$byClass), 3)

  sens <- cbind(sBTC, sETH, sLTC)
  colnames(sens) <- c("BTC", "ETH", "LTC")
  rownames(sens) <- names(c1$byClass)

  aBTC <- round(as.numeric(c1$overall), 3)
  aETH <- round(as.numeric(c2$overall), 3)
  aLTC <- round(as.numeric(c3$overall), 3)

  ac <- cbind(aBTC, aETH, aLTC)
  colnames(ac ) <- c("BTC", "ETH", "LTC")
  rownames(ac ) <- names(c1$overall)

  all <- rbind(ac, sens)
  return(all)
}

#-----

model3d <- model_hmm(BTC3d.model, States, Symbols, cuts)
train.res.BTC3d <- test_hmm (model3d, BTC3d.model, cuts, Symbols)
test.res.BTC3d <- test_hmm (model3d, t.BTC3d.model, cuts, Symbols)

model3d <- model_hmm(ETH3d.model, States, Symbols, cuts)
train.res.ETH3d <- test_hmm (model3d, ETH3d.model, cuts, Symbols)
test.res.ETH3d <- test_hmm (model3d, t.ETH3d.model, cuts, Symbols)

model3d <- model_hmm(LTC3d.model, States, Symbols, cuts)
train.res.LTC3d <- test_hmm (model3d, LTC3d.model, cuts, Symbols)

```

```

test.res.LTC3d <- test_hmm (model3d, t.LTC3d.model, cuts, Symbols)

draw_confusion_matrix(test.res.BTC3d)
draw_confusion_matrix(test.res.ETH3d)
draw_confusion_matrix(test.res.LTC3d)

accuracy_table(train.res.BTC3d,train.res.ETH3d,train.res.LTC3d)
accuracy_table(test.res.BTC3d,test.res.ETH3d,test.res.LTC3d)

model5d <- model_hmm(BTC5d.model, States, Symbols, cuts)
train.res.BTC5d <- test_hmm (model5d, BTC5d.model, cuts, Symbols)
test.res.BTC5d <- test_hmm (model5d, t.BTC5d.model, cuts, Symbols)

model5d <- model_hmm(ETH5d.model, States, Symbols, cuts)
train.res.ETH5d <- test_hmm (model5d, ETH5d.model, cuts, Symbols)
test.res.ETH5d <- test_hmm (model5d, t.ETH5d.model, cuts, Symbols)

model5d <- model_hmm(LTC5d.model, States, Symbols, cuts)
train.res.LTC5d <- test_hmm (model5d, LTC5d.model, cuts, Symbols)
test.res.LTC5d <- test_hmm (model5d, t.LTC5d.model, cuts, Symbols)

par(mfrow(3,1))
draw_confusion_matrix(test.res.BTC5d)
draw_confusion_matrix(test.res.ETH5d)
draw_confusion_matrix(test.res.LTC5d)

accuracy_table(train.res.BTC5d,train.res.ETH5d,train.res.LTC5d)
accuracy_table(test.res.BTC5d,test.res.ETH5d,test.res.LTC5d)

modelw <- model_hmm(BTCw.model, States, Symbols, cuts)
train.res.BTCw <- test_hmm (modelw, BTCw.model, cuts, Symbols)
test.res.BTCw <- test_hmm (modelw, t.BTCw.model, cuts, Symbols)

modelw <- model_hmm(ETHw.model, States, Symbols, cuts)
train.res.ETHw <- test_hmm (modelw, ETHw.model, cuts, Symbols)
test.res.ETHw <- test_hmm (modelw, t.ETHw.model, cuts, Symbols)

modelw <- model_hmm(LTCw.model, States, Symbols, cuts)
train.res.LTCw <- test_hmm (modelw, LTCw.model, cuts, Symbols)
test.res.LTCw <- test_hmm (modelw, t.LTCw.model, cuts, Symbols)

par(mfrow(3,1))
draw_confusion_matrix(test.res.BTCw)
draw_confusion_matrix(test.res.ETHw)
draw_confusion_matrix(test.res.LTCw)

accuracy_table(train.res.BTCw,train.res.ETHw,train.res.LTCw)
accuracy_table(test.res.BTCw,test.res.ETHw,test.res.LTCw)

other_methods <- function(BTC_mod, ETH_mod, LTC_mod, test.BTC,test.ETH,test.LTC) {
control <- trainControl(method="cv", number=10)
metric <- "Accuracy"

```

```

#Neural Network
# nnet
set.seed(124)
fit.nnet.BTC <- train(States~., data=BTC_mod, method="nnet", metric=metric, trControl=control)
fit.nnet.ETH <- train(States~., data=ETH_mod, method="nnet", metric=metric, trControl=control)
fit.nnet.LTC <- train(States~., data=LTC_mod, method="nnet", metric=metric, trControl=control)

predictions.nnet.BTC <- predict(fit.nnet.BTC , test.BTC)
predictions.nnet.ETH<- predict(fit.nnet.ETH, test.ETH)
predictions.nnet.LTC<- predict(fit.nnet.LTC, test.LTC)

r.nnet.BTC <- confusionMatrix(predictions.nnet.BTC, test.BTC$States)
r.nnet.ETH <- confusionMatrix(predictions.nnet.ETH , test.ETH $States)
r.nnet.LTC <- confusionMatrix(predictions.nnet.LTC, test.LTC$States)

sens.nnet.BTC <- round(as.numeric(r.nnet.BTC$byClass), 3)
sens.nnet.ETH <- round(as.numeric(r.nnet.ETH$byClass), 3)
sens.nnet.LTC <- round(as.numeric(r.nnet.LTC$byClass), 3)

sens.nnet<- cbind(sens.nnet.BTC, sens.nnet.ETH, sens.nnet.LTC)
colnames(sens.nnet) <- c("BTC", "ETH", "LTC")
rownames(sens.nnet) <- names(r.nnet.BTC$byClass)

acc.nnet.BTC <- round(as.numeric(r.nnet.BTC$overall), 3)
acc.nnet.ETH <- round(as.numeric(r.nnet.ETH$overall), 3)
acc.nnet.LTC <- round(as.numeric(r.nnet.LTC$overall), 3)

ac.nnet <- cbind(acc.nnet.BTC, acc.nnet.ETH, acc.nnet.LTC)
colnames(ac.nnet) <- c("BTC", "ETH", "LTC")
rownames(ac.nnet) <- names(r.nnet.BTC$overall)

all.nnet <- rbind(ac.nnet, sens.nnet)
print("-----")
print("Neural Network")
print(all.nnet)
print("-----")

# Random Forest
set.seed(124)
fit.rf.BTC <- train(States~., data=BTC_mod, method="rf", metric=metric, trControl=control)
fit.rf.ETH <- train(States~., data=ETH_mod, method="rf", metric=metric, trControl=control)
fit.rf.LTC <- train(States~., data=LTC_mod, method="rf", metric=metric, trControl=control)

predictions.rf.BTC<- predict(fit.rf.BTC, test.BTC)
predictions.rf.ETH <- predict(fit.rf.ETH , test.ETH)
predictions.rf.LTC <- predict(fit.rf.LTC, test.LTC)

r.rf.BTC <- confusionMatrix(predictions.rf.BTC, test.BTC$States)
r.rf.ETH <- confusionMatrix(predictions.rf.ETH , test.ETH$States)
r.rf.LTC<- confusionMatrix(predictions.rf.LTC, test.LTC$States)

```

```

sens.rf.BTC <- round(as.numeric(r.rf.BTC$byClass), 3)
sens.rf.ETH <- round(as.numeric(r.rf.ETH$byClass), 3)
sens.rf.LTC <- round(as.numeric(r.rf.LTC$byClass), 3)

sens.rf<- cbind(sens.rf.BTC, sens.rf.ETH, sens.rf.LTC)
colnames(sens.rf) <- c("BTC", "ETH", "LTC")
rownames(sens.rf) <- names(r.rf.BTC$byClass)

acc.rf.BTC <- round(as.numeric(r.rf.BTC$overall), 3)
acc.rf.ETH <- round(as.numeric(r.rf.ETH$overall), 3)
acc.rf.LTC <- round(as.numeric(r.rf.LTC$overall), 3)

ac.rf <- cbind(acc.rf.BTC, acc.rf.ETH, acc.rf.LTC)
colnames(ac.rf) <- c("BTC", "ETH", "LTC")
rownames(ac.rf) <- names(r.rf.BTC$overall)

all.rf <- rbind(ac.rf, sens.rf)
print("-----")
print("Random Forest")
print(all.rf)
print("-----")

# Naive Baiesan
set.seed(124)
fit.nb.BTC <- train(States~., data=BTC_mod, method="nb", metric=metric, trControl=control)
fit.nb.ETH <- train(States~., data=ETH_mod, method="nb", metric=metric, trControl=control)
fit.nb.LTC <- train(States~., data=LTC_mod, method="nb", metric=metric, trControl=control)

predictions.nb.BTC <- predict(fit.nb.BTC , test.BTC)
predictions.nb.ETH<- predict(fit.nb.ETH, test.ETH)
predictions.nb.LTC<- predict(fit.nb.LTC, test.LTC)

r.nb.BTC <- confusionMatrix(predictions.nb.BTC, test.BTC$States)
r.nb.ETH <- confusionMatrix(predictions.nb.ETH , test.ETH $States)
r.nb.LTC <- confusionMatrix(predictions.nb.LTC, test.LTC$States)

sens.nb.BTC <- round(as.numeric(r.nb.BTC$byClass), 3)
sens.nb.ETH <- round(as.numeric(r.nb.ETH$byClass), 3)
sens.nb.LTC <- round(as.numeric(r.nb.LTC$byClass), 3)

sens.nb<- cbind(sens.nb.BTC, sens.nb.ETH, sens.nb.LTC)
colnames(sens.nb) <- c("BTC", "ETH", "LTC")
rownames(sens.nb) <- names(r.rf.BTC$byClass)

acc.nb.BTC <- round(as.numeric(r.nb.BTC$overall), 3)
acc.nb.ETH <- round(as.numeric(r.nb.ETH$overall), 3)
acc.nb.LTC <- round(as.numeric(r.nb.LTC$overall), 3)

ac.nb <- cbind(acc.nb.BTC, acc.nb.ETH, acc.nb.LTC)
colnames(ac.nb) <- c("BTC", "ETH", "LTC")

```

```
rownames(ac.nb) <- names(r.rf.BTC$overall)

all.nb <- rbind(ac.nb, sens.nb)
print("-----")
print("Naive Bayesian")
print(all.nb)
print("-----")
}

other_methods(BTC3d.model,ETH3d.model,LTC3d.model,t.BTC3d.model,t.ETH3d.model, t.LTC3d.model)
other_methods(BTC5d.model,ETH5d.model,LTC5d.model,t.BTC5d.model,t.ETH5d.model, t.LTC5d.model)
other_methods(BTCw.model,ETHw.model,LTCw.model,t.BTCw.model,t.ETHw.model, t.LTCw.model)
```