Kauno technologijos universitetas

Matematikos ir gamtos mokslų fakultetas

Mašininio mokymosi pritaikymas blokų grandinių tyrimui ir sukčiavimo aptikimui

Baigiamasis magistro studijų projektas

Evaldas Žilinskas

Projekto autorius

prof. dr. Robertas Alzbutas

Vadovas

doc. dr. Alfreda Šapkauskienė

Vadovė

Vilnius, 2020

Kauno technologijos universitetas

Matematikos ir gamtos mokslų fakultetas

Mašininio mokymosi pritaikymas blokų grandinių tyrimui ir sukčiavimo aptikimui

Baigiamasis magistro studijų projektas

Didžiųjų verslo duomenų analitika (6213AX001)

Evaldas Žilinskas

Projekto autorius

prof. dr. Robertas Alzbutas

Vadovas

doc. dr. Alfreda Šapkauskienė

Vadovė

doc. dr. Audrius Kabašinskas

Recenzentas

prof. dr. Lina Dagilienė

Recenzentė

Vilnius, 2020

Kauno technologijos universitetas

Matematikos ir gamtos mokslų fakultetas

Evaldas Žilinskas

Mašininio mokymosi pritaikymas blokų grandinių tyrimui ir sukčiavimo aptikimui

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Evaldo Žilinsko, baigiamasis projektas tema „Mašininio mokymosi pritaikymas blokų grandinių tyrimui ir sukčiavimo aptikimui“ 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)

Žilinskas, Evaldas. Mašininio mokymosi pritaikymas blokų grandinių tyrimui ir sukčiavimo aptikimui. Magistro studijų baigiamasis projektas / vadovas prof. dr. Robertas Alzbutas ir vadovė doc. dr. Alfreda Šapkauskienė; Kauno technologijos universitetas, Matematikos ir gamtos mokslų fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Taikomoji matematika (Matematikos mokslai).

Reikšminiai žodžiai: blokų grandinė; bitkoinas; eteris; išskirčių nustatymas; sukčiavimo aptikimas; k-vidurkių metodas; izoliavimo miškas.

Vilnius, 2020. 61 p.

Santrauka

Susidomėjimas blokų grandinės technologija auga nuo šios idėjos atsiradimo 2008 metais. Tai palyginti nauja technologija, kuri gyvuoja dar tik apie 12 metų, tačiau sulaukia nemažai žiniasklaidos ir mokslininkų dėmesio. Pagrindinis domėjimosi objektas yra bitkoino kriptografinė valiuta, kuriai blokų grandinė pirmiausia ir buvo sukurta. Kriptografinių valiutų populiarumas pritraukia ir įvairių sukčių, kurie vykdo nekorektiškas veiklas ir stengiasi gauti finansinės naudos. Kol kas 2019 metai pagal sukčiavimo apimtis buvo patys didžiausi ir per šiuos metus padaryta žala yra vertinama 4,3 milijardais dolerių. Bitkoinui tenka didžiausia šių nuostolių suma, tačiau antra pagal kapitalizacijos dydį eterio kriptografinė valiuta taip pat sulaukia sukčių dėmesio.

Sukčiavimo aptikimas yra pirmas veiksmas norint sumažinti riziką ir apsisaugoti nuo galimų vagysčių ir apgavysčių. Šio darbo tikslas – pritaikant didžiųjų duomenų analitikos metodus sukurti mašininio mokymosi modelį, kuris įgalintų apdoroti didelius duomenų kiekius ir sėkmingai aptikti sukčiavimus bitkoino ir eterio blokų grandinėje. Visi bitkoino ir eterio sandoriai yra viešai prieinami. Panaudojant šiuos didžiuosius duomenis buvo išskirti požymiai (atliktų pavedimų skaičius, vidutinė atliktų pavedimų vertė ir pan.), kurie naudojami kuriant modelius. Sukčiavimo aptikimui buvo sukurti išskirčių nustatymo modeliai paremti k-vidurkių ir izoliavimo miško metodais. Dėl turimo didelio duomenų kiekio buvo kuriami atitinkamų modelių ansambliai.

Sukurti mašininio mokymosi modeliai leido identifikuoti kriptografinių valiutų adresus (bankinės sąskaitos atitikmuo kriptografinių valiutų blokų grandinėje), kurie yra susiję su sukčiavimo atvejais. Bitkoino kriptografinės valiutos blokų grandinėje vienas k-vidurkių modelis, k-vidurkių modelių ansamblis, izoliavimo miško modelių ansamblis aptiko panašų apgaulių kiekį (29–30). Eterio kriptografinės valiutos blokų grandinėje sukčiavimą geriausiai sekėsi aptikti k-vidurkių modelių ansambliui, kuris iš viso aptiko 65 apgaules. Rezultatams patikrinti buvo naudojami trys skirtingi apgaulių duomenų rinkiniai. Iš „BitcoinTalk“ duomenų rinkinio pavyko identifikuoti 15 iš 16 bitkoino adresų, susijusių su apgaulėmis. Tai yra labai geras rezultatas, nes panašiuose tyrimuose, iš šio duomenų rinkinio, daugiausiai pavykdavo aptikti tik 5 apgaulės atvejus. Iš Ponzi schemų duomenų rinkinio eterio blokų grandinėje pavyko identifikuoti 64 apgaules iš 102. Iš „CryptoScamDB“ duomenų rinkinio bitkoino blokų grandinėje dėl ten patenkančių mažesnio mąsto apgaulių identifikuoti pavyko 14 apgaulių iš 140. Taip pat atlikti tyrimai parodė, kad mašininio mokymosi modeliai sukurti naudojant bitkoino sandorių duomenis, gali būti sėkmingai panaudoti aptinkant sukčiavimo atvejus eterio blokų grandinėje. Tačiau, modeliai, kurti naudojant bitkoino sandorių duomenis, aptinka mažiau apgaulių eterio blokų grandinėje, nei modeliai sukurti naudojant eterio sandorių duomenis.

Žilinskas, Evaldas. Machine learning application for blockchain analysis and fraud detection. Master's Final Degree Project / supervisor prof. dr. Robertas Alzbutas and doc. dr. Alfreda Šapkauskienė; Faculty of Mathematics and Natural Sciences, Kaunas University of Technology.

Study field and area (study field group): Applied Mathematics (Matchematical Sciences).

Keywords: blockchain; bitcoin; ethereum; outlier identification; fraud detection; k-means; isolation forest.

Vilnius, 2020. 61.

Summary

Interest in blockchain technology has been growing since 2008 when this concept was created. It is a relatively new technology that has been around for only 12 years but has received much attention in the media and from academics. The main object of the media's focus is the bitcoin cryptocurrency, for which blockchain technology was first developed. The popularity of cryptocurrencies also attracts various scammers who engage in improper activities and seek financial gain. So far, the most significant damage has been done in 2019 and is estimated at $ 4.3 billion. As bitcoin is the most popular cryptocurrency, it bears the most considerable amount of damage caused by these thefts and frauds. Ethereum, the second-largest cryptocurrency by capitalization, is also receiving attention from scammers.

Fraud detection is the first step in reducing risk and preventing potential theft and fraud. This study aims to develop a machine learning model using big data analytics methods that would be able to process large amounts of data and successfully identify fraud within the bitcoin and ethereum blockchain. All bitcoin and ethereum transactions are publicly available. Using these big data, the features (number of transactions received, an average value of the received transaction, etc.) that were used to develop the models were extracted. The k-means and the isolation forest methods were applied to create fraud detection model. Due to the big amount of data available, ensembles of these methods were developed.

The developed machine learning models identified addresses that are associated with cases of fraud and scam. Looking at the overall results, one k-means model, an ensemble of k-means models, and an ensemble of isolation forest models found almost the same number of frauds in the bitcoin blockchain (29–30). In the ethereum blockchain, frauds were best detected by using an ensemble of k-means models, which caught a total of 65 scams. Three different data sets of fraud were used to verify the results. The developed models in the BitcoinTalk dataset identified 15 of the 16 bitcoin addresses associated with frauds. This is very good result, as a maximum 5 cases of fraud were detected in similar studies before. In the Ponzi schemes dataset were identified 64 scams in the ethereum blockchain out of 102. The developed models in the CryptoScamDB dataset identified 14 scams in the bitcoin blockchain of 140 because this dataset included smaller scams. Studies by other authors using Ponzi schemes and CryptoScamDB dataset use different methods (e.g. classification methods are used, results are calculated differently) and therefore the results are not comparable. This study has also shown that machine learning models developed using bitcoin transaction data can be successfully used to detect fraud in the ethereum blockchain. However, models developed using bitcoin transaction data detect fewer cases of scam in the ethereum blockchain than models developed using ethereum transaction data.

Turinys

Lentelių sąrašas

Paveikslų sąrašas

Santrumpų ir terminų sąrašas

Santrumpos:

DER – Distinguished Encoding Rules.

SegWit – Segregated Witness.

SHA-256 – Secure Hash Algorithm.

UTXO – Unspent transaction output.

HDD – Hard Disk Drive.

SSD – Solid State Drive.

CSV – Comma-separated values.

JSON – JavaScript Object Notation.

ERC20 – Ethereum Request for Comments 20.

Terminai:

Kriptografinės valiutos adresas – identifikatorius sudarytas iš raidžių ir skaitmenų, kuris yra naudojamas, nurodant gavėją ir mokėtoją, atliekant kriptografinės valiutos sandorį. Adresas kriptografinės valiutos blokų grandinėje yra kaip bankinės sąskaita.

Kriptografinės valiutos piniginė – tai programinė įranga sauganti privačius ir viešuosius raktus, taip pat sąveikaujanti su įvairiomis blokų grandinėmis, kad naudotojai galėtų siųsti ir gauti kriptografines valiutas bei stebėti savo balansą.

Privatus raktas – kriptografinis raktas, kuris gali būti naudojamas iššifruoti rakto savininkui su viešuoju raktu užšifruotus pranešimus, taip pat jį panaudojus gali būti sukurtas skaitmeninis parašas, kurio autentiškumas gali būti patikrintas turint atitinkamą viešąjį raktą.

Satošis – smulkiausioji bitkoino dalis, lygi vienai šimtamilijoninei bitkoino**.**

SegWit procesas – bitkoino blokų grandinės patobulinimas, kuris sumažino vietą reikalingą sandoriams saugoti bloke.

UTXO – elektroninių pinigų abstrakcija. Kiekvienas UTXO perteikia nuosavybės grandinę, kuri yra įgyvendinta kaip skaitmeninių parašų grandinė, kur savininkas pasirašo sandorį, perkeldamas savo UTXO nuosavybės teises į gavėjo viešąjį raktą.

Vėjus – smulkiausioji eterio dalis. 1 eteris = 1 000 000 000 000 000 000 vėjų (10^{18}).

Viešas raktas – kriptografinis raktas, kuris naudojamas šifruoti konkrečiam gavėjui skirtus pranešimus, kad užšifruotas pranešimas būtų iššifruojamas tik naudojant privatų raktą, kuris yra žinomas tik gavėjui.

Įvadas

Susidomėjimas blokų grandinės technologija auga nuo šios idėjos atsiradimo 2008 metais. Tai palyginti nauja technologija, kuri gyvuoja dar tik apie 12 metų, tačiau sulaukia nemažai tiek žiniasklaidos, tiek mokslininkų dėmesio. Pagrindinis žiniasklaidos traukos objektas yra bitkoino kriptografinė valiuta, kuriai blokų grandinės technologija pirmiausia ir buvo sukurta. Kol kas bitkoino populiarumo viršūne galima laikyti 2017 metų pabaigą, kai šios kriptografinės valiutos kaina pasiekė beveik 20 000 dolerių už vieną bitkoiną. Taip pat susidomėjimą šia kriptografine valiuta atspindi „Google“ paieškų rezultatai, kurie buvo gerokai išaugę 2017 metų pabaigoje, o po to grįžo į prieš tai buvusį lygį. Nors bitkoino kriptografinės valiutos kaina pastaruoju metu yra nukritusi lyginant su 2017 metais pasiektoms aukštumoms ir svyruoja apie 8 900 dolerių [1] (2020 gegužės 26 d. duomenys), tačiau bitkoino rinkos kapitalizacija išlieka didžiausia ir siekia beveik 164 milijardus dolerių [2] (2020 gegužės 26 d. duomenys). Antra pagal kapitalizaciją yra eterio kriptografinė valiuta ir jos kapitalizacija yra daugiau nei 22 milijardų dolerių [2] (2020 gegužės 26 d.). Nemenka kriptografinių valiutų kapitalizacija ir technologijos naujumas yra pagrindiniai traukos objektai, skatinantys domėjimąsi tiek kriptografinėmis valiutomis, tiek blokų grandinės technologijomis.

Blokų grandinė nėra vien tik apie bitkoiną ar kitas kriptografines valiutas. Šios technologijos pritaikymo ribos nuolat plečiasi ir surandama vis naujų sričių ir sprendimų, kur šią technologiją galima pritaikyti. Prie to labai daug prisideda ir mokslininkai, tyrinėjantys blokų grandines. „Science Direct“ duomenų bazėje atlikta paieška rodo, kad su blokų grandinėmis (paieškos frazė „block chain“ arba „blockchain“) susijusių mokslinių straipsnių kiekis per pastaruosius metus išaugo beveik trigubai (2018 metais buvo publikuota 320 mokslinių straipsnių, o 2019 – 926), o bitkoino – daugiau nei 2,5 karto (2018 metais buvo publikuoti 206 moksliniai straipsniai, o 2019 – 532). Yli-Huumo ir kt. [3] atliekamoje straipsnių apžvalgoje net 80 % straipsnių yra susiję su bitkoino kriptografine valiuta, o likę 20 % su kitomis blokų grandinės technologijomis. Autoriai apskaičiavo, kad net 34 % straipsnių nagrinėja blokų grandinės ir bitkoino saugumo klausimus. Ne veltui toks didelis dėmesys yra skiriamas blokų grandinės saugumui, nes nuo 2011 iki 2018 metų didžiausios kriptografinių valiutų vagystės ir apgavystės yra vertinamos 1,7 milijardo dolerių [4]. 2019 metai kol kas buvo patys didžiausi pagal sukčiavimo apimtis ir sukčių padaryta žala siekė 4,3 milijardo dolerių (išaugo tris kartus lyginant su 2018 metais) [5]. Taigi nors blokų grandinė gali būti pritaikoma įvairiose srityse, tačiau daugiausiai ne tik mokslininkų, bet ir sukčių, norinčių nelegaliu būdu pasipelnyti, dėmesio susilaukia kriptografinės valiutos.

Nusikalstamos veiklos kenkia tiek bitkoinų rinkai, tiek pačiai blokų grandinės reputacijai. Todėl norint suteikti daugiau pasitikėjimo, blokų grandinės technologiją reikia identifikuoti ir, jei įmanoma, užkirsti kelią sukčiavimui. Tai yra sudėtingas uždavinys, reikalaujantis daug resursų ir didžiųjų duomenų analizės metodų, nes blokų grandinėje susiduriama su dideliais sandorių kiekiais, taip pat aktualus ir informacijos apdorojimo greitis. Nors ši sritis ir susilaukia daug mokslininkų dėmesio, tačiau kol kas efektyvus ir geras metodas, leidžiantis aptikti sukčiavimus, nėra pasiūlytas. Dažniausiai susiduriama su problema, kad modeliai negali apdoroti visos turimos informacijos, kitais atvejai modelių veikimas nėra pakankamai tikslus. Taip pat keliamas klausimas, ar įmanomas toks modelis, kuris tiktų visoms blokų grandinių sistemoms. Norint, kad kriptografinių valiutų rinkos ir blokų grandinės žmonėms keltų daugiau pasitikėjimo ir pritrauktų daugiau šios technologijos naudotojų, reikalingi sprendimai, galintys užkirsti kelią sukčiavimui.

Šio darbo tikslas – pritaikant didžiųjų duomenų analitikos metodus sukurti mašininio mokymosi modelį, kuris įgalintų apdoroti didelius duomenų kiekius ir sėkmingai aptikti sukčiavimus bitkoino ir eterio blokų grandinėje.

Siekiant įgyvendinti tikslą **keliami šie uždaviniai**:

1. Išanalizuoti blokų grandinės tyrimo galimybes ir sukčiavimo atvejus, su kuriais susiduriama kriptografinių valiutų atveju.
2. Atrinkti metodus, kurie kitų autorių tyrimuose yra pritaikomi aptinkant sukčiavimus ir išskirti iššūkius kylančius su šiais metodais.
3. Pasiūlyti naują metodiką, kuri, pritaikant didžiųjų duomenų analitikos metodus, įgalintų aptikti sukčiavimus su kriptografinėmis valiutomis.
4. Sukurti modelius, leidžiančius aptikti sukčiavimo atvejus bitkoinų blokų grandinėje, ir palyginti modelių veikimą su kitų autorių rezultatais.
5. Patikrinti sukurtų modelių pritaikymą eterio kriptografinės valiutos blokų grandinėje, ir palyginti su eterio blokų grandinei sukurtais modeliais.

Mokslinio tyrimo metodai. Teorinei kriptografinės valiutos blokų grandinės veikimo bei sandaros apžvalgai darbe naudojami mokslinės literatūros analizės, sisteminimo, apibendrinimo ir palyginimo metodai. Atliekant tyrimą ir interpretuojant rezultatus buvo naudojami modeliavimo, duomenų analizės ir grafinio duomenų vaizdavimo metodai.

Darbo struktūrą sudaro trys dalys. Pirmoje dalyje pateikiama blokų grandinės panaudojimo, sandaros ir veikimo principų analizė. Nagrinėjama detali bitkoino ir eterio blokų grandinės sandara. Taip pat aprašomi blokų grandinėje galimi sukčiavimo atvejai bei analizuojami metodai ir modeliai leidžiantys aptikti šiuos atvejus. Antroje dalyje parenkami tyrimui tinkami duomenų rinkiniai, išskiriami požymiai, aprašomi metodai, kuriami modeliai ir rezultatų vertinimo principai. Trečiojoje dalyje yra lyginami sukurti sukčiavimo aptikimo modeliai, pateikiami tyrimų rezultatai. Pabaigoje pateikiamos išvados.

1. Blokų grandinės ir su ja susijusių sukčiavimo atvejų apžvalga

1.1. Blokų grandinės samprata

Blokų grandinė yra decentralizuota sandorių ir duomenų valdymo technologija, kuri pirmiausia buvo sukurta bitkoino kriptografinei valiutai. Tačiau blokų grandinės galimybės leidžia kur kas plačiau pritaikyti šią technologiją – patobulinant esamas technologijas ir diegiant programas, kurios anksčiau nebūdavo praktiškos. Blokų grandinė, kaip ir internetas, yra atvira ir globali infrastruktūra, kuri leidžia įmonėms ir fiziniams asmenims atlikti sandorius išvengiant tarpininkų. Atsisakant tarpininkų, yra sumažinami sandorio aptarnavimo kaštai ir sutrumpinamas jo apdorojimo laikas [6]. Toliau esančiuose poskyriuose bus detaliau pateikta informacija apie blokų grandinės pritaikymo galimybes, jos teikiamus privalumus ir veikimo principus.

1.1.1. Blokų grandinės pritaikymo galimybių raida

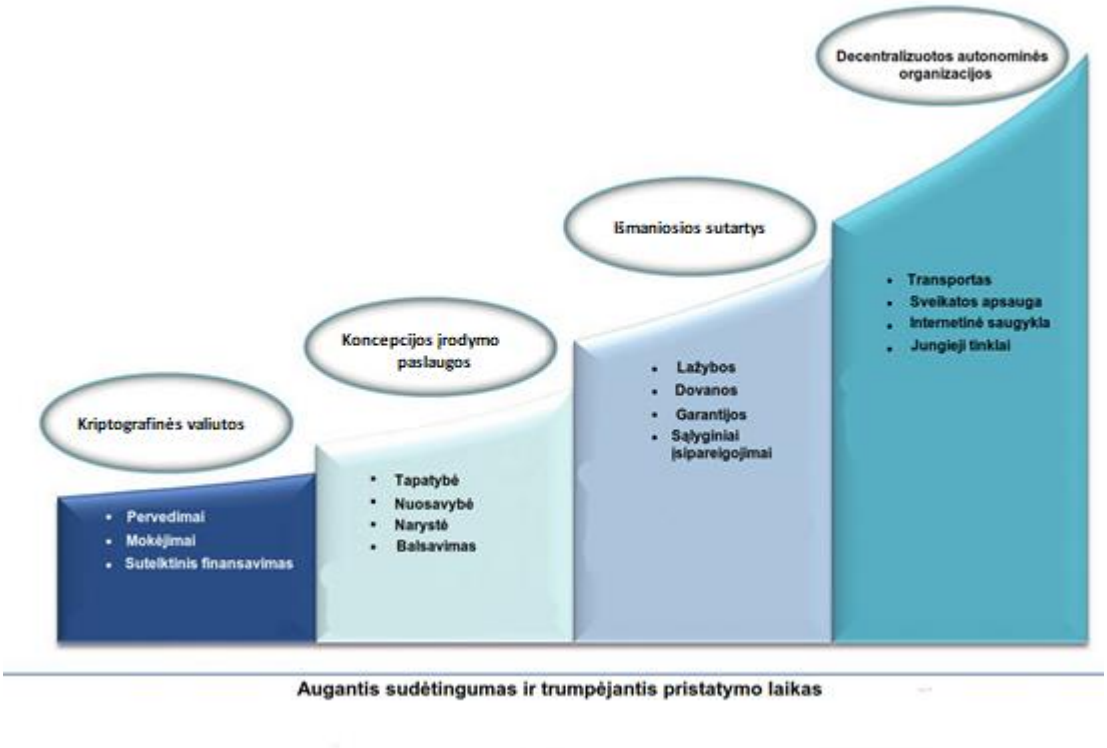
Nors blokų grandinės technologija atsirado tik 2008 metais, tačiau jau spėjo pereiti nemažai vystymosi etapų. Swan'as [7] išskiria tris pagrindines blokų grandinės revoliucijas:

- Blokų grandinė 1.0. Ji yra susijusi su valiutomis. Ji apima kriptografinių valiutų pervedimus, skaitmenines mokėjimo sistemas ir pan.
- Blokų grandinė 2.0. Ji yra susijusi su sutartimis. Ji apima ekonomines, rinkos ir finansines operacijos, kurios yra daugiau nei paprasti valiutų sandoriai: akcijų, obligacijų, ateities sandorių, paskolų, būsto paskolų ir kt. sutartys.
- Blokų grandinė 3.0. Tai yra blokų grandinės pritaikymas už valiutų, finansų ir rinkos ribų. Tai gali būti pritaikymai vyriausybės, sveikatos, mokslo, kultūros, meno ir kt. srityse.

Umeh'as [8] grupuodamas blokų grandinės pritaikymo atvejus (1 pav.) juos suskirsto labai panašiai kaip ir Swan'as [7] kalbėdama apie revoliucijos etapus:

- Kriptografinės valiutos. Ši grupė apima sprendimus naudojamus pavedimams ir mokėjimams. Šie patikimi sandoriai vyksta tarp nežinomų šalių, mažomis kainomis ir be jokių tarpininkų.
- Koncepcijos įrodymo paslaugos. Šiai grupei priskiriamos sistemos, kurios yra pagrįstos pagrindine blokų grandinės galimybę saugoti informaciją atominiame lygyje (identifikavimo, nuosavybės ar kitai informacijai). Naudojantis šia galimybe yra vystomi sudėtingesni sprendimai.
- Išmaniosios sutartys. Šioje grupėje esančios sistemos leidžia vykdyti sutartis be trečiųjų šalių įsikišimo.
- Decentralizuotos autonominės organizacijos. Šiai grupei priklauso sistemos turinčios turbūt svarbiausią blokų grandinės vaidmenį, t. y., pasitikėjimo mechanizmą tarp tarpusavyje priklausomų žmonių ir mašinų. Tokios sistemos veikia internete ir gali egzistuoti autonomiškai. Tačiau jos taip pat stipriai priklauso ir nuo žmonių, kurie atlieka užduotis, kurios negali būti automatizuotos.

Tarp Swan'o [7] ir Umeh'o [8] požiūriu galima įžvelgti sąsajas, nes blokų grandinė 1.0 atitinka kriptografinių valiutų grupę, blokų grandinė 2.0 atitikta koncepcijos įrodymo paslaugų ir išmaniųjų sutarčių grupę, o blokų grandinė 3.0 atitinka decentralizuotų autonominių organizacijų grupę. Tai rodo, kad skirtingi autoriai, blokų grandinės vystymosi eigą ir pritaikymo galimybes, mato panašiai.



1 pav. Pagrindinių blokų grandinių pritaikymų grupavimas [8]

Toks spartus blokų grandinės vystymasis ir pritaikomumo plėtra nėra atsitiktinė, nes blokų grandinės yra plačiai nagrinėjamos ir surandama būdų, kaip šią technologiją galima pritaikyti įvairiose srityse. Finansuose blokų grandinę galima pritaikyti atliekant mokėjimus, identifikuojant klientą, darant pirminį vertybinių platinimą, prekiaujant vertybiniais popieriais ir išvestinėmis finansinėmis priemonėmis [9]. Medicinoje yra siūlomi blokų grandinių pritaikymai kuriant decentralizuotą medicininių įrašų saugojimo ir valdymo sistemą; kovojant su padirbtų vaistų gamyba; atliekant į vartotoją orientuotus medicininius tyrimus; stebint opioidų receptų išrašymą; kuriant vėžio susirgimų registrą; priimant sprendimus dėl sveikatos draudimo išmokų [10, 11, 12, 13, 14]. Švietime blokų grandinę galima pritaikyti, kuriant paskirstytą sistemą, kurioje būtų kaupiama informacija apie mokslo pasiekimus, reputaciją ir apdovanojimus; sertifikatų išdavimui; intelektualinės nuosavybės valdymui [15, 16]. Taip pat siūlomi įvairūs pritaikymai ir kitose srityse: chemijos pramonėje [17], asmeninių duomenų apsaugos srityje [18], logistikoje ir tiekimo grandinėje [19]. Kai kurios iš šių idėjų jau turi veikiančiu prototipus ir yra išbandomos praktikoje:

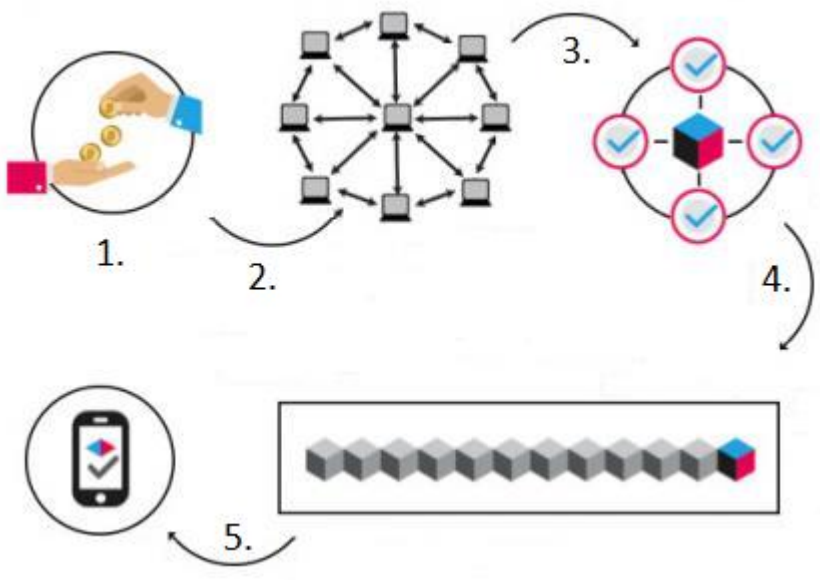
- *DApp for Smart Health* (DASH). Šios sistemos prototipas yra įgyvendintas eterio kriptografinės valiutos tinkle ir juo siekiama išbandyti elektroninių sveikatos įrašų sistemos veikimą [14].
- *OpenCerts*. Šis produktas leidžia išduoti mokslo sertifikatus eterio kriptografinės valiutos tinkle kaip išmaniąsias sutartis. Taip pat naudojant šią sistemą galima patikrinti ar sertifikatas galioja. Tai gali būti aktualu darbdaviui ar universitetui, kuris nori patikrinti gautą kandidato sertifikato galiojimą [16].
- *Grandbase*. Šis produktas yra įgyvendintas remiantis bitkoino kriptografinės valiutos tinklo pagrindu ir leidžia bent kuriuo metu patikrinti asmeninę kvalifikaciją [16].

– *uPort*. Sistema įgyvendinta eterio kriptografinės valiutos tinkle ir leidžianti saugiai dalintis informacija su klientais ar partneriais [16].

Nors pritaikymo atvejų yra sugalvota įvairiose srityse, tačiau daugiausiai pažengusi ir plačiausiai naudojama blokų grandinės technologija yra finansuose. Labiausiai paplitusios yra kriptografinės valiutos. Kriptografinių valiutų iš viso yra priskaičiuojama daugiau nei 5 500 [2] (duomenys 2020 gegužės 26 d.). Tačiau daugumos kriptografinių valiutų rinkos vertė yra nedidelė, o rinkoje dominuoja bitkoinas sudarantis 65,6 % ir eteris sudarantis 9,1 % nuo visos kriptografinių valiutų rinkos vertės [2] (duomenys 2020 gegužės 26 d.).

1.1.2. Blokų grandinės sandaros ir jos veikimo principų analizė

Blokų grandinės idėja buvo pasiūlyta 2008 metais ir įgyvendinta 2009 metais. Ši idėja buvo sugalvota, norint išvengti tarpininkų, atliekant elektroninius mokėjimus. Paprastai apdorojant mokėjimus yra pasikliaujama beveik vien tik finansinėmis institucijomis, kurios veikia kaip patikimos trečiosios šalys. Šis sistema daugeliu atvejų veikia gerai, tačiau kartais atsiranda ginčų dėl atliktų operacijų, todėl visiškai neatšaukiami sandoriai, naudojant šią sistemą, nėra įmanomi. Taigi toks veikimo principas, kelia šiokį tokį nepasitikėjimą, nes gavus mokėjimą tu nesi tikras, kad jis nebus užginčytas. Be to, tarpininkavimo paslaugos padidina operacijų sąnaudas, atsiranda apribojimai minimaliai operacijos sumai, nes mažos operacijos nėra finansiškai naudingos, dėl taikomų mokesčių. Norint to išvengti, yra reikalinga elektroninė mokėjimo sistema, kuri leidžia sudaryti sandorius tarpusavyje, nereikalaujant patikimos trečiosios šalies. Taigi buvo pasiūlyta blokų grandinės idėja, kuri yra pagrįsta kriptografiniais įrodymais, o ne pasitikėjimu trečiąja šalimi [20].



2 pav. Blokų grandinės veikimo diagrama[1]

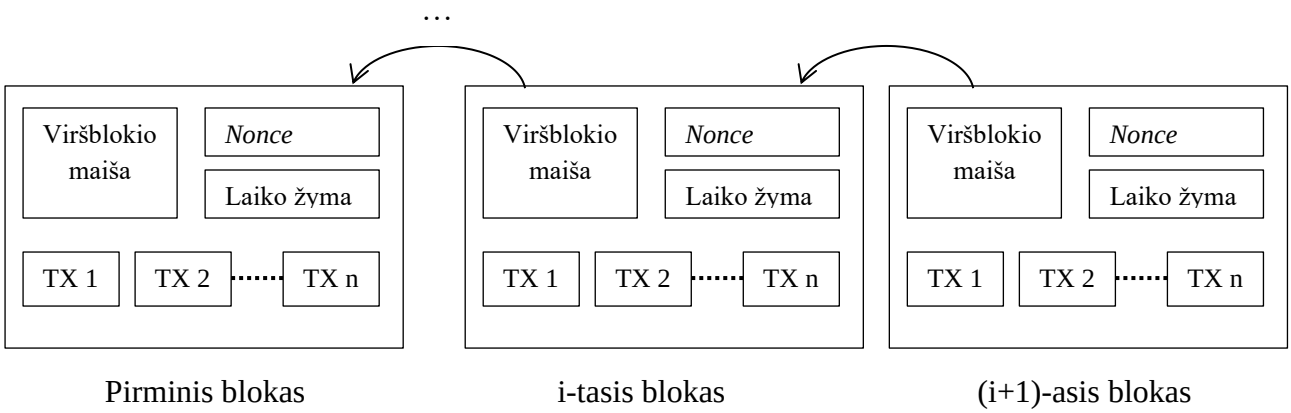
Blokų grandinės veikimą galima būtų trumpai aprašyti taip (2 pav.):

1. Kažkuris iš naudotojų nori atlikti sandorį.

[1] Pagal https://commons.wikimedia.org/wiki/File:Blockchain_CS_BLK_0.jpg

2. Norimas atlikti sandoris yra ištransliuojama į lygiarangį tinklą, kurį sudaro kompiuteriai, kitaip dar vadinami mazgais.
3. Tinklo mazgai naudodami specialius algoritmus patikrina sandorį ir naudotojo statusą. Tikrinamas sandoris gali būti susijęs su kriptografine valiuta, sutartimi, įrašu ar kita informacija.
4. Kai sandoris patikrinamas, jis yra prijungiamas prie kitų sandorių, taip sudarant naują duomenų bloką. Naujasis duomenų blokas yra prijungimas prie egzistuojančios blokų grandinės.
5. Sandoris laikomas įvykdytu.

Pirmiausia, norint detaliau suprasti blokų grandinės veikimą, reiktų panagrinėti blokų grandinės architektūrą. Blokų grandinė yra blokų seka, kuri turi visų sandorių sąrašą (3 pav.). Kiekvienas blokas turi nuorodą į prieš tai buvusį bloką, kuris yra vadinamas viršblokiu. Pirmasis blokas, kuris neturi viršblokio, yra vadinamas pirminiu bloku [21].



3 pav. Blokų grandinės pavyzdys

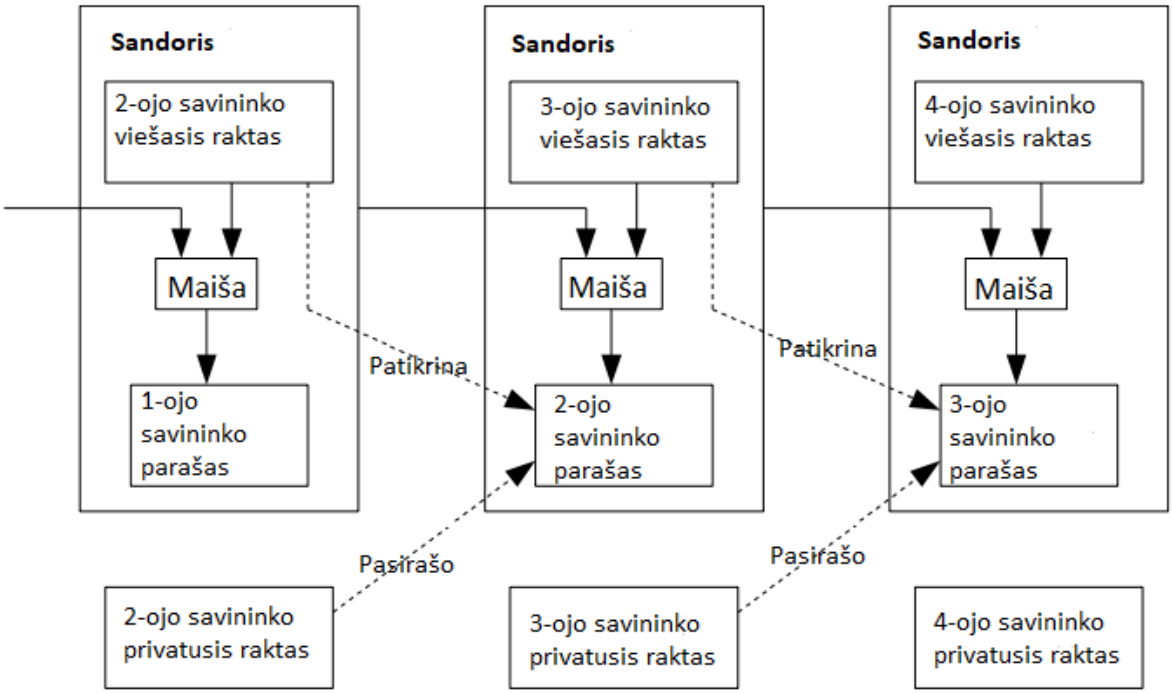
Bloką sudaro bloko antraštė ir bloko pagrindinė dalis. Paprastai bloko antraštę sudaro [21]:

- Bloko versija. Nurodo, kurių bloko tikrinimo taisyklių reikia laikytis.
- Viršblokio maiša. 256 bitų maišos reikšmė, kuri rodo į prieš tai buvusį bloką.
- Merkle medžio šakninė maiša. Visų sandorių esančių bloke maišos reikšmė.
- Laiko žyma. Dabartinė laiko žyma sekundėmis nuo 1970-01-01 00:00:00 UTC.
- *nBits*. Dabartinis maišos objektas suspaustu formatu.
- *Nonce* (sutrumpinimas iš angliško termino „number only used once“). 4 baitų laukas, kuris prasideda nuo nulio ir auga su kiekvienu maišos apskaičiavimu.

Bloko pagrindinę dalį sudaro sandorių skaitiklis ir sandoriai. Bloke talpinamų sandorių skaičius priklauso nuo bloko dydžio ir kiekvieno sandorio dydžio.

Norint sudaryti sandorį, kiekvienas blokų grandinės naudotojas turi turėti privatųjį ir viešąjį raktą. Sandorių pasirašymas ir patikrinimas yra pavaizduota schemoje (4 pav.). Sandorį inicijuojantis naudotojas, pasirašant sandorį, turi panaudoti privatų raktą. Paimamas prieš tai buvusio sandorio maišos kodas, ir jis pasirašomas privačiu raktu. Prie sandorio taip pat pridedamas ir kitos sandorio pusės (gavėjo) viešasis raktas. Kadangi viešieji raktai yra prieinami, tai gavėjas gali atlikti patikrinimą ir įsitikinti dėl tikrosios nuosavybės. Gavėjas šioje vietoje negali būti tikras ar nuosavybė nėra perduodama antrą kartą. Norint išvengti trečiosios šalies paslaugų, visi sudaryti sandoriai turi būti viešai skelbiami. Turint viešai paskelbtus visus sandorius, galima atsekti ir nustatyti tikrąjį savininką. Dar vienas svarbus dalykas, kurį reikia turėti, tai susitarimas dėl

sistemos, kuri parodytų kokia tvarka sandoriai buvo sudaryti. Savininkui sudarius du vienodus sandorius su ta pačia nuosavybe, reikia žinoti, kuris buvo įvykdytas pirma, kad antrąjį būtų galima atmesti. Taigi sandorio sudarymo laikui nustatyti yra naudojami laiko žymos serveriai. Šie serveriai paima elementą, kuriam reikia uždėti laiko žymą, ir ją uždeda. Laiko žyma įrodo, kad duomenys turėjo tuo metu egzistuoti, kai pateko į maišą [20]. Taigi, panaudojant privačiuosius ir viešuosius raktus, padarant viešai prieinamus visus sandorius ir uždedant laiko žymas, yra sukuriamas sprendimas leidžiantis atsekti nuosavybę ir vykdyti sandorius.



4 pav. Sandorių sudarymo blokų grandinėje pavyzdys [20]

Kitas svarbus aspektas yra sandorių patikrinamas, kurį atlieka blokų grandinės tinklui priklausantys mazgai. Tam, kad sandoris būtų patvirtintas turi būti priimtas bendras sutarimas. Bitkoino tinkle bendram sutarimui pasiekti yra naudojama atlikto darbo įrodymo (angl. *proof of work*) sistema, kuri vertina centrinių procesorių darbą [20]. Atlikto darbo įrodymo sistema reikalauja sudėtingų skaičiavimo procesų. Kiekvienas mazgas tinkle skaičiuoja nuolatos besikeičiančio bloko antraštės maišos reikšmę. Sutarimas reikalauja, kad ši reikšmė būtų lygi arba mažesnė už nurodytą skaičių. Kai vienas iš mazgų suranda reikiamą reikšmę, visi kiti mazgai turi patvirtinti reikšmės teisingumą. Tuomet sandorių rinkinys, kuris buvo naudojamas atliekant skaičiavimus, yra įtraukiamas kaip naujas blokas į blokų grandinę [21]. Bitkoino blokų grandinė yra kritikuojama už tokį beprasmį resursų švaistymą. Vertinama, kad bitkoino blokų grandinė per metus suvartoja tiek elektros energijos kiek visa Austrija, ir prognozuojama, kad elektros energijos suvartojimas augs toliau [22]. Kai kurios kriptografinės valiutos stengiasi spręsti prasmingus matematinius uždavinius, kurie dar turėtų praktinės naudos. Pavyzdžiui, *primecoin* kriptografinė valiuta ieško pirminių skaičių [23]. Taip pat siūlomos kitos alternatyvos, kurios nereikalautų tiek daug resursų. Viena iš alternatyvų yra vietos įrodymo sistema (angl. *proof of space*), kurioje vietoje centrinių procesorių darbo yra prašoma suteikti atminties diske [24]. Kita alternatyva, taip pat reikalaujanti mažiau resursų, yra dalyvavimo įrodymo (angl. *proof of stake*). Šioje sistemoje mazgas turi pateikti tinklui įrodymą, kad

jis turi priėjimą prie nuosavybės. Kuriant bloką reikia nusiųsti nuosavybę sau, tai patvirtina nuosavybės teises [25]. Taigi sandoriams patikrinti yra naudojami tinklo mazgai, kurie turi panaudoti savo resursus, kad būtų priimtas bendras sutarimas ir patvirtintas sandoris.

Blokų grandinės technologiją tokia populiaria daro šios blokų grandinės savybės, kurios priklauso ir nuo aukščiau aprašytos blokų grandinės architektūros ir veikimo principų [6, 26, 27]:

- Decentralizuota. Pagrindinė blokų grandinės savybė yra ta, kad technologija neturi pasikliauti centralizuotu mazgu.
- Realaus laiko įrašai. Paskirstyta duomenų bazė (angl. *distributed ledger*) atnaujinamos realiuoju laiku, kai vyksta sandoriai ar kiti įvykiai.
- Nekintantys įrašai. Blokų grandinės technologija leidžia įmonėms kurti nuolatinius, nepakeičiamus operacijų įrašus.
- Skaidri. Blokų grandinės sistemos įrašas yra prieinamas visiems mazgams, todėl blokų grandine galima pasitikėti.
- Anonimiška. Blokų grandinės technologija palengvina tinklo naudotojui galimybę prisidengti slapyvardžiu.
- Atviro kodo. Dauguma blokų grandinės sistemų yra atviros visiems, įrašai gali būti patikrinti viešai ir žmonės gali naudoti blokų grandinės technologijas sukurti savo norimoms sistemoms.
- Kibernetinis saugumas. Kol kas jokia blokų grandinė nebuvo nulaužta ir ja nebuvo manipuliuojama, tačiau kompanijos ir technologijos susijusios su ja yra nuolatinis įsilaužėlių taikinys.

Šie privalumai ir technologijos inovatyvumas daro šią technologiją labai patrauklia ir atsiranda daug jos panaudojimo atvejų. Vienas iš pirmųjų ir plačiausiai iki šiol naudojamų blokų grandinės pritaikymo atvejų yra kriptografinės valiutos.

1.2. Kriptografinės valiutos samprata

Kaip buvo minėta anksčiau blokų grandinės technologija pirmiausiai buvo pritaikyta bitkoino kriptografinei valiutai. Kriptografine valiuta yra laikoma skaitmeniniu turtu, kuriam yra sukurta mainų terpė, paremta kriptografijos technologija, siekiant užtikrinti operacijų srautą ir kontroliuoti naujos valiutos atsiradimą [28]. Netrukus po bitkoino atsirado ir kitos kriptografinės valiutos, kurių dabar priskaičiuojama daugiau nei 5000, o didžiausios pagal rinkos vertę yra eteris, *ripple*, *tether* [2]. Toliau šiame poskyryje apžvelgiamos dvi pagal rinkos vertę didžiausios kriptografinės valiutos – bitkoinas ir eteris.

1.2.1. Bitkoino blokų grandinės apžvalga

Bitkoinas yra koncepcijų ir technologijų rinkinys, sudarantis skaitmeninės pinigų ekosistemos pagrindą. Valiutos vienetai yra vadinami bitkoinais. Jie yra naudojami kaupti ir perduoti vertę tarp bitkoino tinklo dalyvių. Tinklo dalyviai tarpusavyje bendrauja naudodami bitkoino protokolą. Bitkoino protokolas yra atvirojo kodo, todėl jį galima naudoti įvairiuose įrenginiuose, įskaitant išmaniuosius telefonus. Naudotojai tinkle su bitkoinais gali atlikti lygiai tokias pat operacijas, kaip ir su tradicinėmis valiutomis: persiųsti valiutą kitiems tinklo dalyviams (tai gali būti fiziniai asmenys ar organizacijos), atsiskaityti už perkamas prekes, gauti apmokėjimus už parduodamas prekes. Skirtumas nuo tradicinės valiutos yra tas, kad bitkoinas yra visiškai virtuali, t. y., neturi

jokių fizinių ir skaitmeninių pinigų. Bitkoinas yra labai tinkama pinigų internete forma, nes jis yra atviro kodo, lengvai prieinama, saugi ir greita [29].

Norint gauti ir atlikti mokėjimus kriptografinėmis valiutomis reikia turėti kriptografinę valiutos piniginę. Piniginė sukuria adresą, kuris yra tarsi banko sąskaitos atitikmuo. Bitkoino adresas yra unikali raidžių ir skaitmenų seka, kuria naudodamasis vartotojas gali pradėti gauti mokėjimus. Paprastai bitkoinus galima nusipirkti biržoje arba gauti apmokant už prekes ar paslaugas. Aptariant blokų grandinės technologiją jau buvo minėta, kad viena iš iškylančių problemų yra pinigų išleidimo du ar daugiau kartų galimybė, nes bitkoinai yra virtualūs ir neturi jokio fizinio ar skaitmeninio pavidalo. Šiai problemai spręsti yra sukurta viešai prieinama paskirstyta duomenų bazė, kurioje yra saugomi visi sudaryti sandoriai. Tai reiškia, kad bent kuris bitkoino tinklo dalyvis, turintis tam skirtą programinę įrangą, gali parsisiųsti visus įvykusius sandorius ir patikrinti atliekamo sandorio pagrįstumą. Naujai atliekami sandoriai yra patikrinami su blokų grandine, kad būtų įsitikinama, kad siunčiami bitkoinai nėra jau išleisti [30].

Sandorius atliekamus bitkoinų tinkle reikia apdoroti ir sukurti bloką, kuris būtų prijungtas prie blokų grandinės. Sandorių apdorojimas atliekamas decentralizuotame tinkle, kitaip sakant lygiarangiame tinkle. Net ir decentralizuotame tinkle sandorių apdorojimas reikalauja resursų. Bitkoinų tinklo dalyviams, norintiems apdoroti sandorius, yra sukurtas procesas vadinamas gavyba. Gavyba yra procesas, kurio metu bitkoinų tinklo dalyviai atiduoda savo kompiuterio skaičiavimo galią mainais į bitkoino kriptografinę valiutą. Apytiksliai kas 10 minučių yra sukuriamas naujas blokas, kuriame yra išsaugomi naujai patvirtinti sandoriai. Sandorių tikrinimas yra tarsi varžybos, kuriose visi dalyviai varžosi kas greičiau išspręs matematinį uždavinį. Uždavinio sudėtingumas yra reguliuojamas dinamiškai kas 2 016 blokų, kad visą laiką uždavinio sprendimas užtruktų apie 10 minučių. Jei prie bitkoinų tinklo prisijungtų naujų dalyvių, norinčių užsiimti bitkoinų gavybą, ir taip padidintų skaičiavimo galią, tuomet uždavinys taptų sudėtingesnis ir jo sprendimas vis tiek užtruktų apie 10 minučių [29]. Apdovanojimas skiriamas už sėkmingą uždavinio išsprendimą yra vadinamas bloko atlygiu. Laikui bėgant blogo atlygis kinta. Pačioje pradžioje bloko gavybos atlygis buvo 50 bitkoinų. Atlygio suma kas 210 000 blokų yra sumažinamas per pus (apytiksliai kas ketverius metus). Dabar bloko atlygis yra 6,25 bitkoino, ir 2024 metų gegužės 11 d. turėtų sumažėti per pus iki 3,125 bitkoino [31]. Maksimali bitkoinų suma yra fiksuota ir yra lygi 21 milijonui bitkoinų, jei visi turėtų būti baigti išgauti 2140 metais [29]. Gavybos procesas yra reikalingas tam, kad būtų apdoroti bitkoinų sandoriai ir už šį procesą yra skiriamas bloko atlygis, kuris skatina tinklo dalyvius varžytis sprendžiant matematinį uždavinį.

Surasti sprendimą matematiniam uždaviniui reikia daug resursų. Norint išspręsti matematinį uždavinį, reikia du kartus apskaičiuoti SHA-256 maišą. Maišos funkcija yra funkcija, kuri naudojama susieti bent kokius duomenis su fiksuoto ilgio reikšme. SHA-256 algoritmas sugeneruoja beveik unikalią 256 bitų reikšmę. Norint išspręsti uždavinį reikia surasti *nonce* reikšmę su kuria bloko maiša atitiktų arba būtų mažesnė už tikslinę maišą. Taigi norint išspręsti uždavinį reikia spėti *nonce* reikšmę, ją įdėti į bloko antraštę, apskaičiuoti bloko dvigubą maišą ir gautą maišą palyginti su tiksline maiša. SHA-256 yra sukurtas taip, kad algoritmo sugeneruotos reikšmės nebūtų įmanoma atstatyti į pradinius duomenis. Todėl norint surasti sprendimą uždaviniui reikia taikyti jėgos metodą. Tai reiškia, kad jei uždavinio sunkumas yra padvigubinamas, tai jį išspręsti reiks maždaug dvigubai daugiau bandymų spėjant *nonce* reikšmę [32]. Galiausiai atradus sprendimą yra sukuriamas blokas, kuris yra patalpinamas į blokų grandinę.

1 lentelė. Bloko struktūra [29]

Lauko dydis	**Lauko pavadinimas**	**Lauko apibūdinimas**
4 baitai	Bloko dydis	Bloko dydis baitais
80 baitų	Bloko antraštė	Metaduomenys apie bloką
1–9 baitai	Sandorių skaitiklis	Kiek iš viso sandorių yra bloke
Kintantis	Sandoriai	Sandoriai esantys šiame bloke

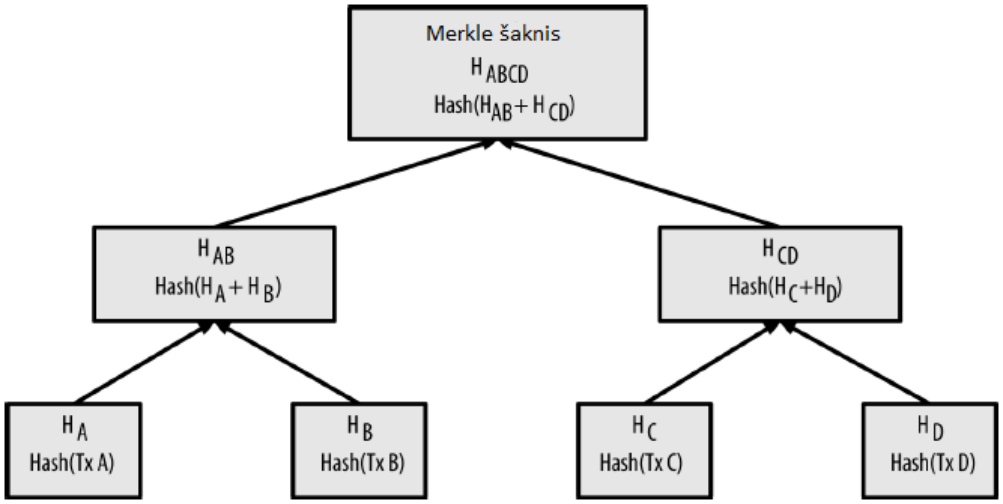
Blokų grandinė yra sudaryta iš blokų, o kiekvieną bloką sudaro bloko dydžio, bloko antraštės, sandorių skaitiklio ir sandorių laukai (1 lentelė). Bloko dydis parodo, kiek baitų užima visas blokas. 625 229 blokas, kuris buvo sukurtas 2020 metų balandžio 10 dieną ir jo dydis buvo 419 205 baitai. Pirminiame variante maksimalus bloko dydis buvo nustatytas 1 000 000 baitų. Tokio dydžio bloke gali tilpti apie 4000 sandorių. Kadangi kiekvienas blokas yra suformuojamas kas 10 minučių, tai esant tokiems apribojimams per sekundę galima apdoroti 7 sandorius per sekundę. Tai yra gana nedidelis kiekis ir populiarėjant bitkoino kriptografinei valiutai buvo nerimaujama, kad bus pasiektas šitas limitas. Buvo įvairiausių pasiūlymų kaip praplėsti šį limitą, tačiau galiausiai limitas buvo praplėstas įgyvendinus *SegWit* procesą. *SegWit* procesas atskiria skaitmeninius parašus nuo sandorių informacijos ir taip padidina lankstumą. Su šiuo pakeitimu maksimalus bloko dydis gali būti 4 000 000 baitų [33]. Blokų antraštę sudaro metaduomenys apie bloką ir ji bus apžvelgta toliau. Sandorių skaitiklis parodo, kiek iš viso bloke yra sandorių. Patys sandoriai – tai pagrindiniai duomenys, kurie bloke užima daugiausiai vietos ir dėl jų reikėjo atlikti pakeitimus dėl maksimalaus bloko dydžio. Taigi bitkoino protokolas nėra pastovus ir jis nuolat atnaujinamas, kad būtų pritaikytas prie dabartinių rinkos poreikių.

2 lentelė. Bloko antraštės struktūra [29]

Lauko dydis	**Lauko pavadinimas**	**Lauko apibūdinimas**
4 baitai	Versija	Versijos numeris, kad būtų galima sekti pakeitimus
32 baitai	Prieš tai einančio bloko maiša	Nuoroda į prieš tai einantį bloką (viršblokį)
32 baitai	Merkle šaknis	Šio bloko sandorių Merkle medžio šaknies maiša
4 baitai	Laiko žyma	Apytikslė bloko sukūrimo data (sekundės nuo 1970-01-01 00:00:00 UTC)
4 baitai	Tikslinis sudėtingumas	Parodo atlikto darbo įrodymo sudėtingumą bloke
4 baitai	*Nonce*	Skaitiklis naudojamas atlikto darbo įrodymo algoritme

Bloko antraštę sudaro tokie laukai kaip versija, prieš tai buvusio bloko maiša, Merkle šaknis, laiko žyma, tikslinis sudėtingumas ir *nonce*. Versija parodo, kuri programinės įrangos / protokolo versija yra naudojama, kad būtų galima atsekti pakeitimus. 625 229 bloko versija yra 536 870 912. Iš tikrųjų yra keturios versijos. Pirmoji versija buvo naudojama nuo pirminio bloko. Antroji versija atsirado 2012 metais ir papildomai prie bloko reikėjo nurodyti bloko dydį, kuris parodo, per kiek blokų dabartinis blokas yra nutolęs nuo pirminio bloko. Trečiojoje versija atsirado 2015 metais ir joje reikalavo, kad parašai bloke koduoti naudojant DER kodavimą. Ketvirtojoje versijoje buvo pridėta nauja operacija OP_CHECKLOCKTIMEVERIFY, kuri leidžia uždėti laiko apribojimą bitkoinų išleidimui [29, 33]. Tačiau naujausiuose blokuose versijos jau nebeatitinka apibrėžtų

versijų, nes buvo pastebėta, kad versiją nustatant atsitiktinai, paspartėja bitkoinų gavybos procesas. Todėl dabartinės blokų versijos laukas neatspindi tikrosios savo paskirties. Prieš tai einančio bloko maiša yra prieš tai einančio bloko antraštės maiša, apskaičiuota naudojant SHA-256 algoritmą. Merkle šaknis, tai yra maišos medžio šakninio mazgo maiša (5 pav.). Kiekvienam sandoriui yra apskaičiuojama dviguba maiša, naudojant SHA-256 algoritmą, t. y., kiekvienam medžio lapui, kuris atitinka sandorį yra du kartus pritaikomas SHA-256 algoritmas ir gaunamos maišos H_A, H_B, H_C, H_D. Tuomet sujungiamos dvi lapų maišos ir joms du kartus pritaikomas SHA-256 algoritmas, taip gaunami mazgai H_{AB} ir H_{CD}. Galiausiai vėl sujungiama po du mazgus iš antro lygio ir du kartus pritaikomas SHA-256 algoritmas ir taip gaunama Merkle medžio šakninio mazgo maiša H_{ABCD} [29]. Laiko žyma rodo apytikslį bloko sukūrimo laiką sekundėmis, kuris yra skaičiuojamas nuo pradinės datos 1970-01-01 00:00:00 UTC laiko zonoje. Tikslinis sudėtingumas yra parametras, kuris nusako skaičiavimo sudėtingumą. Šis tikslinis sudėtingumas yra koreguojamas pagal tinklo skaičiavimo pajėgumą, taip, kad kiekvienas naujas blokas būtų sukuriamas apytiksliai kas 10 minučių. Ir paskutinis laukas yra *nonce*, kuris yra keičiamas gavybos metu norint surasti mažesnę maišą.



5 pav. Mazgų apskaičiavimas Merkle medyje [29]

Informacija apie sandorius yra pati svarbiausia bloke. Bloke yra laukas, skirtas sandorių skaitliukui, kuris parodo sandorių skaičių bloke, ir laukas, skirtas surašyti informacijai apie bloke esančius sandorius. Visa kita informacija yra reikalinga tam, kad sandoriai būtų priimti, apdoroti, patikrinti ir galiausiai pridėti į blokų grandinę. Bitkoinas neturi sąskaitos likučio, o naudoja UTXO modelį likučiui sužinoti. UTXO modelio atveju yra saugomos neišleistų sandorių išvestys, t. y., tokie sandoriai, kurie gali būti įvestimis naujuose sandoriuose. Norint sužinoti turimą likutį, reikia apskaičiuoti visų neišleistų sandorių sumą. Sandorių struktūra sudaro versijos, įvesčių skaitiklio, įvesčių, išvesčių skaitiklio, išvesčių ir užrakinimo laiko laukai (3 lentelė). Kiekvienas bitkoino sandoris sukuria išvestį bloke ir papildo UTXO rinkinį įrašu apie turimus bitkoinus, kurie yra pripažįstami visame tinkle ir gali būti savininko išleisti ateityje. Sandorio išvestį sudaro suma ir kriptografinė dėlionė, nusakanti sąlygas, reikalingas išvesties išleidimui. Sandorio įvestys nustato, kuri UTXO bus naudojama. Atliekant sandorį, piniginė iš jos valdomų UTXO parenka tą, kurio turima vertė yra pakankama atlikti mokėjimui. Kartais mokėjimui gali pakati vieno UTXO, o kartais gali prireikti ir kelių. Įvestį sudaro sandorio identifikatorius, rodantis į sandorį, turintį UTXO, išvesties indeksas, atrakinimo skripto ilgis, pats skriptas ir sekos numeris. Paprastai atrakinimo skriptas yra skaitmeninis parašas ir viešasis raktas, įrodantis bitkoinų nuosavybę [29].

Taip pat sandoriams dar gali būti nurodytas užrakinimo laikas, t. y,. laiko žyma arba bloko numeris, kuris parodo, nuo kada bus galima išleisti gautus bitkoinus. Sandorių apdorojimui UTXO modelis buvo pasirinktas todėl, kad jis yra tinkamas konkurencingai ir paskirstytai aplinkai, kuri yra būdinga blokų grandinei. Tačiau jis yra kritikuojamas dėl riboto programavimo galimybių, nes jį naudojant sudėtingesni skaičiavimai beveik neįmanomi [34].

3 lentelė. Sandorių struktūra [29]

Lauko dydis	**Lauko pavadinimas**	**Lauko apibūdinimas**
4 baitai	Versija	Nurodomos, kurios taisyklės taikomos sandoriui
1–9 baitai	Įvesčių skaitiklis	Kiek įvesčių yra įtraukta į bloką
Kintantis	Įvestys	Viena ar kelios sandorio įvestys
32 baitai	Sandorio maiša	Nuoroda į sandorį turintį UTXO
4 baitai	Išvesties indeksas	UTXO indeksas (numeruojamas nuo 0)
1–9 baitai	Atrakinimo skripto dydis	Atrakinimo skripto dydis baitais
Kintantis	Atrakinimo skriptas	Skriptas, kuris atitinka UTXO užrakto skripto sąlygas
4 baitai	Sekos numeris	Naudojamas užrakinimo laikui arba neveiksnus (0xFFFFFFFF)
1–9 baitai	Išvesčių skaitiklis	Kiek išvesčių yra įtraukta į bloką
Kintantis	Išvestys	Viena ar kelios sandorio išvestys
8 baitai	Suma	Bitkoinų suma satošiais
1–9 baitai	Užrakinimo skripto ilgis	Užrakinimo skripto ilgis baitais
Kintantis	Užrakinimo skriptas	Skriptas, apibrėžiantis sąlygas, kurių reikia išvesčiai išleisti
4 baitai	Užrakinimo laikas	Laiko žyma arba bloko numeris

Bitkoinas turi neįprastą standartinėje bankininkystėje naudojamą likučių valdymo mechanizmą. Todėl, norint sužinoti turimą kriptografinės valiutos likutį, reikia atlikti papildomų veiksmų. Tačiau, žiūrint į pačią blokų grandinės struktūrą, ji nėra per daug sudėtinga ir ją panaudojant galima nesunkiai atsekti vykdomus sandorius.

1.2.2. Eterio blokų grandinės apžvalga

Eteris yra atviro kodo, globaliai decentralizuota skaičiavimo sistema, kuri vykdo programas vadinamas išmaniąja sutartimi. Eteris turi tokius pat elementus, kaip ir kitos blokų grandinės – lygiarangį tinklą jungianti dalyvius, bendram susitarimui naudojamą atlikto darbo įrodymo sistemą, kriptografinius elementus (skaitmeninis parašas ir maiša), skaitmeninę valiutą. Tačiau eterio pagrindinis tikslas nėra elektroninių mokėjimų ekosistema. Nors skaitmeninė valiuta eteris yra būtina eterio tinklo veikimui, tačiau ji yra labiau pagalbinė valiuta, kuria yra mokama už naudojimąsi eterio platforma. Skirtingai nuo bitkoino, kuris turi labai ribotą skriptų kalbą, eteris yra sukurtas kaip bendrosios paskirties programuojama blokų grandinė, kuri paleidžia virtualią mašiną, galinčią vykdyti neriboto sudėtingumo kodą [35]. Eteris, lyginant su bitkoinu, sukuria universalesnį tinklą, kurio galimybės neapsiriboja vien tik kriptografine valiuta.

Eterio idėja 2013 metais pristatė Vitalikn'as Buterinn'as. Ši idėja atsirado dėl to, kad buvo matomi bitkoino modelio trūkumai ir buvo norima žengti toliau, nei blokų grandinės panaudojimas kriptografinėms valiutoms. Pasidalinus pirminiu eterio protokolo aprašymu, prie jo pradėjo jungtis entuziastai, norėję prisidėti prie šio protokolo vystymo. Pirmoji veikianti versija buvo paleista 2015 metais [36]. Po bitkoino atsiradimo jau buvo praėję 5 metai, todėl kuriant šį protokolą jau buvo informacijos apie blokų grandinės veikimą. Kuriant naują protokolą buvo galima atsižvelgti į bitkoino ribotumus ir sukurti universalesnį protokolą.

Taip pat, kaip ir bitkoino atveju, norint įsigyti eterio kriptografinės valiutos, pirmiausia reikia turėti kriptografinę valiutos piniginę. Eterio piniginė laiko raktus ir gali atlikti sandorius jūsų vardu. Kiekvienas privatus raktas yra tarsi sąskaitos atitikmuo. Naudojant privatų raktą, galima kontroliuoti lėšas ir išmaniąsias sutartis. Jeigu privatus raktas bus pamestas, tuomet bus prarastas priėjimas prie lėšų ir sutarčių. Niekas negali padėti atstatyti priėjimo prie lėšų ir sutarčių be privatus rakto [35]. Tas pats yra ir su bitkoino kriptografinės valiutos piniginėmis. Gavybos procesas principai tiek bitkoino, tiek eterio atveju yra panašūs. Eterio gavybos metu yra vykdomos išmaniosios sutartys ir patvirtinama sandorių tvarka. Gavybos metu sandoriai yra apdorojami ir patalpinami į bloką. Kaip ir bitkoino atveju yra varžomasi kas greičiau išspręs matematinį uždavinį. Šiuo metu eteris naudoja atlikto darbo įrodymo sistemą, tačiau ateityje yra numatytas perėjimas prie dalyvavimo įrodymo sistemos. Nauji blokai yra kuriami kas 15 sekundžių ir sprendžiamo uždavinio sudėtingumas dinamiškai koreguojamas, kad būtų išlaikytas pastovus blokų sukūrimo laikas. Tuo pačiu metu įvykdžius sandorį eterio ir bitkoino tinkle, jis eterio blokų grandinėje atsiras greičiau, nei bitkoino blokų grandinėje. Kadangi eterio blokai yra kuriami dažniau, tai ir jų dydis yra mažesnis. Viename eterio bloke gali tilpti apie 70 sandorių [37]. Už bloko apdorojimą, kaip ir bitkoino tinkle, skiriamas bloko atlygis, kuris iš pradžių buvo apie 5 eterius, bet po to buvo sumažintas iki 3 eterių, o dabar yra apie 2 eterius [38]. Dar vienas bitkoino ir eterio skirtumas yra toks, kad eterio maksimaliai galima išgauti suma nėra apribota. Be to, eteris nenaudoja UTXO modelio, o naudoja sąskaita pagrįstą modelį, kuris yra įprastas tradicinėje bankininkystėje [34]. Bitkoino ir eterio veiklos principai yra labai panašūs, tačiau eteris turi patobulinimų, kurie išplečia eterio pritaikymo galimybes.

Kaip ir bitkoino atveju eterio blokas yra sudarytas iš antraštės ir pagrindinės dalies. Vienas skirtumas, kad maišai skaičiuoti naudojama Keccak 256 (kitaip dar vadinama SHA3) maišos funkcija. Bloko antraštę sudaro tokia informacija [39]:

- Viršblokio maiša (*parentHash*). Viršblokio antraštės Keccak 256 bitų maiša.
- Dėdžių blokų maiša (*ommersHash*). Dėdžių blokų Keccak 256 bitų maiša. Kadangi laikas tarp blokų sukūrimo yra tik 15 sekundžių, tai tuo pat metu būti išgauti keli galiojantys blokai. Blokai, kurie nepatenka į sutartą blokų grandinę, yra vadinami dėdės bloku (viršblokio poblokiai neaptekę į sutartą blokų grandinę).
- Naudos gavėjas (*beneficiary*). 160 bitų adresas, kuris nurodo kam turi būti pervesti visi mokesčiai surinkti už bloko gavybą.
- Būsenų medžio maiša (*stateRoot*). Būsenų medžio šakninio mazgo Keccak 256 bitų maiša, kai visi sandoriai jau yra įvykdyti. Pasaulio būsenos (angl. *world state*) suriša adresus su sąskaitų būsena (likutis ir kita informacija). Ši informacija nėra saugoma blokų grandinėje, bet yra laikoma modifikuotame Merkle Patricia medyje.
- Sandorių medžio maiša (*transactionsRoot*). Bloko sandorių medžio šakninio mazgo Keccak 256 bitų maiša.

- Kvitų medžio maiša (*receiptsRoot*). Bloko sandorio kvitų medžio šakninio mazgo Keccak 256 bitų maiša. Kiekvieną kartą įvykdžius sandorį yra sugeneruojamas sandorio kvitas, turintis informaciją apie sandorio įvykdymą.
- Bloom žurnalas (l*ogsBloom*). Bloom filtras, sudarytas iš indeksuojamos informacijos. Taupant vietą, sugeneruotas įvykių žurnalas nesaugomas bloke, o saugomas tik Bloom filtras, kuris gali būti naudojamas surasti įrašus žurnale.
- Sudėtingumas (*difficulty*). Reikšmė atitinkanti šio bloko sudėtingumo lygį.
- Numeris (*number*). Kelintas šis blokas yra blokų grandinėje.
- Degalų limitas (*gasLimit*). Degalų sunaudojimo riba vienam blokui. Kiekvienas sandoris naudoja degalus. Šis limitas nurodo maksimalų degalų kiekį, kurį galima sunaudoti vykdant sandorį bloke. Tai būdas apriboti sandorių skaičių bloke.
- Sunaudoti degalai (*gasUsed*). Kiek degalų buvo sunaudota apdorojant sandorius šiame bloke.
- Laiko žyma (*timestamp*). Bloko sukūrimo data sekundės nuo 1970-01-01 00:00:00 UTC.
- Papildomi duomenys (*extraData*). Baitų masyvas, kuriam yra su bloku susijusių duomenų.
- Sumaišyta maiša (*mixHash*). 256 bitų maiša, kuri kartu su *nonce* įrodo, kad šiam blokui buvo atliktas pakankamas kiekis skaičiavimų.
- *Nonce* (*nonce*). 64 bitų maiša, kuri kartu su sumaišyta maiša įrodo, kad šiam blokui buvo atlikta pakankamai skaičiavimų.

Bloko pagrindinė dalis apima sandorių sąrašą ir dėdžių blokų sąrašą. Operacijos gali būti dviejų tipų: piniginiai pavedimai ir išmaniosios sutartys. Apie sandorius saugoma tokia informacija [39]:

- *Nonce* (*nonce*). Parodo kelintas iš viso yra siuntėjo sandoris (pradedama numeruoti nuo 0).
- Degalų kaina (*gasPrice*). Vertė vėjais, kuri yra mokama už degalų vienetą, apskaičiuojant šio sandorio vykdymo išlaidas.
- Degalų limitas (*gasLimit*). Maksimali degalų suma, kuri gali būti sunaudota vykdant sandorį.
- Gavėjas (*to*). Adreso numeris, kuriam yra pervedama suma.
- Suma (*value*). Suma vėjais, kuri bus pervedama gavėjui.
- v,r,s (v,r,s). Reikšmės, kurios yra naudojamos kriptografiniame sandorių paraše tam, kad būtų galima nustatyti sandorio siuntėją.
- Inicializavimas (*init*). Šis laukas yra naudojamas tik išmaniosioms sutartims. Tai yra kodas, kuris naudojamas sandorio inicializavimui.
- Duomenys (*data*). Šis laukas naudojamas vertės perdavimui ir žinutės siuntimui išmaniajai sutarčiai. Lauke yra nurodomi įvesties žinutės duomenys.

Bitkoino ir eterio blokų struktūros yra gana skirtingos, tačiau abi šios technologijos sugeba puikiai veikti vykdant kriptografinių valiutų pavedimus. Lyginant bloko antraštes galima teigti, kad bitkoino antrašte yra paprastesnė. Tačiau, lyginant sandorių patalpinimą į bloką, paprastesni ir mums artimesnis mechanizmas yra eterio atveju.

1.3. Sukčiavimo atvejų blokų grandinėje vertinimas

Bitkoinas ir eteris yra didžiausios kriptografinės valiutos pagal rinkos vertę, kuri bendrai šioms valiutoms siekia beveik 187 milijardus dolerių [2] (2020 gegužės 26 d.). Nenuostabu, kad šios valiutos sulaukia sukčių dėmesio, kurie nori nelegaliais būdais pasipelnyti. Kol kas, jokia blokų

grandinė, nebuvo nulaužta ir ja nebuvo manipuliuojama, tačiau kompanijos ir technologijos, susijusios su ja, yra nuolatinis įsilaužėlių taikinys [26]. Vertinama, kad 2019 metais sukčiavimas sudarė didžiausią dalį iš neteisėtų veiklų, antroje vietoje palikdama juodosios rinkos operacijas [5]. Išskiriami tokie sukčiavimo atvejai – Ponzi schemos, išpirkos reikalaujanti programinė įranga, šantažuojantys laiškai, kriptografinių valiutų biržų nulaužimai, gavybos apgaulės, kriptografinių valiutų piniginių apgaulės, sukčiavimas apsimetant [5, 40]. Dauguma šių sukčiavimų atvejų nėra nauji, jie tik buvo perkelti į kriptografinių valiutų sistemą.

Aukšto pajamingumo investavimo programos yra internetinės Ponzi schemos, kurios moka nepaprastai aukštas palūkanas (gali siekti net 1–2 % grąžą per dieną) [41]. Šios programos investuotojus pritraukia siūlydamos aukštą pajamingumą ir gali žadėtų palūkanų ir investuotų lėšų negrąžinti. Tačiau kartais būna tokių atvejų, kad pirmiesiems investuotojams yra išmokamos palūkanos ar grąžinamos investuotos lėšos, norint suteikti šiai programai daugiau patikimumo. Ponzi schemos paprastai gyvuoja trumpai, nes yra greitai išaiškinamos. Tačiau jas uždarius tie patys nusikaltėliai sukuria naujas programas vėl siūlančias dideles grąžas.

Kriptografinės valiutos yra gana patogios Ponzi schemų įgyvendinimui, nes atliktų mokėjimų negalima atšaukti, taip pat lėšų gavėjai yra anonimiški, todėl jiems lengviau pasislėpti. Ponzi schemos, pagal padarytus nuostolius, pirmauja tarp visų apgaulių. Vertinama, kad 2019 metais iš visų sukčiavimų padarytų nuostolių, Ponzi schema sudarė 92 % (Ponzi schemų padaryti nuostoliai siekė beveik 4 milijardus dolerių). Iš viso nukentėjo daugiau nei 2,4 milijono investuotojų, kurie vidutiniškai pervesdavo 1 676 dolerių vertės kriptografinių valiutų [5]. Šie skaičiai nestebina, nes sukčiai siūlo aukštas grąžas investuojant į netikras įmones, agresyviai reklamuoja savo produktus socialiniuose tinkluose, sukuria modernius ir patikimai atrodančius projekto tinklapius. Taip pat naudojamos kriptografinės valiutos apsukina sukčių identifikavimą, nes lėšoms surinkti naudojamos keli adresai, pinigams nuslėpti naudojamos tarpiniai adresai.

Šantažuojančių laiškų atveju sukčiai aukai atsiunčia elektroninį laišką, kuriame teigiama, kad buvo įsilaužtą į jų kompiuterį ir pavogta kompromituojanti informaciją. Aukai grasinama, kad jei ji neperves reikiamos pinigų sumos, tai bus paviešinta visa informacija. Dažniausiai įsilaužėliai net neturi jokios medžiagos, tiesiog gąsdina auką. Šantažuojantys laiškai kaip ir brukalai yra išsiunčiami daugybei gavėjų. Šantažuojančiuose laiškuose reikalaujama sąlyginai nedidelės sumos už tylėjimą. Vertinama, kad vidutinė šios apgaulės metu išviliojama suma yra apie 306 dolerius ir spėjama, kad per 2019 metus nukentėjo panašus skaičius žmonių kaip ir su Ponzi schemomis [5]. Didelis mastas ir baimės jausmas leidžia šantažuojantiems laiškams užimti antrą vietą, po Ponzi schemos, pagal padaromą žalą.

Išpirkos reikalaujanti programinė įranga paprastai užšifruoja kompiuterio duomenis ir už jų atšifravimą reikalauja išpirkos. Sumokėjus išpirką yra duodamas raktas, kurį panaudojus galima atšifruoti duomenis. Vertinama, kad iš tokių apgaulių per 2019 metus buvo surinkta 6,6 milijonai dolerių [5]. Tačiau šis skaičius tikriausiai yra per mažas, nes ši apgaules nėra taip gerai matoma ir vieša, kaip Ponzi schemos ar šantažuojantys laiškai.

Kriptografinių valiutų biržų nulaužimas yra visai kitoks, nei prieš tai nagrinėti trys atvejai. Prieš tai nagrinėtos apgaulės yra tiesiogiai nukreiptos į žmones, o čia yra atakuojama kriptografinių valiutų birža, kuri saugo naudotojų kriptografines valiutas. 2019 metais tokių atakų buvo daugiau, nei bet kuriais metais iki šiol – 11. Tačiau žalos apimtimi, 283 milijonai dolerių, atsiliko nuo praeitų metų

žalos – 876 milijonai dolerių [5]. Šių atakų metu yra ieškoma techninių pažeidžiamumų, taip pat pritaikoma socialinė inžinerija ir kiti apgaulės metodai, bet pagrindinis taikinys yra ne pavieniai naudotojai, o birža prekiaujanti kriptografinėmis valiutomis. Nuostoliai šio įsilaužimo atveju būna žymiai didesni. Pavyzdžiui, 2019 metais didžiausia pavogta suma buvo 105 milijonai dolerių iš „CoinBene“ biržos, o 2018 metais įvyko didžiausia iki šiol vagystė, kuri vertinama 534 milijonais dolerių iš „Coincheck“ biržos. Kriptografinių valiutų biržų atakų skaičius nėra didelis, tačiau jomis padaroma piniginė žala yra labai didelė.

Kiti sukčiavimo atvejai savo apimti yra nedideli, lyginant su čia aptartais atvejais. Nors visi sukčiavimo atvejai yra šiek tiek skirtingi, tačiau turi ir panašumų. Juos visus yra gana sudėtinga identifikuoti ir nėra mechanizmo leidžiančio nustatyti ar tai yra apgaulė ar ne. Dažniausiai apie sukčiavimo atvejus yra pranešama tam skirtuose forumuose, kuriuose savo patirtimi pasidalina jau nukentėję asmenys. Taigi nėra kažkokios vieningos sistemos, kuris leistų aptikti sukčiavimą. Taip pat žiūrint iš sandorių pusės, visi šie atvejai turėtų būti panašūs. Nusikaltėliai surenka kriptografinę valiutą į vieną adresą, ir tuomet stengiasi išgryninti pinigus. Aišku sukčiavimo schemos tobulėja (naudojamos keli adresai pinigų surinkimui, prieš išsigryninant pinigus jie pereina keletą adresų), tačiau daugeliu atveju schemos būna panašios. Adresai susiję su apgaulėmis turėtų išsiskirti iš visų kitų adresų, t. y., būti neįprasti ar kitokie. Taigi būtų gerai turėti sistemą, kuri leistų aptikti sukčiavimo atvejus ir apsisaugoti nuo jų.

1.4. Blokų grandinių tyrimai ir sukčiavimo aptikimas

Daugiausia tyrimų, aptinkant sukčiavimo atvejus, yra atlikta naudojant bitkoino kriptografinės valiutos duomenis [42, 43, 44, 45, 46]. Tačiau atsiranda ir tyrimų bandančių nustatyti apgaulės atvejus ir eterio blokų grandininėje [47, 48]. Eterio blokų grandinėse nagrinėjamos tik Ponzi schemos. Identifikuojant Ponzi schemas eterio blokų grandinėje yra sprendžiamas klasifikavimo uždavinys [47, 48], t. y., suformuojamas duomenų rinkinys, kurį sudaro eterio adresai (tai yra bankinės sąskaitos atitikmuo blokų grandinėje), kuriose buvo vykdomos Ponzi schemos ir adresai, kuriose nebuvo vykdomos Ponzi schemos. Taip pat išbandomas išskirčių aptikimo ir dviejų žingsnių metodai [48]. Kaip buvo minėta anksčiau, Ponzi schemos atvejai nustatomi remiantis investuotojų patirtimi, t. y., sukurti specialūs forumai, kuriuose investuotojai pasidalina savo skaudžia patirtimi, ir taip nustatomos Ponzi schemos. Šiuose tyrimuose atrenkama tik 4 000 adresų (iš jų iki 200 turi požymi, kad buvo vykdyta Ponzi schema) ir jas bandoma klasifikuoti ir nustatyti, kurie kriterijai geriausiai nusako Ponzi schemos buvimą. Nagrinėjant bitkoino blokų grandinės duomenis, naudojami įvairesni mokymosi be mokytojo metodai. Naudojant mokymosi be mokytojo metodus yra daromos prielaidos, kad sukčiavimo atvejai bus išskirtys, todėl ieškoma adresų ar sandorių išsiskiriančių iš kitų. Galima sakyti, kad yra naudojami išskirčių nustatymo metodai. Tyrimuose, kuriuose naudojami mokymosi be mokytojo algoritmai yra išnagrinėjamos didesnės duomenų imtys. Tačiau rezultatai nebūna tokie tikslūs. Norint, kad būtų praktinė taikymo nauda, reiktų nagrinėti didesnius duomenų rinkinius, o ne apsiriboti tik 4 000 adresų. Todėl mokymo be mokytojo algoritmai yra praktiškesni.

Beveik visuose tyrimuose, kuriuose buvo naudojamas mokymosi be mokytojo principas, yra naudojamas k-vidurkių metodas [42, 43, 44, 45]. Taip pat išbandomi ir kiti metodai, tačiau be k-vidurkių metodo nėra jokio kito metodo, kuris būtų panaudotas bent dviejuose tyrimuose. Nenuostabu, kad šis metodas yra naudojamas, nes jį nesudėtinga naudoti, galima lengvai modifikuoti, jis pasižymi pakankamai geru greičiu ir gautus rezultatus nesudėtinga interpretuoti.

Todėl tyrimuose k-vidurkių metodas yra laikomas kaip etalonas ir šiuo metodu gauti rezultatai yra lyginami su kitų metodų rezultatais. Nors šis metodas gali suskirstyti duomenų taškus į klasterius, tačiau klasteriai turi būti sferos pavidalo, taip pat šiam metodui trūksta mokėjimo atpažinti išskirtis, kas sukčiavimo atveju yra labai svarbu. Todėl daugumoje tyrimų bandoma pritaikyti ir kitus metodus, tokius kaip Mahalanobio atstumų, mokymosi be mokytojo atraminių vektorių metodą, nupjautų k-vidurkių metodą, tankio ir galios dėsnių (angl. *power degree & desinfication laws*) metodą, vietinių išskirčių faktorių (angl. *local outlier factor*), izoliavimo miško.

4 lentelė. Sukčiavimo aptikimui naudojami požymiai

Požymis	**Pham, Lee (2016) [42]**	**Pham, Lee (2016) [43]**	**Monamo, Marivate, Twala (2016) [44]**	**Zambre, Shah (2013) [45]**	**Bartoletti, Pes, Serusi (2018) [46]**	**Jung et al. (2019) [47]**
Įeinančių sandorių skaičius	+	+	+	+	+	+
Išeinančių sandorių skaičius	+	+	+	+	+	+
Unikalių (iš skirtingų adresų) įeinančių sandorių skaičius	+	+		+	+	+
Unikalių išeinančių sandorių skaičius	+	+		+	+	+
Klasterizavimo koeficientas	+	+	+			
Vidutinė įeinančio sandorio vertė	+	+	+	+	+	+
Vidutinė išeinančio sandorio vertė	+	+	+	+	+	+
Vidutinis laikas tarp išeinančių sandorių	+	+				+
Vidutinis laikas tarp įeinančių sandorių	+	+				+
Balansas	+	+		+		
Sukūrimo data	+	+			+	
Aktyvumo trukmė	+	+			+	+
Naudotojo turimų viešųjų raktų skaičius		+		+		
Visų išeinančių sandorių suma			+		+	+
Visų įeinančių sandorių suma			+		+	+
Išeinančių sandorių verčių standartinis nuokrypis			+	+		+
Įeinančių sandorių verčių standartinis nuokrypis			+	+		+
Trikampių skaičius			+			
4 požymiai nusakantys kaimynus			+			
Įeinančių sandorių dažnis				+		
Išeinančių sandorių dažnis				+		
Vidutinis įeinančių sandorių greitis				+		
Vidutinis išeinančių sandorių greitis				+		
Įeinančių sandorių greičio standartinis nuokrypis				+		
Išeinančių sandorių greičio standartinis nuokrypis				+		
Vidutinis įeinančių sandorių augimo tempas				+		

Požymis	**Pham, Lee (2016) [42]**	**Pham, Lee (2016) [43]**	**Monamo, Marivate, Twala (2016) [44]**	**Zambre, Shah (2013) [45]**	**Bartoletti, Pes, Serusi (2018) [46]**	**Jung et al. (2019) [47]**
Vidutinis išeinančių sandorių augimo greitis				+		
Vidutinis kaimynų išeinančių sandorių greitis				+		
Pervestos sumos Gini koeficientas					+	+
Didžiausias sandorių skaičius per dieną					+	
Įeinančių ir išeinančių sandorių santykis					+	
Didžiausias balanso skirtumas tarp dviejų dienų					+	
Adresų skaičius iš kurių gautos lėšos ir tiems patiems adresams pervestos lėšos						+
Požymiai nusakantys išmaniosios sutarties vykdymo kodą						+

Kalbant apie požymių išskyrimą, pirmiausia reikia paminėti kaip buvo paruošiami duomenys. Iš bitkoino kriptografinės valiutos sandorių buvo konstruojami naudotojų grafai [42, 43, 44, 45]. Naudotojas gali turėti vieną ar kelis adresus, bet ryšys tarp naudotojo ir jam visų priklausomų adresų ne visada gali būti nustatytas. Paprasčiausiu atveju galima laikyti, kad kiekvienas adresas yra atskiras naudotojas. Naudotojų grafas yra gana intuityviai suvokiamas: naudotojai yra laikomi grafo viršūnėmis, o sandoriai briaunomis tarp jų. Kai kuriais atvejais papildomai dar buvo konstruojamas ir sandorių grafas [42, 43]. Tokiu atveju sandoriai yra grafo viršūnės, o briaunos yra bitkoino kriptografinė valiutos srautai. Kituose tyrimuose duomenims surinkti buvo parašyti skriptai, kurie duomenis ištraukia iš blokų grandinės [46, 47, 48]. Iš 4 lentelės matome, kad visuose tyrimuose yra naudojamos požymiai, kurie vienaip ar kitaip nusako įeinančių ir išeinančių sandorių kiekį, vidutinį išeinančių ir įeinančių sandorių verčių vidurkį. Taip pat dažnai naudojamas klasterizavimo koeficientas, kuris įvertina jungumą, tarp nagrinėjamų naudotojų kaimynų. Kituose tyrimuose yra pasiūlomi įdomesni požymiai: vidutinis įeinančių sandorių greitis, vidutinis išeinančių sandorių greitis, vidutinis įeinančių sandorių augimo tempas, vidutinis išeinančių sandorių augimo greitis, vidutinis kaimynų išeinančių sandorių greitis [45] arba didžiausias sandorių skaičius per dieną, pervestos sumos Gini koeficientas, didžiausias balanso skirtumas tarp dviejų dienų [46]. Dauguma išskiriamų požymių parinkimą nulemia duomenų reprezentavimo pasirinkimas, t. y., grafas. Eterio tyrimuose yra naudojamos panašūs požymiai. Tačiau be adresą nusakančių požymių, dar yra įtraukiami požymiai, nusakantys kodą esantį išmaniojoje sutartyje [47, 48]. Daugumoje tyrimu nėra pagrindžiama kodėl išskiriami vieni ar kiti požymiai. Zambre'as ir Shah'as [45] iš viso išskyrė 21 požymį ir bandymų metodu atrinko galutinius 6 požymius, su kurių rinkiniu buvo gauti geriausi rezultatai. Taigi visuose tyrimuose branduolį sudaro labai panašūs požymiai, tačiau dalis tyrėjų pasiūlo ir dalį unikalių požymių.

Taip pat reiktų paminėti, kad nemaža dalis tyrimų įsiveda apribojimus. Atraminių vektorių metodas apsimoko labai ilgai ir neturint galimybių išlygiagretinti skaičiavimų, todėl naudojami duomenys buvo apriboti iki 100 000 duomenų taškų [42]. Kitame tyrime buvo naudojama tik 1 000 000 pirmų bitkoino adresų (iš daugiau nei 6 000 000) [44]. Taip pat buvo apribojamas naudojamų požymių

skaičius iki 6 [43]. Atvejuose, kai buvo nagrinėjamos Ponzi schemos iš viso buvo atrenkama tik iki 4 000 adresų [46, 47, 48]. Taigi reikia suprasti, kad tyrimuose dažniausiai nėra išnagrinėjama visa duomenų aibė. Paprastai ji yra apribojama dėl skaičiavimų sudėtingumo.

5 lentelė. Sukčiavimo aptikimo metodų vertinimo kriterijai ir rezultatai

Nuoroda	**Vertinimo kriterijus**	**Rezultatas**
Pham, Lee (2016) [42]	1. Atrastų išskirčių santykinis atstumas nuo centroidų (naudojant k-vidurkių rezultatus kaip atskaitos tašką). Jei šios reikšmės mažos, tai reiškia metodas nėra pakankamai geras. 2. Įtartinų naudotojų palyginimas su įtartinais sandoriais, t. y,. ar įtartinus sandorius atlieka įtartini naudotojai ir atvirkščiai (kuo arčiau 1 tuo geriau) 3. Kiek metodas aptinka vagysčių iš žinomų 30 atvejų	1. Mahalanobis (naudotojų grafui 0,7619; sandorių grafui 0,8277); SVM (naudotojų grafui 0,7192; sandorių grafui 0,8584) 2. Mahalanobis 0,025633; SVM 0,14415 3. Mahalanobis aptiko vieną atvejį iš 30; SVM aptiko taip pat vieną atvejį
Pham, Lee (2016) [43]	1. Atrastų išskirčių santykinis atstumas nuo centroidų (naudojant k-vidurkių rezultatus kaip atskaitos tašką). Jei šios reikšmės mažos, tai reiškia metodas nėra pakankamai geras. 2. Įtartini naudotojų palyginimas su įtartinais sandoriais, t. y., ar įtartinus sandorius atlieka įtartini naudotojai ir atvirkščiai (kuo arčiau 1 tuo geriau) 3. Kiek suranda iš žinomų 30 bitkoino vagysčių	Tankio ir galios dėsnių metodas parodė, kad egzistuoja išskirtys bitkoino tinkle, t. y., egzistuoja tokie naudotojai, kurie vykdo neįprastas veiklas. 1. LOF 0,965 naudotojų grafui ir 0,914 sandorių grafui 2. LOF 0,55 3. LOF aptikto vieną iš 30 atvejų
Monamo, Marivate, Twala (2016) [44]	1. Kiek aptiko iš žinomų 30 bitkoino vagysčių	1. Nupjautų k-vidurkių metodu buvo aptiktos 5 vagystės iš 30
Zambre, Shah (2013) [45]	1. Kiek algoritmas aptiks iš 3 pasirinktų vagystės atvejų. Iš viso 3 naudotojai buvo priskirti blogiems (vagys) ir 628 naudotojai geriems (aukos) ir klasterizavimo algoritmas šiuos atvejus turėjo išskirti į skirtingus klasterius.	1. Naudojant visus požymius nė vienas iš 3 atvejų nebuvo aptiktas. Naudojant tik atrinktus požymius, buvo išskirti 5 klasteriai ir į vieną iš jų pateko 3 blogi naudotojai (tame klasteryje iš viso buvo 124 761 naudotojas), o kitą 628 geri naudotojai (jame buvo iš viso 756 916 naudotojų).
Bartoletti, Pes, Serusi (2018) [46]	1. Kiek atpažino iš 32 Ponzi schemų.	• RIPPPER aptiko 24 iš 32. Klaidingų priėmimų 226. • Bajeso tinklas aptiko 25 iš 32. Klaidingų priėmimų 266. • Atsitiktinis miškas 25 iš 32. Klaidingų priėmimų 70 • Geriausiu atveju pavyko išgauti 31 iš 32 tikslumą, klaidingų priėmimų buvo apie 1 %.
Jung et al. (2019) [47]	Tikslumas (angl. *precision*), jautrumas ir F-įvertis.	1. Naudojant tik adresų požymius • J48 (0,93; 0,87;0,9) • Atsitiktinis miškas (0,98; 0,84;0,91) • SGD (0,98; 0,84;0,91) 2. Naudojant visus požymius • J48 (0,98; 0,97;0,97) • Atsitiktinis miškas (0,93; 0,92;0,93) • SGD (0,99; 0,94;0,96)

Dar vienas svarbus aspektas, kurį reiktų aptarti, tai tyrimuose gautų rezultatų vertinimas. Vertinimo kriterijai ir rezultatai pateikti 5 lentelėje. Visuose tyrumose vienas iš kriterijų yra kiek atvejų pavyko nustatyti iš žinomų vagysčių (apgaulių) atvejų. Šis kriterijus yra aktualiausias ir jo įvertinimas yra aiškiausias, taip pat aiškiai suvokiama jo praktinė nauda. Kiti vertinimo kriterijai [42, 43] daugiau yra tarsi gautų rezultatų validavimas, nes yra lyginami ar naudotojų ir sandorių grafo atveju gaunami tokie patys rezultatai arba k-vidurkių metodo rezultatai lyginami su kitų metodų rezultatais. Kalbant bendrai apie rezultatus tikrai nėra pasiekiama labai gerų rezultatų. Geriausiu atveju pavyko identifikuoti 5 vagystes iš 30 [44]. Čia nekalbama apie Ponzi schemų identifikavimą [46, 47, 48], nes identifikuojant Ponzi schemas buvo sprendžiamas klasifikavimo uždavinys, todėl rezultatų negalima lyginti su kitais tyrimais.

Apibendrinant, galima teigti, kad tokio pobūdžio tyrimuose kaip etalonas yra naudojamas k-vidurkių metodas. Dažniausiai pasirenkami požymiai yra įeinančių ir išeinančių sandorių kiekiai, vidutiniai išeinančių ir įeinančių sandorių verčių vidurkiai, klasterizavimo koeficientai. Tačiau bandoma įvesti ir originalių požymių, tokių kaip vidutinis įeinančių ir išeinančių sandorių greitis, vidutinis įeinančių ir išeinančių sandorių augimo tempas, kurie galėtų padėti geriau identifikuoti apgaulingus sandorius. Viena didžiausių problemų, su kuria susiduria tyrėjai yra didelis duomenų kiekis, todėl jie yra priversti sumažinti duomenų kiekį. Sumažinus duomenų imtį yra prarandama dalis informacijos, ir dėl to gali nukentėki gaunami rezultatai. Tyrimų autoriai pripažįsta, kad apdorojami tik dalis duomenų ir reiktų plėsti apdorojamų duomenų kiekį. Tam siūlomi įvairūs sprendimai – algoritmų išlygiagretinimas [42], naudojamų požymių išskyrimo paspartinimas ir naudojamų metodų pagerinimas ar kitų metodų išbandymas [43, 44, 46, 48]. Taip pat svarstoma ar tą patį metodą būtų galima pritaikyti kitai blokų grandinei [46]. Sukčiavimų aptikimas bitkoino blokų grandinėje nėra ypač geras, nes visuose tyrimuose nepavyko identifikuoti daugiau nei 5 apgaulių (vagysčių) atvejų iš žinomų 30 atvejų, išskyrus klasifikavimo atvejus. Taigi atsižvelgiant į šiuos iki šiol atliktų tyrimų ribotumus aš keliu tokias užduotis: pasiūlyti naują metodiką, kuri, pritaikant didžiųjų duomenų analitikos metodus, įgalintų aptikti sukčiavimus su kriptografinėmis valiutomis; sukurti modelius, leidžiančius aptikti sukčiavimo atvejus bitkoinų blokų grandinėje, ir palyginti modelių veikimą su kitų autorių rezultatais; patikrinti sukurtų modelių pritaikymą eterio kriptografinės valiutos blokų grandinėje, ir palyginti su eterio blokų grandinei sukurtais modeliais.

2. Sukčiavimo aptikimo metodika

Šiame skyriuje yra detaliai aprašomi, kokie duomenų rinkiniai buvo naudojami šiame darbe, kokie požymiai buvo išskiriami, kokie metodai buvo naudojami aptinkant sukčiavimą ir kaip buvo vertinami gauti rezultatai. Tyrimas buvo atliktas naudojant stacionarų kompiuterį su 4 branduolių procesoriumi *Intel Core i7-4770K CPU @ 3.50GHz*, 16 GB operatyviosios atminties ir 4 TB HDD tipo kietuoju disku.

2.1. Tinkamo duomenų rinkinio paieška

Daugumoje tyrimų aptinkančių sukčiavimo atvejus bitkoino blokų grandinėje buvo naudojamas jau iš anksto paruoštas Ilinojaus universiteto kompiuterinė biologijos laboratorijos duomenų rinkinys, kuris apėmė visus sandorius nuo pirminio bloko iki 2013 balandžio 7 d. sukurto 230 686 bloko [42, 43, 44]. Kitam tyrime buvo naudojamas dar trumpesnis laikotarpis iki 2011 liepos 13 d. [45]. Tyrimuose, kuriuose buvo sprendžiami klasifikacijos uždaviniai nustatant Ponzi schemas tiek bitkoino, tiek eterio blokų grandinėje iš viso atsirinkdavo tik kelis tūkstančius sandorių [46, 47, 48]. Šiuose tyrimuose naudojami duomenų rinkiniai ir jų sudarymo metodai netinka, nes apima tik dalį blokų grandinės duomenų.

Antras galimas variantas yra duomenų rinkinių susidarymas savarankiškai. Bitkoino ir eterio blokų gandinės yra atviro kodo, todėl visa blokų informacija yra viešai prieinama. Norint parsisiųsti bitkoino blokų grandinės visus blokus galima naudoti programinę įranga, kuri palaiko bitkoino protokolą. „Bitcoin Core“ yra pirmoji programa palaikantį bitkoino protokolą, kurią išleido Satoshi Nakamoto [49]. Pirminis blokų grandinės duomenų parsiuntimas (iki 2019 metų spalio mėnesio) truko apie 4 paras. Atnaujinant blokų grandinės duomenis yra parsiunčiami blokų informacija nuo paskutinio atnaujinimo, todėl duomenų atnaujinimo procesas yra greitesnis. Visa bitkoino blokų grandinės informacija iki 2020 vasario mėnesio užėmė apie 280 GB. Taip pat yra programinė įranga „Geth“ ar „Parity“, kuri palaiko eterio protokolą ir leidžia parsisiųsti eterio blokų grandinės blokus [50]. Nors eterio blokų grandinė gyvuoja trumpiau nei bitkoino blokų grandinė, tačiau jos blokų struktūra yra sudėtingesnė ir užima daugiau vietos. Todėl eterio visų blokų informacija iki 2020 vasario mėnesio užėmė apie 260 GB, ir tų duomenų parsisiuntimas truko apie savaitę. Tačiau neapdorotų blokų grandinės duomenų naudoti negalima. Duomenys yra saugomi nevienalytėse ir sudėtinguose duomenų struktūrose, kurios buvo aprašytos ankstesniuose skyriuose, todėl tokie duomenys analizei yra netinkami. Čia kyla pagrindinė problema, nes nėra vieningų įrankių skirtų ištraukti duomenis iš blokų grandinės failų [51]. Duomenų iš blokų ištraukimui buvo išbandyti keli atviros programinės įrangos variantai.

„BitcoinDatabaseGenerator“ yra didelio našumo įrankis, kuris naudojamas bitkoino neapdorotiems duomenims perkelti į SQL reliacinę duomenų bazę [52]. Šis įrankis yra tikrai našus ir sparčiai veikia, tačiau paskutinį kartą jis buvo atnaujintas tik 2017 metais. Ir nuo to laiko bitkoino blokų grandinėje atsirado pasikeitimų. Priėjus prie blokų grandinės failo „blk00909.dat“ (kuriama saugomi blokai nuo 471 981 iki 472 117) apdorojimo yra aptinkama nežinoma bloko versija 5 36 870 930 ir programa nulūžta. Kaip buvo minėta anksčiau, gavybos metu buvo pastebėta, kad bloko versiją nurodant atsitiktinį skaičių paspartėja gavybos procesas. Taigi nuo to bloko buvo pradėtos naudoti blokų versijos ne pagal standartą ir ši programinė įranga aptikus nežinomą bloko versiją nustoja apdoroti bloko duomenis. Apie šią klaidą programos kūrėjui buvo pranešta dar 2018 metų pradžioje, tačiau ji nebuvo pataisyta.

Kitas išbandytas variantas yra neapdorotų bloko grandinės duomenų perkėlimas į grafų duomenų bazę. Toks perkėlimas būtų naudingas, nes galima būtų vizualiai pavaizduoti sandorius, taip pat toks duomenų reprezentavimas yra patogus ir naudojamas kituose tyrimuose [42, 43, 44, 45]. Įrankis „bitcoin-to-neo4j“ importuoja neapdorotus bitkoino blokų grandinės duomenis į „Neo4j“ grafų duomenų bazę. Tačiau šis įrankis yra ganėtinai lėtas. *Thinkpad X220* (8 GB RAM, 4x2.60GHz CPU) bandomąjį duomenų rinkinį, kurio dydis yra 50 GB, importavo 2 savaites [53]. Šiame darbe naudojam techninė įranga yra geresnė, tačiau ir reikiamas importuoti duomenų kiekis yra žymiai didesnis – 280 GB. Taigi paleidus šią programą buvo įvertinta, kad visų duomenų perkėlimas į grafų duomenų bazę truks apie 2 mėnesius. Duomenų importavimui reikalingas dažnas skaitymas ir rašymas į kietąjį diską. Todėl norint, kad duomenų perkėlimas vyktų greičiau yra rekomenduojama naudoti SSD. 2017 gegužės 17 dienai iš viso blokų grandinėje buvo 466 874 blokai ir jų neapdoroti duomenys užėmė apie 114GB, o šie duomenys „Neo4j“ duomenų bazėje iš viso užėmė 625 GB [53]. Taigi norint sutalpinti visus dabartinius bitkoino blokų grandinės duomenis reiktų 2 TB dydžio SSD disko, tačiau šio tyrimo metu tokio dydžio disko nebuvo galimybės panaudoti. Su šia problema susiduria daugiau tyrėjų ir galvoja, kaip būtų galima paspartinti procesą. Vienas iš siūlomų variantų pirmiausia iš neapdorotų duomenų sugeneruoti JSON ir CSV failus ir tada juos importuoti į „Neo4j“ duomenų bazę. Bet šiam sprendimui taip pat reikalingas SSD diskas, nes su juo procesas yra 7 kartus spartesnis [54]. „Bitcoin-to-TigerGraph“ programa perkelia bitkoino blokų grandinės duomenis į „TigerGraph“ grafų duomenų bazę. Perkėlimas trunka 2 valandas ir 50 minučių, bet tam duomenų perkėlimui yra naudojamas virtualus kompiuteris su 96 procesoriais ir 768 GB operatyviosios atminties [55]. Su šiame tyrime naudojama programine įranga duomenų perkelti nepavyko, nes po kurio laiko duomenų apdorojimui pritrūksta operatyviosios atminties. Su eterio blokų grandinės duomenis egzistuoja panašios problemos [51]. Taigi savarankiškai pasidaryti duomenų rinkinio nepavyko, dėl turimų ribotų techninių išteklių.

Trečiasis variantas yra panaudoti viešai prieinamus „BigQuery“ duomenų rinkinius. „BigQuery“ yra duomenų sandėlis ir jame talpinamuose viešai prieinamuose duomenų rinkiniuose yra pateikiama informacija apie 8 kriptografines valiutas: bitkoiną, eterį, *bitcoin cash, dash, dogecoin, ethereum classic, litecoin* ir *zcash.* Visi duomenų rinkiniai atnaujinami kas 24 valandas [56]. Tačiau norint analizuoti ir saugot išanalizuotus duomenis yra taikomi mokesčiai. Pavyzdžiui, 1 TB duomenų apdorojimas kainuoja 5 dolerius, 1 GB duomenų laikymas 0,02 dolerio per mėnesį [57]. Tačiau „Google“ suteikia 300 dolerių kreditą ir leidžia juos išnaudoti „Google Cloud“ paslaugoms per 12 mėnesių. Įvertinus, kad šių kreditų turėtų užtekti duomenų apdorojimui buvo nuspręsta, kad šis variantas yra tinkamiausias šiam tyrimui.

„BigQuery“ duomenų sandėlyje yra *crypto_bitcoin*[1] duomenų rinkinys, kurį sudaro dvi duomenų lentelės *blocks* ir *transactions*. Šiose lentelėse yra talpinami duomenys iš bitkoino blokų grandinės, kurie jau yra apdoroti ir paruošti analizei. Šių lentelių struktūra (1 priedas ir 2 priedas) yra labai panaši literatūros apžvalgoje aprašytam bitkoino bloko ir sandorio sandarai. Tačiau yra keletas skirtumų:

- pridėta keletas laukų, kurie atsirado naujesnėse bitkoino protokolo versijose, po *SegWit* proceso atsiradimo (*weight, stripped_size, virtual_size*);
- prie bloko struktūros nėra saugomos nuorodos į prieš tai einantį bloką;

[1] https://console.cloud.google.com/bigquery?p=bigquery-public-data&d=crypto_bitcoin&page=dataset

- prie sandorio duomenų yra pridėta bloko informacija (*version, block_hash, block_number, block_timestamp, block_timestamp_month*);
- pridėti papildomi išskaičiuojami laukai (*timestamp_month, coinbase_param, input_value, output_value, is_coinbase, fee*).

Šie visi skirtumai nėra esminiai, tiesiog jie yra padaryti norint supaprastinti ir palengvinti duomenų analizę. Iki 2020 metų balandžio 26 d. bitkoino blokų grandinėje iš viso buvo 627 638 blokai ir atlikti 524 140 159 sandoriai, o ši informacija „BigQuery“ duomenų saugykloje atitinkamai užėmė apie 191 MB ir 1,09 TB.

„BigQuery“ duomenų sandėlyje yra *crypto_ethereum*[1] duomenų rinkinys, kurį sudaro aštuonios duomenų lentelės *balances*, *blocks*, *contracts*, *logs*, *token_transfers*, *tokens*, *traces* ir *transactions*. Kaip buvo minėta eterio blokų grandinė yra universalesnė, todėl ir visai informacijai išsaugoti reikia daugiau duomenų lentelių. *Balances* lentelėje yra saugomi visų adresų (sąskaitų) likučiai; *blocks* lentelėje yra saugoma informacija apie eterio blokų grandinės blokus; *contracts* – išmaniosios sutartys; *logs* – visi išmaniųjų sandorių įvykiai; *token_transfers* – saugomas pogrupis sandorių, kuriems reikalaujama ERC20 sutartis; *tokens* – žetonų duomenys; *traces* – sandorių pėdsakai gauti naudojant „Parity“ pėdsakų modulį; *transactions* – visų sandorių esančių bloke duomenys. 3, 4 ir 5 priede yra parašytos atitinkamai *blocks, transactions* ir *traces* lentelių struktūros. Jos yra panašios į eterio bloko sandarą aprašytą literatūros apžvalgoje. Yra keletas skirtumų, kaip ir bitkoino atveju: keletas laukų turi kitokius pavadinimus (pvz.: *benificiery* laukas bloke vadinamas *miner*), pridėti apskaičiuojami laukai (pvz.: *from_address* sandoryje yra nustatytas siuntėjas), prie sandorių pridėta papildoma informacija iš bloko (pvz.: bloko laiko žyma, bloko numeri). Tačiau šie skirtumai nėra esminiai, o skirti palengvinti atliekamas analizes. Iki 2020 metų balandžio 26 d. eterio blokų grandinėje iš viso buvo 9 944 730 blokai ir atlikti 690 244 510 sandorių, o ši informacija „BigQuery“ duomenų saugykloje atitinkamai užėmė apie 10,2 GB ir 307 GB. *Transactions* lentoje yra saugoma informacija tik apie paprastus eterio pavedimus, o *traces* lentoje yra detalesnė informacija, kuria paima ir vidinius sandorius, sutarčių informaciją ir kt., šioje lentoje iš viso iki 2020 metų balandžio 26 d. buvo įrašai apie 1 753 477 515 įvykius ir jie užėmė apie 1 TB.

„BigQuery“ duomenų saugykloje galima patogiai pasiekti bitkoino ir eterio kriptografinių valiutų duomenis. Nors eterio ir bitkoino pateikiamų duomenų struktūros yra skirtingos, tačiau jos turi visus duomenis susijusius su sandoriais. Taigi toliau šiame darbe bus naudojami duomenys iš „BigQuery“ duomenų saugyklos.

2.2. Modeliui naudojamų požymių išskyrimas

Kiekviena kriptografinės valiutos sandoris yra atliekama iš adreso. Adresas kriptografinės valiutos blokų grandinėje yra kaip bankinės sąskaita. Daugumoje tyrimų nagrinėjančių sukčiavimo atvejus, požymiai yra išskiriami adresams ir nustatinėjama, kuris adresas pasižymi išskirtiniais požymiais. Šiame darbe bus daroma taip pat. Parenkant požymius yra atsižvelgiama į prieš tai atliktus tyrimus [42, 43, 44, 45, 46, 47, 48, 56, 58], tik atsisakoma požymių, kurie yra paprasčiau paskaičiuojami turint duomenis reprezentuotus grafu (pvz.: klasterizavimo koeficientas), taip pat atsižvelgiama, kad požymius būtų įmanoma paskaičiuoti tiek bitkoino, tiek eterio blokų grandinės sandoriams.

[1] https://console.cloud.google.com/bigquery?p=bigquery-public-data&d=crypto_ethereum&page=dataset

Žemiau pateikiami išskirti požymiai ir jų apskaičiavimas:

- Kada buvo padarytas pirmas pavedimas (*min_sent_date*). Išreiškiamas sekundėmis nuo 1970-01-01 00:00:00 UTC. Apskaičiuojamas kaip anksčiausia bloko sukūrimo data, kuriame adresas yra prie siuntėjų.
- Kada buvo padarytas paskutinis pavedimas (*max_sent_date*). Išreiškiamas sekundėmis nuo 1970-01-01 00:00:00 UTC. Apskaičiuojamas kaip vėliausia bloko sukūrimo data, kuriame adresas yra prie siuntėjų.
- Kada buvo gautos pirmos įplaukos (*min_received_date*). Išreiškiamas sekundėmis nuo 1970-01-01 00:00:00 UTC. Apskaičiuojamas kaip anksčiausia bloko sukūrimo data, kuriame adresas yra prie gavėjų.
- Kada buvo gautos paskutinės įplaukos (*max_received_date*). Išreiškiamas sekundėmis nuo 1970-01-01 00:00:00 UTC. Apskaičiuojamas kaip vėliausia bloko sukūrimo data, kuriame adresas yra prie gavėjų.
- Laikas sekundėmis tarp pirmo ir paskutinio pavedimo (*sent_active_time*). Apskaičiuojamas kaip *max_sent_date - min_sent_date*.
- Laikas sekundėmis tarp pirmos ir paskutinės įplaukos (*received_active_time*). Apskaičiuojamas kaip *max_received_date - min_received_date*.
- Laikas sekundėmis tarp paskutinio pavedimo ir paskutinės įplaukos (*sent_received_max_lag*). Apskaičiuojamas kaip *max_sent_date - max_received_date*.
- Laikas sekundėmis tarp pirmo pavedimo ir pirmos įplaukos (*sent_received_min_lag*). Apskaičiuojamas kaip *min_sent_date - min_ received_date*.
- Kiek mėnesių buvo atliekami pavedimai (*sent_active_months*). Apskaičiuojamas kaip mėnesių skaičius, kuriais buvo atliekami pavedimai.
- Kiek iš viso buvo atlikta pavedimų (*total_sent_trn_count*). Apskaičiuojamas kaip atliekamų pavedimų skaičius.
- Kokia visų atliktų pavedimų suma (*total_sent_trn_value*). Apskaičiuojamas kaip atliekamų pavedimų verčių suma.
- Keliems unikaliems adresams daromi pavedimai (*total_sent_unique_address*). Apskaičiuojamas kaip unikalių adresų skaičius, kuriems yra atliekami pavedimai.
- Koks yra visų atliktų pavedimų verčių vidurkis (*mean_sent_trn_value*). Apskaičiuojamas kaip atliekamų pavedimų verčių vidurkis.
- Koks yra visų atliktų pavedimų verčių standartinis nuokrypis (*stddev_sent_trn_value*). Apskaičiuojamas kaip atliekamų pavedimų verčių standartinis nuokrypis.
- Už kokią vidutinę vertę per mėnesį yra padaroma pavedimų (*mean_monthly_sent_value*). Apskaičiuojama kaip *total_sent_trn_value* / *sent_active_months.*
- Kiek vidutiniškai kartų per mėnesį padaroma pavedimų (*mean_monthly_sent_count*). Apskaičiuojama kaip *total_sent_trn_count* / *sent_active_months.*
- Kiek mėnesių buvo gaunamos įplaukos (*received_active_months*). Apskaičiuojamas kaip mėnesių skaičius, kuriais buvo gaunamos įplaukos.
- Kiek iš viso kartų buvo gautos įplaukos (*total_received_trn_count*). Apskaičiuojamas kaip gautų įplaukų skaičius.
- Kokia visų gautų įplaukų suma (*total_received_trn_value*). Apskaičiuojamas kaip gautų įplaukų verčių suma.
- Iš kelių unikalių adresų yra gaunamos įplaukos (*total_received_unique_address*). Apskaičiuojamas kaip unikalių adresų skaičius, iš kurių yra gaunamos įplaukos.

- Koks yra visų gautų įplaukų verčių vidurkis (*mean_received_trn_value*). Apskaičiuojamas kaip gautų įplaukų verčių vidurkis.
- Koks yra visų gautų įplaukų verčių standartinis nuokrypis (*stddev_received_trn_value*). Apskaičiuojamas kaip gautų įplaukų verčių standartinis nuokrypis.
- Už kokią vidutinę vertę per mėnesį yra gaunama įplaukų (*mean_monthly_received_value*). Apskaičiuojama kaip *total_received_trn_value* / *received_active_months.*
- Kiek vidutiniškai kartų per mėnesį gaunama įplaukų (*mean_monthly_received_count*). Apskaičiuojama kaip *total_received_trn_count* / *received_active_months.*
- Adreso likutis (*balance*). Kiek yra kriptografinės valiutos turi adresas.

2.3. Duomenų parengimas

Bitkoino ir eterio blokų grandinės duomenims iš viso buvo apskaičiuoti 25 požymiai. Atliekant skaičiavimus buvo stengiamasi, kad pagal logiką bitkoino ir eterio duomenys būtų kuo artimesni. Viso trūkstamos reikšmės buvo užpildytos nuliais, nes ši reikšmė labiausiai tinka, kur yra skaičiuojami valiutos požymiai, taip pat logiškai galima paaiškinti nulines reikšmes ir laiko požymiuose. Pavyzdžiui, jei nebuvo iš adreso daryti pavedimai ar į jį gautos įmokos, tai nulinė reikšmė kaip tik identifikuotų ir atskirtų tokius atvejus. Bitkoino tinkle pasitaiko nestandartinių sandorių, kurie paprastai yra sukuriami naudojant nestandartinius skriptus užrakinant ar atrakinant sandorius. Nestandartinių sandorių adresai turi požymį *nonstandard* ir jų duomenys yra iškraipyti, todėl tokie adresai buvo pašalinti iš duomenų rinkinio. Taip pat dalis eterio adresų nebuvo įvykdę nė vieno sėkmingo sandorio, todėl jie iš duomenų rinkinio buvo taip pat pašalinti. Kadangi vieni požymiai buvo išreikšti sekundėmis, kiti satošiais ar vėjais ir jų skaičiavimo matai yra gana skirtingi, tai visi požymiai buvo standartizuoti naudojant z-įvertį. Galutinai parengus duomenis bitkoino duomenų rinkinį sudarė 654 480 740 adresas su 25 požymiais, o eterio duomenų rinkinį sudarė 114 492 230 adresų su 25 požymiais.

2.4. Sukčiavimo aptikimui naudojami metodai

Norint aptikti sukčiavimo atvejus reikia surasti adresus, iš kurių yra vykdomi sandoriai susiję su apgaulėmis. Adresų identifikavimas, norint apsisaugoti nuo apgaulių, yra svarbiausias, nes žinant, jog šis adresas yra įtartinas, paskatintų naudotojus atsisakyti pinigų siuntimo tokiems adresams. Taigi norint surasti adresus susijusius su sukčiavimu yra sprendžiamas išskirčių aptikimo uždavinys. Išskirčių nustatymo metodai yra dažnai naudojamas aptinkant sukčiavimą. Šie metodai yra pritaikomi aptinkant kreditinių kortelių [59, 60, 61, 62, 63], draudimo [63, 64, 65], pridėtinės vertės mokesčio [66], akcijų biržos [63] sukčiavimus. Taip pat išskirčių nustatymo metodas yra naudojami aptinkant ir sukčiavimo atvejus blokų grandinėje. Kaip buvo minėta anksčiau, beveik visuose tyrimuose, kuriuose buvo naudojamas mokymosi be mokytojo principas, yra naudojamas k-vidurkių metodas [42, 43, 44, 45]. Šiame darbe šis metodas naudojamas kaip atskaitos taškas lyginant kitus modelius. Taip pat turint modelį sukurtą naudojant k-vidurkių metodą bus galima padaryti detalesnius palyginimus su kituose tyrimuose gautais rezultatais. Antrasis pasirinktas metodas yra izoliavimo miškas. Šis metodas buvo pasirinktas, nes jis gali susidoroti su dideliais duomenų kiekiais [67]. Taip pat, naudojant šiuos metodus, dėl turimų didelių kiekių duomenų yra pritaikomi ansamblio mokymosi principai. Turimi duomenys yra atsitiktinai išskaidomi į smulkesnius duomenų rinkinius ir su jais yra sudaroma daug modelių, kurie vėliau balsuoja priimdami bendrą sprendimą. Be to, k-vidurkių modelių ansamblis jau buvo sėkmingai pritaikytas aptinkant sukčiavimus [68].

2.4.1. K-vidurkių metodas

K-vidurkių metodas yra klasterizavimo algoritmas, kurio tikslas yra padalinti turimus *n* taškų į *k* grupes taip, kad tose grupėse būtų kuo panašesni taškai. Šis metodas nėra skirtas išskirčių nustatymui, tačiau kartais yra naudojamas išskirčių nustatymui. Naudojant k-vidurkių metodą išskirtys gali būti nustatomos dviem būdais – priskiriant ekstremalius (pvz.: mažiausius) klasterius išskirtimis arba iš kiekvieno klasterio atrenkant toliausiai nuo centroido nutolusius taškus ir juos priskiriant išskirtims. Šiame darbe yra naudojamas abu išskirčių nustatymo būdai.

K-vidurkių metodas yra centroidais pagrįstas klasterizavimas. Tarkime turime duomenų rinkinį *D*, kuris yra sudarytas iš *n* objektų esančių Euklido erdvėje. Klasterizavimo metodas objektus esančius duomenų rinkinyje *D* padalina į *k* klasterius C_1, C_2, …, C_k taip, kad $C_i \subset D$ ir $C_i \cap C_j = \emptyset$. Kur $1 \leq i,j \leq k$. Norint įvertinti klasterių kokybę yra naudojamos funkcijos, kurios žiūri kiek objektai viename klasteryje yra panašūs ir, kiek skiriasi nuo objektų kituose klasteriuose. Sudarant klasterius stengiamasi minimizuoti kvadratinę paklaidą tarp centroido c_i ir visų objektų priklausančių klasteriui C_i [69]:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p, c_i)^2$$

K-vidurkių algoritmas yra gana nesudėtingas:

1. Nurodomas klasterių skaičius k.
2. Kiekvienam klasteriui parenkame centroidą (dažniausiai parenkami atsitiktinai).
3. Kiekvienas taškas iš duomenų rinkinio yra priskiriamas artimiausia centroidui ir taip suformuojami klasteriai.
4. Pagal suformuotus klasterius paskaičiuojamas naujas klasterio centroidas.
5. Kartojami 3 ir 4 žingsnis, kol klasteriai nusistovi.

Dėl turimo didelio duomenų kiekio ir ribotų techninių resursų neįmanoma sudaryti modelį naudojant visą duomenų rinkinį. Todėl sudarant modelius, bus naudojamas k-vidurkių modelių ansamblis. Klasterizavimo ansamblių pritaikymas yra nagrinėtas įvairiuose darbuose tiek iš teorinės, tiek iš praktinės pusės ir gauti rezultatai rodo, kad šis pritaikymas gali būti naudingas paspartinant algoritmo veikimą (galima išlygiagretinti skaičiavimus), o kartais net gaunant geresnius rezultatus [70, 71, 72]. Kadangi klasteriai bus naudojami išskirtims nustatyti, tai modelio sudarymas yra paprastesnis, nes nereikia sujungti klasterių iš skirtingų modelių:

1. Duomenys atsitiktinai yra padalinami į *m* dalių.
2. Kiekvienai duomenų daliai *m* yra sukuriamas k-vidurkių modelis ir nustatomi klasteriai arba taškai toliausiai nutolę nuo centroido, kurie yra priskiriami išskirtims.

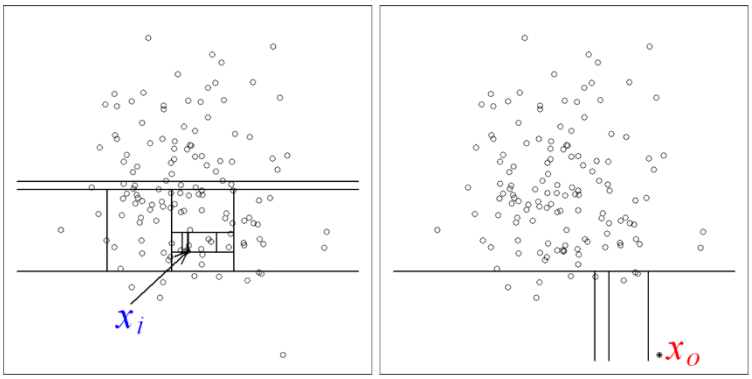
Norint nustatyti ar adresas yra išskirtis visi sudaryti *m* modeliai atliks klasterizavimą ir balsuos už adresą. Jei tyrimo metu nustatyta balų riba bus peržengta, tuomet adresas bus pripažįstamas išskirtimi.

K-vidurkių metodas veikia gana sparčiai, o viena iš sparčiausių realizaciją turi Python biblioteka *SKLearn* [73]. Ši biblioteka bus naudojama kuriant k-vidurkių modelius. Taip pat norint palyginti ar k-vidurkių modelių ansamblis veikia geriau negu vienas k-vidurkių metodas, k-vidurkių metodui

sudaryti bus naudojamas „BigQuery ML“ produktas, kuris leidžia naudotojams kurti ir vykdyti kompiuterinio mokimosi modelius „BigQuery“ naudojant standartines SQL užklausas. „BigQuery ML“ buvo naudojamas kuriant tik pavienius modelius palyginimui, nes jis yra per brangus norint atlikti detalesnę analizę (modelio kūrimo kaštai yra 250 dolerių už 1 TB duomenų). Pavyzdžiui, norint nustatyti optimalų klasterių skaičių būtų reikėję kurti 20 modelių su skirtingais klasterių skaičiais.

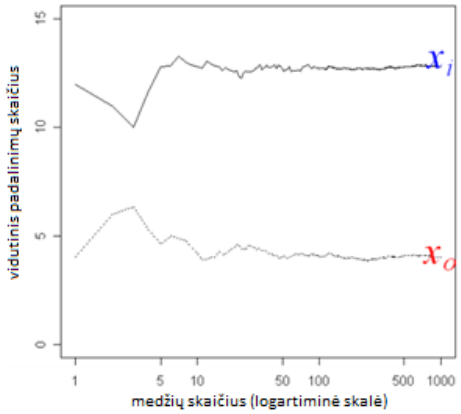
2.4.2. Izoliavimo miškas

Domingues ir kiti [67] savo tyrime atlikto išskirčių metodų vertinimą. Jie vertino įvairius metodus su skirtingais duomenų rinkiniais. Tyrime buvo naudojama 15 skirtingų realių duomenų rinkinių, kurie turėjo iki 107 savybių. Norint patikrinti, kaip metodai veikia su didelių matmenų duomenimis buvo sugeneruoti atsitiktiniai duomenų rinkiniai turintys iki 10 milijonų įrašų ir iki 10 000 savybių. Mokymosi ir išskirčių nustatymui buvo uždėtas 24 valandų limitas ir naudotas kompiuteris su 10 branduolių procesoriumi ir 256 GB operatyviosios atminties. Tyrime buvo naudoti viešai prienami algoritmai ir didžioji dalis jų buvo realizuoti Python kalba (taip pat buvo kelios realizacijos R ir Matlab kalba). Iš visų metodų autoriai išskyrė izoliavimo mišką, kuris yra puikus metodas efektyviai identifikuoti išskirtis, taip pat jis puikiai susidoroja su dideliais duomenų rinkiniais ir efektyviai naudoja atmintį. Taip pat ir kituose darbuose yra nagrinėjamos izoliavimo miško pritaikymo galimybės aptinkant sukčiavimo atvejus blokų grandinėje [48, 74]. Todėl šis metodas gali duoti gerų rezultatų nustatant sukčiavimą blokų grandinėje.



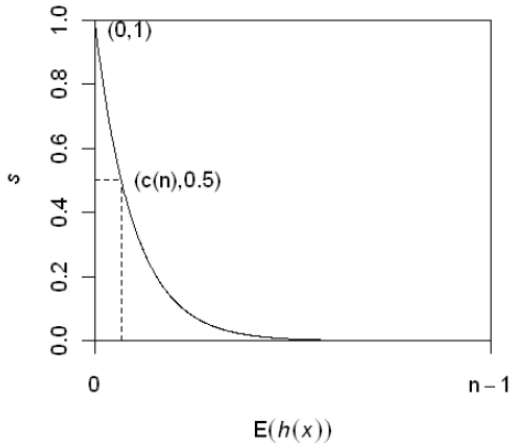
6 pav. Taško x_o ir x_i izoliavimas [75]

Izoliavimo miško metodą 2008 metais pristatė Liu, Ting, ir Zhou [75]. Šis metodas vertina kaip lengvą izoliuoti vieną duomenų tašką nuo kitų duomenų. Jei tą padaryti yra nesudėtinga, tai tikėtina, kad tas taškas yra išskirtis. 6 paveikslėlyje pavaizduota taškai išsibarstę pagal normalųjį skirtinį. Kaip matome izoliuoti tašką x_i prireikė iš viso 12 atsitiktinių padalinimų, o taškui x_o izoliuoti prireikė tik 4 atsitiktinių padalinimų. Toks atsitiktinis erdvės padalinimas yra ne kas kita kaip dvejetainis medis, kuris padalina visą erdvę į dvi dalis. Kaip matome 7 paveikslėlyje yra nagrinėjami tie patys x_i ir x_o taškai ir žiūrima kiek vidutiniškai prireiks padalinimų (arba koks bus kelio ilgis nuo medžio viršūnės iki to taško), kai bandysime daugiau kartų izoliuoti tašką (didinsime naudojamų medžių skaičių). Kaip matome didėjant medžių skaičiui, vidutinis kelio ilgis konverguoja į tam tikrą skaičių. Šiuo atveju x_i į 12,82, o x_o į 4,02. Taigi kuo lengviau yra atskirti stebėjimą, tuo didesnė tikimybė, kad jis bus išskirtis. Šiuo atveju gauti rezultatai tą ir patvirtina. Nes žiūrint į sklaidos diagramą matosi, kad x_o yra labiau nutolęs nuo kitų taškų ir tikėtina, kad jis yra išskirties taškas.



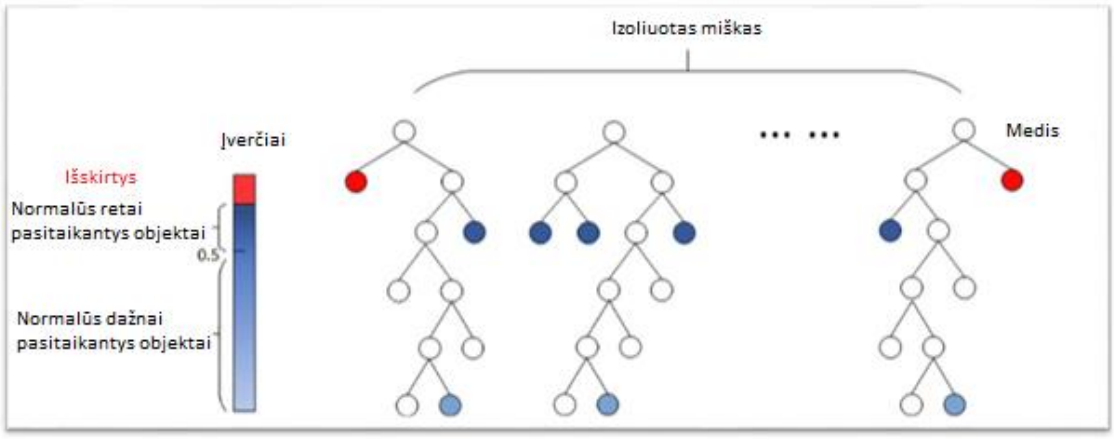
7 pav. Taško x_o ir x_i izoliavimui reikalingų padalinimų skaičius nuo medžių kiekio [75]

Toliau nagrinėjant šį metodą reiktų apsibrėžti sąvokas. Liu, Ting, ir Zhou [75] izoliavimo medį apibrėžia kaip dvejetainį medį, kurio kiekvienas mazgas gali neturėti nė vieno arba turėti du pomazgius. Jeigu neturime sutampančių duomenų, tai sudarytame medyje, kiekvienas duomenų taškas atsidurs kažkuriame medžio lape. Tačiau efektyvumo sumetimais gali būti apribotas medžio aukštis, tuomet į lapus gali patekti keltas duomenų taškų. Kelio ilgis yra žymimas *h(x)* ir rodo kiek medžio briaunų turime praeiti nuo viršūnės, kad pasiektume tašką *x*. Tam, kad nustatytume ar taškas yra anomalija reikia turėti įvertį. Anomalijos įvertis apskaičiuojamas taip: $s(x,n) = 2^{-\frac{E(h(x))}{c(n)}}$. Čia x yra taškas, kurio anomalijos įverti skaičiuojame, *n* rodo, kiek iš viso duomenų taškų turime. $c(n) = 2H(n-1) - \frac{2(n-1)}{n}$ parodo, koks yra vidutinis kelio ilgis medyje iš *n* taškų. Čia *H(i)* yra harmoninis skaičius $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} = \sum_{k=1}^{n} \frac{k}{n}$, jis gali būti įvertintas naudojant Eulerio konstantą $H(i) = \ln(i) + 0{,}5772156649$. $E(h(x))$ yra $h(x)$ surinktų iš skirtingų izoliavimo medžių vidurkis. 8 paveiksle pavaizduotas $E\big(h(x)\big)$ sąryšis su $s(x,n)$. Kai $E\big(h(x)\big) \to c(n)$, tai $s \to 0{,}5$; kai $E\big(h(x)\big) \to 0$, tai $s \to 1$; Kai $E\big(h(x)\big) \to n-1$, tai $s \to 0$.



8 pav. *E(h(x))* sąryšis su anomalijos įverčiu *s(x,n)* [75]

Labai gerai savo darbe Chen ir kt. [76] iliustruoja anomalijos įverčius (9 pav.). Kuo taškas yra aukščiau izoliavimo medyje tuo jo anomalijos įvertis bus arčiau 1. Ir atvirkščiai, kuo žemiau izoliavimo medyje yra taškas, tuo jo anomalijos įvertis bus arčiau 0.



9 pav. Anomalijos įvertis [76]

Atliktas empirinis vertinimas rodo, kad izoliavimo mišku gauti rezultatai yra geresni nei ORCA, LOF ar atsitiktinio miško metodais, vertinant AUC ir vykdymo laiką, ypač dideliuose duomenų rinkiniuose [75]. Turint didelių matmenų duomenis, kuriuose didelė dalis yra netinkamų savybių, izoliavimo miškas gali taip pat veikti puikiai su papildomu savybių parinkimu. Jam taip pat kaip ir k-vidurkių modeliui bus sudaromi ansambliai tam, kad pavyktų apdoroti visus duomenis. *SKLearn* biblioteka turi ir šio metodo realizaciją. Ji toliau ir bus naudojama šiame darbe.

2.5. Modelių rezultatų vertinimo principų aprašymas

Kadangi ieškant išskirčių yra taikomi mokymosi be mokytojo metodai, tai modelių įsivertinimui bus naudojami tokie pat principai kaip panašaus pobūdžio tyrimuose [42, 43, 44, 45]. Yra sudaryti adresų sąrašai, kurie yra susiję su sukčiavimais. Bus lyginama kiek modeliai iš šių adresų identifikavo kaip išskirtis.

Modelių patikrinimui ir įvertinimui bus naudojami šie adresų rinkiniai:

- Žinomų 30 atvejų susijusių su bitkoino vagystėmis ir apgaulėmis aprašytų „BitcoinTalk“ forume (https://bitcointalk.org/index.php?topic=576337) [42, 43, 44].
- Viešai prieinamas Ponzi schemų sąrašas bitkoino blokų grandinėje (http://goo.gl/ToCho7) [46]. Jį sudaro 52 adresai, kurie yra susiję su Ponzi schemų įgyvendinimu.
- Viešai prieinamas Ponzi schemų sąrašas eterio blokų grandinėje (http://goo.gl/CvdxBp) [47, 77]. Jį sudaro 184 adresai, kurie yra susiję su Ponzi schemų įgyvendinimu.
- „CryptoScamDB“ surinkti duomenys apie kriptografinių valiutų sukčiavimus ir apgaudinėjimus (https://github.com/CryptoScamDB/blacklist). Balandžio 22 dieną ją sudarė 2 982 eterio kriptografinės valiutos adresai ir 838 bitkoino kriptografinės valiutos adresai susiję su sukčiavimais ir apgaulėmis.

„BitcoinTalk“ duomenų rinkinys yra dažniausiai naudojamas tyrimuose ir jis apima 30 žinomų bitkoino apgaulės atvejų. Jis yra sudarytas „BitcoinTalk“ forumo bendruomenės ir surinktas į vieną sąrašą, kuris apima apgaules įvykdytas iki 2014 metų. Vėliau šis sąrašas buvo nustotas atnaujinti. Taip pat įdomu, kad šiame sąraše yra daugiau nei 30 apgaulių, tačiau darbuose yra naudojamas skaičius 30. Kai kuriuose darbuose aprašant šį duomenų rinkinį paminima, kad yra apie 30 apgaulės atvejų [42, 43], bet vertinant rezultatus kalbama apie tikslų 30 apgaulių atvejų kiekį. Taip atsitinka todėl, kad ne visos apgaulės yra identifikuotos. Kai kurios apgaulės yra žinomos, tačiau su jomis susiję adresai ir sandoriai nėra nustatyti. Šis duomenų rinkinys daugiausiai yra susijęs su

vagystėmis, kai yra įsilaužiama į serverius ar kompiuterius, taip pat pasinaudojama programinės įrangos spragomis, ir taip pasisavinama informacija apie pinigines ir raktus. Tuomet pasinaudojus šiais duomenimis yra atliekami pavedimai į sukčiams priklausančius adresus.

Viešai prieinami Ponzi schemų duomenų rinkiniai taip pat yra sudaryti remiantis forumų pagalba. Sudarant šiuos sąrašus, rankiniu būdu, buvo su bitkoinu susijusiuose forumuose ieškoma informacijos apie aukšto pajamingumo investicijų programas. Po šiomis programomis dažniausiai slepiasi Ponzi schemos. Tuomet buvo tikrinami internetinių puslapių archyvai ir ieškoma adresų su kuriais būtų siejamos šios Ponzi schemos [46, 77]. Šis duomenų rinkinys apima Ponzi schemas, kurios buvo nustatytos iki 2018 metų.

„CryptoScamDB“ yra atvirojo kodo duomenų bazė, kuri stebi kenksmingus internetinius adresus ir su jais susijusius kriptografinės valiutos adresus. Šis duomenų rinkinys yra didžiausias ir jis yra nuolatos atnaujinimas. Tačiau jis daugiausiai apima smulkius sukčiavimus (angl. *phishing*).

2.6. Naudojama programinė įranga

Šiame darbe duomenų apdorojimui ir k-vidurkių modelio kūrimui buvo naudojami šie „Google Cloud Platform“ produktai:

- „BigQuery“ buvo naudojamas duomenų saugojimui, požymių išskyrimui, požymių standartizavimui, skaičiuojant rodiklius žvalgomajai analizei.
- „BigQuery ML“ buvo naudojamas kuriant k-vidurkių modelius.
- „Cloud Storage“ buvo naudojamas paruoštų duomenų parsisiuntimui į kompiuterį.

Prieš pasirenkant duomenų apdorojimui „Google Cloud Platform“ buvo išbandyta „Bitcoin Core“ ir „Geth“ programinė įranga, kuri leidžia parsisiųsti bitkoino ir eterio blokų grandinės visus blokus. Blokams konvertuoti į analizei tinkamą formatą buvo naudojamos atviro kodo programos:

- „BitcoinDatabaseGenerator“ blokų grandinės blokus perkelia į „Microsoft SQL Server“ reliacinę duomenų bazę.
- „bitcoin-to-neo4j“ blokų grandinės blokus perkelia į „Neo4j“ grafų duomenų bazę.
- „Bitcoin-to-TigerGraph“ blokų grandinės duomenis perkelia į „TigerGraph“ grafų duomenų bazę

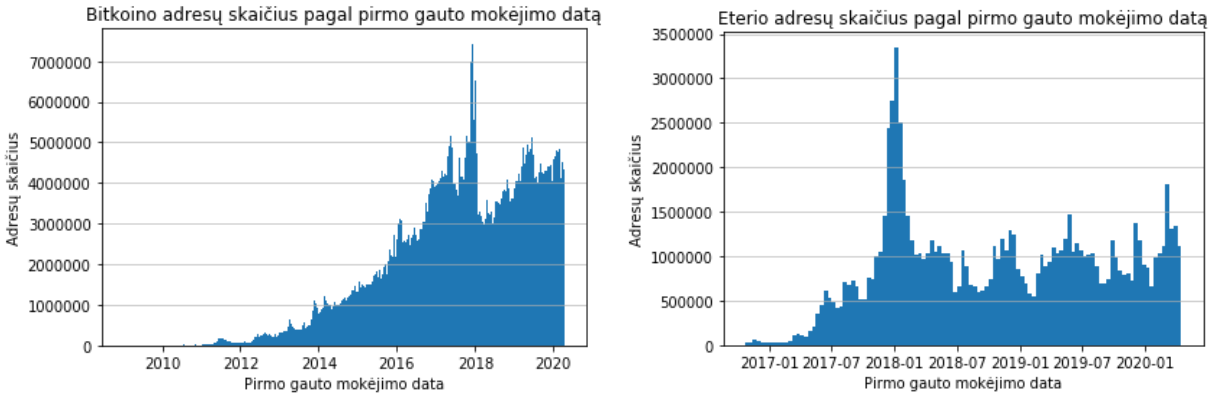
Kuriant k-vidurkių ir izoliavimo miško modelių ansamblius, taip pat apdorojant duomenis bei juos atvaizduojant buvo naudojamos šios Python bibliotekos:

- *SKLearn* – biblioteka skirta mašininiam mokymuisi.
- *Pandas* – biblioteka skirta duomenų analizei ir duomenų apdorojimui.
- *Matplotlib, IPyvolume* – bibliotekos skirtos duomenų vizualizavimui.

3. Sukčiavimo aptikimo modeliai ir gautų rezultatų aptarimas

3.1. Duomenų rinkinio žvalgomoji analizė

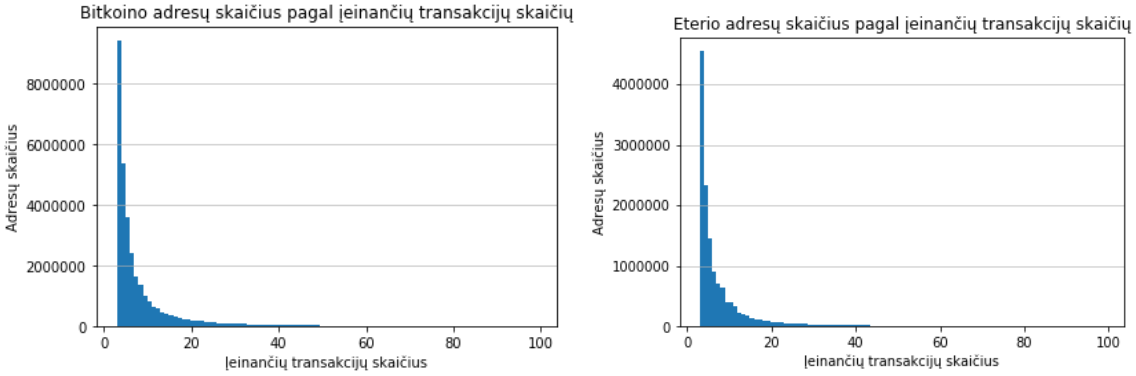
Bitkoino blokų grandinėje iš viso buvo išskirti 654 480 740 adresai, kurie iki 2020 balandžio 26 d. blokų grandinėje atliko sandorius, o eterio blokų grandinėje tokių adresų buvo 114 492 230. Šių adresų pasiskirstymas pagal pirmos įplaukos gavimo laiką yra pavaizduotas 10 paveiksle kairėje pusėje. Adresų skaičius, kurie gavo pirmą mokėjimą, nuo bitkoino atsiradimo stabiliai augo iki 2017 metų, o 2018 metų pradžioje buvo pasiektas pikas. Šis pikas yra dėl to, kad 2017 metų pabaigoje buvo pasiekta iki šiol didžiausia bitkoino vertė. Taip pat žiniasklaida labai daug dėmesio skyrė stipriai augančiai bitkoino kainai, tai ne nuostabu, kad atsirado daug naujų investuotojų norinčių įsigyti bitkoino kriptografinės valiutos. Po piko buvo šioks toks nuosmukis ir po to adresų skaičius grįžo į 2017 metų pradžioje buvusį lygi ir dabar per dieną vidutiniškai atsiranda apie 330 000 adresų, kurie gauna pirmus mokėjimus. Adresų pasiskirstymas pagal paskutinio gauto mokėjimo datą, yra labai panašus į adresų pasiskirstymą pagal pirmo gauto mokėjimo datą. Tai reiškia, kad dauguma adresų tikriausia gauna tik po vieną mokėjimą. Tokie adresai gali būti naudojami investavimui, t. y., gaunami bitkoinai yra laikomi ir neišleidžiami. Kitas galimas variantas, tai adresų vienkartinis naudojimas, t. y., gaunami bitkoinai yra iš karto pervedami į kitą adresą ir šis adresas daugiau nebenaudojamas. Eterio grafikas (10 paveikslėlyje dešinėje) yra rodomas ne nuo pat eterio atsiradimo pradžios, o nuo 2017 metų. 2016 metų pavasarį buvo įvykdyta „DAO“ (Distributed Autonomous Organization) ataka, kurios metu iš sutelktinio finansavimo projekto DAO buvo pavogta daugiau nei 50 milijonų dolerių. Eterio bendruomenė balsavo būdu nusprendė, kad eterio tinklas būtų atstatytas į prieš ataką buvusią būseną. Norint atstatyti būseną buvo naujai sukurti adresai ir į juos grąžintos investuotojų lėšos [78]. Todėl per vieną dieną buvo sukurtą apie 19 milijonų naujų adresų ir jie gavo lėšas, norint eterio tinklą atstatyti į prieš tai buvusią būseną. Toks didelis adresų kiekis vieną dieną iškraipytų bendrą duomenų vaizdą, todėl pateikiami duomenys nuo 2017 metų. Taip pat kaip ir bitkoino atveju, adresų gaunančių pirmą mokėjimą pikas buvo pasiektas 2018 metų pradžioje. Dabar vidutiniškai kasdien atsiranda apie 100 000 eterio adresų, kurie gauna pirmuosius mokėjimus. Kaip ir bitkoino atveju, eterio adresų pagal paskutinį gautą mokėjimą pasiskirstymas yra panašus į adresų pasiskirstymą pagal pirmą gautą mokėjimą. Taigi eteriui galioja tos pačios prielaidos, kaip ir bitkoinui.



10 pav. Adresų skaičius pagal pirmo gauto mokėjimo datą

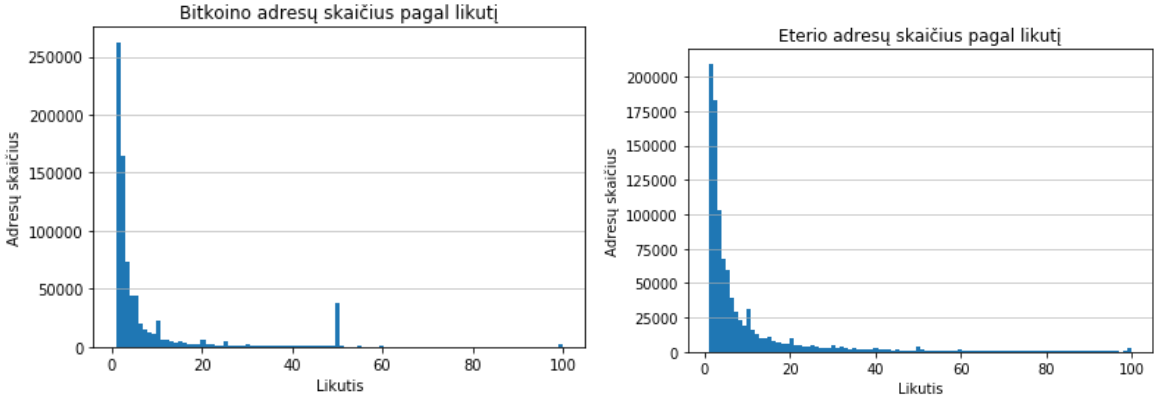
Tiek eterio, tiek bitkoino adresų skaičius pagal pirmo ir paskutinio atlikto mokėjimo datą yra panašūs ir tai patvirtintų prielaidą, kad daug adresų yra naudojami kaip tarpiniai, atliekant vienkartinį pinigų gavimą ir pavedimą. Tą patvirtina ir atliekamų sandorių skaičius. Apie 90 %

adresų bitkoino tinkle turi tik vieną įeinantį ir vieną išeinantį sandorį, o apie 4 % visų adresų turi po du įeinančius ir išeinančius sandorius. Tai rodo, kad norint atlikti pinigų pervedimo operacijas dauguma adresų yra panaudojami tik po vieną kartą. Toks adresų panaudojimas tai pat suteikia ir daugiau anonimiškumo. Su eterio adresais situacija panaši – apie 50 % adresų turi tik vieną įeinantį sandorį, o apie 37 % adresų turi tik po du įeinančius sandorius. Taigi didžioji dalis adresų yra panašūs, nes yra panaudojami tik vieną ar du kartus. 11 paveikslėlyje kairėje yra pateiktas bitkoino adresų, kurie turi du ar daugiau įeinančių sandorių, skaičiaus pasiskirstymas, o dešinėje yra pateiktas analogiškas eterio adresų skaičius. Dar reiktų paminėti, kad iš viso yra beveik 25 milijonai bitkoino adresų, kurie yra tik gavę lėšas, bet nėra padarę nė vieno pavedimo, o eterio tinkle tokių adresų iš viso yra apie 33 milijonus.



11 pav. Adresų skaičius pagal įeinančių sandorių kiekį

Su atliekamų ar gaunamų pavedimų vertės suma vienam adresui yra panaši situacija. Didžiosios dalies adresų (80–90 %) gaunama ir pervedama bendra suma neviršija vieno eterio ar vieno bitkoino. Tai suprantama, nes dauguma adresų atlieka tik po vieną ar du pavedimus ir jų sumos nėra didelės. Tokia pat situacija yra ir su adresuose laikomu kriptografinės valiutos likučiu. Bitkoino atveju 95 % adresų turi likučius lygius nuliui (apie 4,6 % adresų likutis yra mažesnis negu 1 bitkoinas), o eterio atveju – 66 % adresų (33 % adresų likutis yra mažesnis nei 1 eteris). Kas įdomu, kad adresų skaičius turintis 1 ar daugiau bitkoinų ir turintys 1 ar daugiau eterių yra ganėtinai artimas. Bitkoino blokų grandinėje yra apie 0,8 milijono, o eterio blokų grandinėje – apie 1 milijoną adresų, kurie turi daugiau nei vieną vienetą kriptografinės valiutos. Tokių adresų pasiskirstymas pagal sumas pavaizduotas 12 paveikslėlyje. Bitkoino atveju yra nedidelis pakilimas ties 50 bitkonų likučiu, tai yra dėl to, kad pačioje pradžioje už bloko gavybą buvo suteikiama 50 bitkoinų bloko atlygis. Tai bus adresai pirmųjų žmonių užsiėmusių gavybą, kurie gavo šį atlygį ir jo neišleido.



12 pav. Adresų skaičius pagal likutį

Didžioji dalis adresų bitkoino ir eterio blokų grandinėje yra atlikusi tik vieną ar du įeinančius ar išeinančius mokėjimus, ir panašu, kad šie adresai buvo skirti vienkartiniam panaudojimui. Tarp turimų identifikuotų apgaulių atvejų yra tokių adresų, kurie yra padarę po vieną ar du sandorius. Tokie atvejai apima tokias vagystes, kai įsilaužėliams pavyko prisijungti ar kaip kitaip nulaužti savininko piniginę, o iš jos atlikti pervedimą į sau priklausantį adresą, o iš jo išsigryninti pinigus, arba toliau pervedinėti pinigus į kitus adresus, kad būtų sunkiau atsekti vagystę. Tokių apgaulingų sandorių identifikavimui reiktų nagrinėti lėšų keliavimo kelią per adresus ir tam būtų labai naudingas sukonstruotas grafas rodantis lėšų judėjimą. Kadangi šiame darbe dėl resursų trūkumo nepavyko sukonstruoti grafo, tokių sukčiavimo atvejų aptikimas beveik neįmanomas. Atsižvelgiant į tai, kad didžioji dalis adresų yra skirti vienkartiniam panaudojimui, o detalesnių požymių apie tarpusavio adresų ryšius nėra, toliau bus nagrinėjami tik tokie adresai, kurie turi daugiau nei po du įeinančius ir išeinančius sandorius. Pagal šias sąlygas sumažinus adresų kiekį iš viso liko 24 803 831 bitkoino adresas ir 11 412 559 eterio adresas.

3.2. Adresų rinkinių susijusių su apgaulėmis apžvalga

Atitinkamai pagal sumažintą duomenų rinkinį reiktų peržiūrėti ir adresų rinkinį susijusį su apgaulėmis. Iš „BitcoinTalk“ duomenų rinkinio buvo pašalinti bitkoino kriptografinės valiutos netekimai, kai savininkai neturėjo tinkamų piniginių atsarginių kopijų ir neteko piniginės duomenų. Taip pat pašalinti tokie atvejai, kai buvo atliekamas tik vienkartinis pavedimas, t. y., įsilaužėlis gavo priėjimą prie aukos piniginės ir atliko vieną pavedimą iš aukos adreso į savo adresą, o iš jo pinigus išsigrynino arba pervedė į tolimesnius adresus. Tokiems atvejams nustatyti reiktų nagrinėti lėšų keliavimo kelią, o šiame darbe tokios galimybės nėra. Taigi galutiniame „BitcoinTalk“ duomenų rinkinyje iš viso liko 16 bitkoino adresų susijusių su vagystėmis ir apgaulėmis.

Ponzi schemų duomenų rinkiniuose buvo pašalinti adresai, kurie turi mažiau nei du įeinančius ar du išeinančius sandorius. Dėl tokių pakeitimų bitkoino adresų kiekis sumažėjo nuo 52 iki 50, o eterio adresų kiekis nuo 184 iki 102 adresų. Tai rodo, kad eterio tinkle didelė dalis Ponzi schemų nespėjo išplisti ir nuo jų nukentėjusių žmonių nėra arba jų skaičius yra labai mažas.

„CrytoScamDB“ yra didžiausias duomenų rinkinys, tačiau jis yra mažiausiai tvarkingas. Jame yra daug besidubliuojančių adresų, taip pat daug tokių adresų, kurie turi mažiau nei du įeinančias ar du išeinančius sandorius. Taigi sutvarkius šį rinkinį bitkoino adresų kiekis sumažėjo nuo 838 iki 140, o eterio adresų kiekis nuo 2 982 iki 561 adreso.

6 lentelė. Apgaulių rinkinių palyginimas

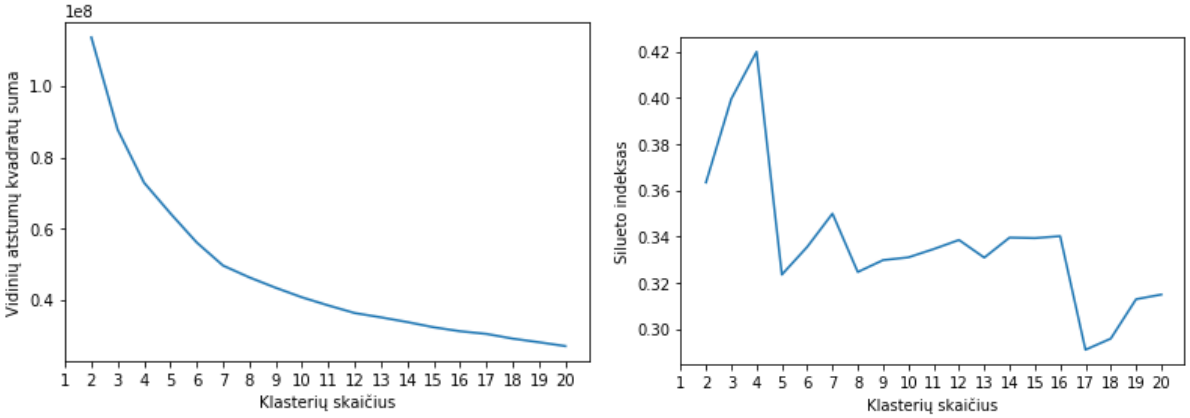
	Vidutinė reikšmė (bitkoinas)			**Vidutinė reikšmė (eteris)**	
Požymiai	**„BitcoinTalk“**	**Ponzi**	**„CryptoScamDB“**	**Ponzi**	**„CryptoScamDB“**
Adresų kiekis	16	50	140	102	561
total_received_trn_count	142	1167	114	149	177
total_sent_trn_count	135	1057	50	129	524
total_received_trn_value	12 164	306	2,5	415	112
total_sent_trn_value	12 164	306	2,5	410	111
balance	0	0	0	5	1

Šie duomenų rinkiniai yra gana skirtingi. „BitcoinTalk“ rinkinį daugiausiai sudaro vagystės atvejai, Ponzi schemų rinkiniai apima apgaules siūlančias aukšto pajamingumo programas, o „CryproScamDB“ daugiausiai orientuojasi į smulkius sukčiavimo atvejus. 6 lentelėje pateiktos kiekvieno apgaulių rinkinio charakteristikos pagal penkis požymius. Vienas bendras dalykas tarp visų rinkinių yra mažas adreso likutis (*balance*), tai rodo, kad sukčiai tuose adresuose nelaiko pinigų, o stengiasi juos kuo greičiau išvesti. „BitcoinTalk“ duomenų rinkinyje esančių sandorių skaičiai (*total_received_trn_count*, *total_sent_trn_count*) nėra dideli, tačiau sumos, kuriomis buvo operuojamos yra didelės (*total_received_trn_value*, *total_sent_trn_value*). Bitkoino tinkle su Ponzi schemos susiję adresai turi didelį kiekį sandorių ir vidutines sumas. Eterio tinkle su Ponzi schemos susiję adresai turi nedidelius sandorių kiekius, bet vidutines sumas, kuriomis operuojama. „CryptoScamDB“ duomenų rinkinyje esantys adresai turi mažas įeinančių ir išeinančių sandorių sumas, taip pat atliekamų sandorių skaičiai nėra dideli. Šie duomenų rinkiniai yra gana skirtingi ir padengia skirtingus apgaulių atvejus. Taip pat, juose nėra nė vieno bendro adreso, kuris pasikartotų bent dviejuose duomenų rinkiniuose.

3.3. K-vidurkių modelių ansamblio pritaikymas bitkoino duomenims

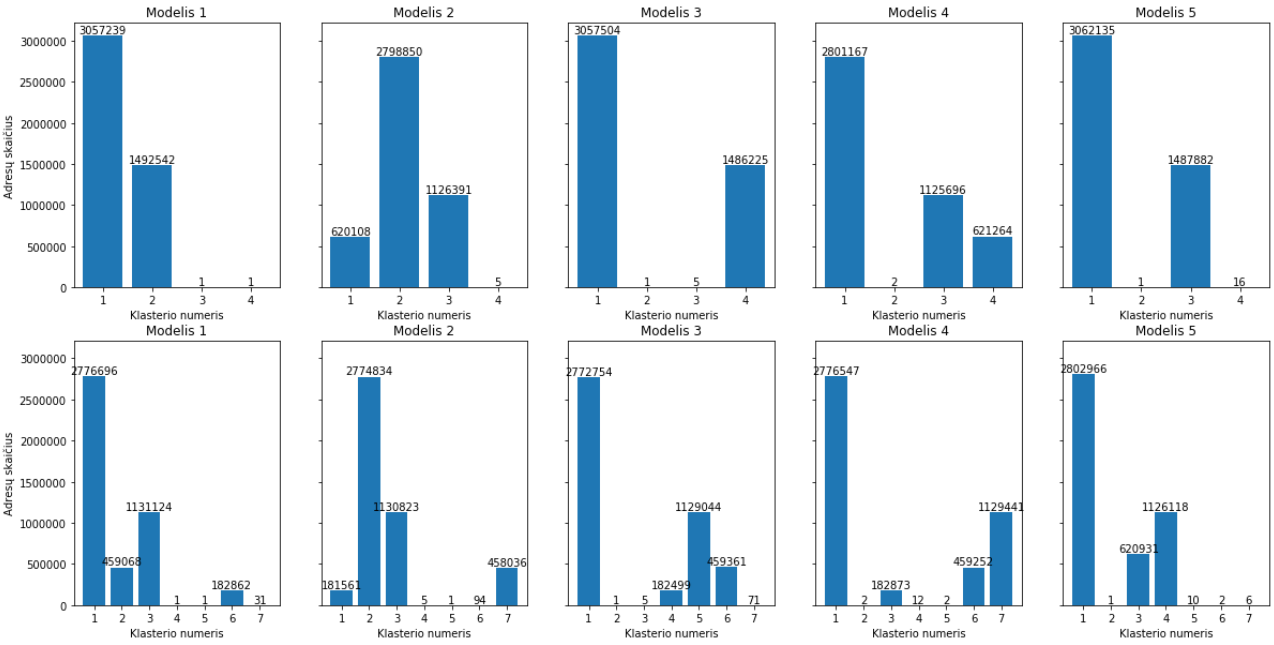
Likęs adresų kiekis, pašalinus adresus turinčius mažiau nei po du įeinančius ar išeinančius sandorius, vis tiek yra per didelis, kad jį būtų įmanoma apdoroti vienu kartu. Bandant apdoroti visą adresų kiekį pritrūksta operatyviosios atminties ir modelio kūrimo procesas yra nutraukiamas. Todėl pirmo modelio sukūrimui buvo naudojamas k-vidurkių modelių ansamblis. Norint, kad balsuojant dėl išskirčių nebūtų neaiškių situacijų, kai vienodas modelių kiekis balsuoja, kad adresas yra išskirtis ir nėra išskirtis, reikia pasirinkti nelyginį modelių kiekį ansamblyje. Padalinus duomenis į tris dalis buvo abejojama, kad užteks resursų tokiam duomenų kiekiui apdoroti, todėl visi turimi adresai buvo atsitiktinai padalinti į panašaus dydžio penkias dalis. Su kiekviena iš šių dalių buvo konstruojamas atskiras modelis ir taip sudaromas modelių ansamblis, kuris bendrai spręs, kurie adresai yra išskirtys.

K-vidurkių modeliui pirmiausia reikia nustatyti klasterių skaičių *k*. Šiam sprendimui priimti buvo naudojama vidinė atstumų nuo centroido kvadratų suma ir silueto (angl. *silhouette*) validavimo indeksas. Su kiekvienu iš penkių duomenų rinkiniu buvo atliekamas klasterizavimas keičiant klasterių skaičių nuo 2 iki 20. Visais atvejais buvo gauti beveik identiški rezultatai. Silueto validavimo indekso skaičiavimas yra labai laikui imlus procesas, todėl silueto validavimo indeksas buvo skaičiuotas su sumažintu duomenų rinkiniu. Kadangi skaičiavimai su skirtingais duomenų rinkiniais duoda labai panašius rezultatus, todėl 13 paveiksle yra pateikti gauti rezultatai su vienu iš duomenų rinkiniu. Vidinė atstumų nuo centroido kvadratų suma (13 paveikslėlis kairėje) pagal alkūnės taisyklę rodo, kad geriausia būrų išskirti 7 klasterius, taip pat būtų galima sudaryti ir 4 klasterius. Silueto validavimo indeksas rodo, kad geriausia yra sudaryti 4 klasterius, bet šis indeksas padidėja ir ties 7 klasteriais. Todėl toliau bus sudaromi du modeliai, vienas naudojant 4 klasterius, o kitas naudojant 7 klasterius. Kituose tyrimuose naudojant k-vidurkių metodą, dažniausiai išskiriami 4, 5, 7 ar 8 klasteriai [42, 43, 44, 45]. Taigi išskiriamų klasterių skaičius yra panašus kaip ir kituose darbuose. Vienas papildomas klasteris kituose darbuose gali atsirasti dėl to, kad šiame darbe yra pašalinti adresai turintys mažiau sandorių, bet šį faktą reiktų dar patikrinti, nes sandorių kiekis yra tik vienas iš požymių lemiančių adresų priskyrimą klasteriams.



13 pav. Klasterių skaičiaus nustatymas bitkoino duomenims

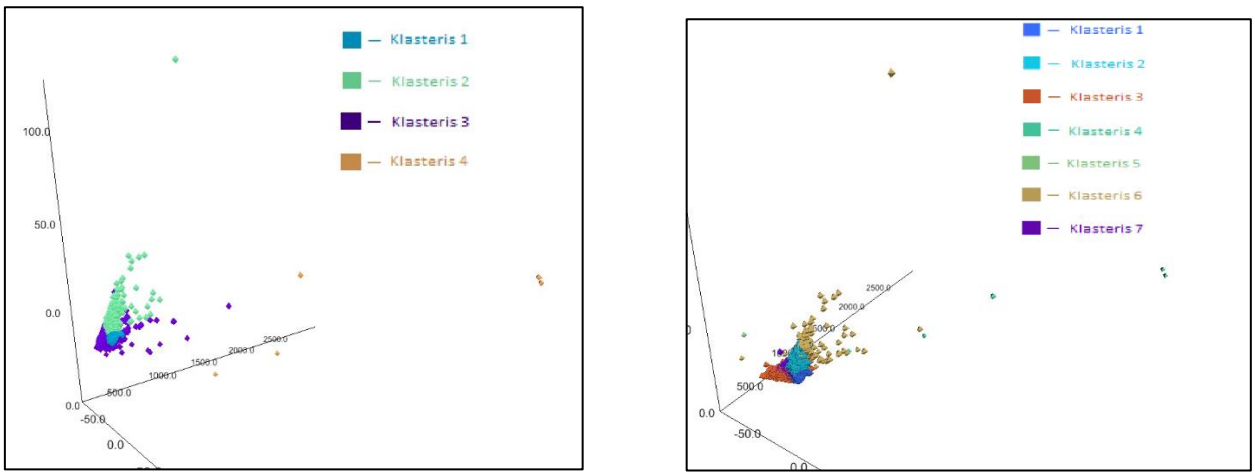
Kiekvienam pasirinktam klasterių skaičiui buvo sukurti penki skirtingi modeliai (kiekvienam atsitiktiniam duomenų rinkiniui po atskirą modelį). Adresų pasidalinimas pagal klasteriuose yra pateiktas 14 paveiksle. Tarp skirtingų modelių pasidalinimas yra gana panašus. Tarp modelių su keturiais klasteriais panašius rezultatus gavo pirmas, antras ir penktas modelis. Juose yra išskiriamas vienas didelis klasteris užimantis apie du trečdalius visų adresų, tuomet antras pagal dydį klasteris užima likusį trečdalį adresų, o likę du mažiausi klasteriai apima pavienius atvejus, kurių kiekis neviršija 16. Antro ir ketvirto modelio pasidalinimas tarp klasterių yra taip pat panašus. Skirtumas toks, kad atsiranda dar vienas didesnis klasteris, kuris apima dalį duomenų iš dviejų didžiausių klasterių, kurie antro ir ketvirto modelio atveju yra šiek mažesni. Septynių klasterių atveju taip pat turime vieną dominuojanti klasterį, kuris apima apie 60 % visų adresų, o likę adresai pasidalina tarp trijų (pirmas, antras, trečias ir ketvirtas modelis) arba dviejų (penktas modelis) klasterių. Kaip ir keturių klasterių atveju, taip ir septynių klasterių atveju, priklausomai nuo modelio yra trys arba keturi klasteriai, kurių adresų kiekis nesiekia 100.



14 pav. Adresų pasidalinimas į klasterius bitkoino duomenims

15 paveiksle yra pavaizduoti antro duomenų rinkinio paskirstymas į klasterius pagal antrą modelį. Požymių suspaudimui yra naudojamos pagrindinės komponentės. Klasteriai yra pateikti pagal

pirmas tris pagrindines komponentes. Klasteriai gražiai neatsiskiria ir yra susispaudę vienoje vietoje, tai gali būti dėl duomenų specifikos, arba dėl to, kad trijų pagrindinių komponenčių yra per mažai norint labiau atskirti tarpusavyje klasterius. 15 paveikslėlio kairėje yra pavaizduotas padalinimas į keturis klasterius, visi trys klasteriai yra gana susispaudę vienoje vietoje, tik ketvirtas klasteris, kuris turi iš viso penkis adresus yra išsibarstęs toliau. Panašiai gaunasi ir su septyniais klasteriais. Didieji klasteriai yra susispaudę vienoje vietoje, o toliau nuo jų yra mažesni ketvirtas, penktas ir šeštas klasteriai. Norint aptikti sukčiavimo atvejus galėtume sakyti, kad adresai patenkantys į mažuosius klasterius ir yra labiausiai tikėtina, kad susiję su apgaulėmis. Tačiau jie nėra susiję su apgaulėmis. Šie adresai yra daugiausiai biržų, kurios užsiima kriptografinių valiutų pardavinėjimu. Pavyzdžiui adresas 19iVyH1qUxgywY8LJSbpV4VavjZmyuEyxV priklauso Singapūre įsikūrusiai biržai „Huobi“, adresas 1Kr6QSydW9bFQG1mXiPNNu6WpJGmUa9i1g priklauso „Bitfinex“ biržai, adresas 17A16QmavnUfCW11DAApiJxp7ARnxN5pGX priklauso „Poloniex“ biržai. Visi šie adresai pakliuvo į ketvirtą klasterį, kuriam priklausė tik 5 adresai. Šie adresai yra išskirtiniai, nes labai aktyviai naudojami ir per juos yra atliekama daug sandorių. Tačiau su apgaulės atvejais jie neturi nieko bendro. Todėl norint aptikti ir apgaulės atvejus reiktų praplėsti ribas ir prie išskirtinių klasterių priskirti ir tuos, kurie apima didesnius kiekius adresų.



15 pav. Bitkoino adresų klasteriai

Taigi, kaip buvo minėta anksčiau, išskirčių aptikimui galima taikyti du skirtingus būdus. Pirmasis būdas yra traktuoti išskirtis kaip elementus priklausančius mažesniems klasteriams. Antrasis, kiekvieno klasterio dalį labiausiai nuo centroido nutolusių taškų laikyti išskirtimis. Toliau tarpusavyje bus palyginti šie abu išskirčių nustatymo būdai. Anksčiau buvo minėta, kad pačių mažiausių klasterių nepakanka identifikuoti apgaulių atvejus, tai todėl prie mažesnių klasterių bus prijungti didesni, kuriuose esantys adresai, pagal jiems apskaičiuotus požymius, bus traktuojami kaip išskirtys. Keturių klasterių atveju išskirtimis bus laikomi trys mažiausi klasteriai, o septynių klasterių atveju – penki mažiausi klasteriai. Antrojo būdo atveju išskirtimis bus laikoma 1 % toliausiai nuo centroido nutolusių adresų [44].

7 lentelėje yra pateiktas išskirčių nustatymo palyginimas bitkoino duomenims su keturių klasterių k-vidurkių metodu, naudojant du skirtingus aukščiau aptartus išskirčių nustatymo metodus. Kadangi pritaikius k-vidurkių metodą buvo gauti arba labai maži, arba ganėtinai dideli klasteriai, tai metodas priskiriant kelis visus klasterius išskirtims, duoda gana didelį išskirčių skaičių. Šie išskirčių skaičiai svyruoja nuo 27 % iki 38 % nuo visų adresų, priklausomai nuo to kokį balų lygį naudosime. Balai parodo kiek modelių iš viso identifikavo, kad adresas yra išskirtis. Skaičius vienas reiškia, kad bent

vienas modelis nurodė, kad adresas yra išskirtis. Skaičius penki reiškia, kad visi penki modeliai identifikavo, kad adresas yra išskirtis.

Modelių rezultatams patikrinti ir palyginti yra naudojami apgaulių duomenų rinkiniai, ir tikrinama, kiek tuose duomenų rinkiniuose esančių adresų buvo priskirta išskirtims. Nors, kai mažiausi klasteriai yra priskiriami išskirtimi pavyko identifikuoti daug apgaulės atvejų („BitcoinTalk“ buvo identifikuota net 15 iš 16 atvejų, Ponzi schemos identifikuotos 23 – 39 iš 50, 21 atvejis iš „CryptoScamDB“), tačiau išskirtims yra priskiriamas labai didelis kiekis visų adresų. Praktikoje šis metodas būtų nelabai naudingas, nes išskirtimis būtų laikoma vos ne pusė visų aktyviau naudojamų adresų. Visai kita situacija yra, kai išskirtims yra priskiriamas 1 % kiekvieno klasterio labiausia nuo centroido nutolusių adresų. Tokiu atveju priklausomai nuo balų skaičiaus išskirtims yra priskiriamas tik 0,2 % – 1,7 % adresų, tačiau ir aptinkamų apgaulių kiekiai yra mažesni. Reikia turėti omeny, kad čia adresų procentas yra skaičiuojamas nuo modeliui sudaryti naudojamų adresų kiekio. Bitkoino atveju tai yra 24 803 831 adresas, nors realiai bitkoino tinkle esančių adresų kiekis beveik 30 kartų didesnis ir siekia 654 480 740. Todėl žiūrint procentą nuo visų adresų sudarančių bitkoino tinklą, šie skaičiai būtų dar mažesni. Naudojamas modelių ansamblis leidžia pasirinkti norimą jautrumą nustatant išskirtis. Norint gauti mažiau adresų galima naudoti didesnį balų skaičių, norint gauti daugiau adresų galima naudoti mažesnį balų skaičių. Tačiau, turint mažiau adresų bus aptinkama mažiau išskirčių. Taigi, atsižvelgiant į šiuos du kriterijus, optimaliausias variantas išskirtims nustatant būtų naudoti 1 % kiekvieno klasterio adresų ir balų skaičių rinktis tarp 3 ir 4. Vienu atveju išskirtims būtų priskirta, tik 0,2 % visų adresų, tačiau identifikuojamas mažesnis skaičius apgaulių. Kitu atveju identifikuojamų apgaulių skaičius išaugtų, tačiau padidėtų ir išskirtims priskiriamų adresų kiekis. Didesnis adresų kiekis yra negerai, nes dalis gerų adresų bus priskirta išskirtims, kas galėtų sulaikyti nuo bendradarbiavimo su šio adreso savininku. Tačiau, šiuo atveju yra svarbiausia saugumas, todėl didesnis adresų kiekis neturėtų būti didelė problemą. Norint būti labiau užtikrintam galima būtų išskirčių nustatymui naudoti ir net dar mažesnį balų skaičių.

7 lentelė. Bitkoino keturių klasterių k-vidurkių modelių ansamblių palyginimas

	Klasteriai priskiriami išskirtims				**1 % kiekvieno klasterio adresų yra išskirtys**			
Balai	**Viso išskirčių (% nuo visų adresų)**	**„Bitcoin Talk“**	**Ponzi**	**„CryptoScamDB“**	**Viso išskirčių(% nuo visų adresų)**	**„Bitcoin Talk“**	**Ponzi**	**„CryptoScamDB“**
1	9444461 (38 %)	15	39	21	422265 (1,7 %)	9	14	14
2	9435110 (38 %)	15	39	21	404442 (1,6 %)	9	13	13
3	7445159 (30 %)	15	23	0	221598 (0,9 %)	7	11	11
4	6739965 (27 %)	15	23	0	47543 (0,2 %)	6	5	5
5	6735965 (27 %)	15	23	0	40999 (0,2 %)	4	5	5

8 lentelėje yra pateiktas analogiškas išskirčių nustatymo metodų palyginimas kaip ir keturių klasterių atveju. Lyginant tarpusavyje metodus gaunami vėl tokie pat rezultatai, kad mažiausius klasterio adresus priskiriant išskirtims, didesnis adresų kiekis priskiriamas išskirtims, bet ir didesnis apgaulių atvejų kiekis identifikuojamas. Tačiau lyginant pagal naudojamą klasterių skaičių jau atsiranda skirtumų. Pirmiausia naudojant septynis klasterius išskirtims yra priskiriamas mažesnis kiekis adresų (12 % – 14 %). Tai atsitinka todėl, kad visus adresus padalinius į daugiau klasterių gaunami mažesni klasteriai, kurie yra priskiriami išskirtims. Tačiau sumažėjus išskirčių skaičiui sumažėja ir apgaulių aptikimo atvejų. Tai galioja ne visais atvejais. Pavyzdžiui, reikalaujant 5 balų

išskirties nustatymui, septynių klasterių atveju yra aptinkamas 21 apgaulė iš „CryptoScamDB“, o keturių klasterių atveju tokių apgaulių neaptinkama. Išskirtims priskiriant mažiausius klasterius, geriau veiktų modeliai su 7 klasteriais, nes adresų priskiriamų išskirtims kiekis stipriai sumažėja, o apgaulių atvejų kiekis taip stipriai nesumažėja (išimtis būtų „BitcoinTalk“ duomenys, kur identifikuojamų adresų kiekis stipriai sumažėja). Priešingi rezultatai gaunami su metodu, kai išskirtims priskiriamas 1 % adresų labiausiai nutolusių nuo centroido. Daugeliu atveju su vienodu ar net didesniu kiekiu adresų priskirtų išskirtims yra aptinkami mažesni kiekiai apgaulių. Septynių klasterių atveju, kai reikalaujama, kad visi modeliai adresą pripažintų išskirtimi, yra išskirtimis nustatoma 0,4 % adresų (8 lentelė), bet aptinkamų apgaulių atvejų kiekiai yra tokie pat ar net mažesni, nei keturių klasterių atveju, kai išskirtimis pripažįstama tik 0,2 % adresų (7 lentelė). Panašios tendencijos matosi ir lyginant kitus rezultatus, kai keturių klasterių modelis, kurie reikalauja trijų balų aptinka daugiau apgaulių nei septynių klasterių modeliai reikalaujantys nuo 1 iki 4 balų, nors išskirtimis laikomų adresų kiekiai yra labai panašūs. Naudojant metodą, kai išskirtims yra priskiriama dalis kiekvieno klasterio toliausiai nuo centroido nutolusių adresų, geresni rezultatai yra gaunami naudojant keturių klasterių modelius. Jei žiūrėtume į rezultatus, kai išskirtims priskiriami mažiausi klasteriai, tai geresni rezultatai būtų septynių klasterių atveju.

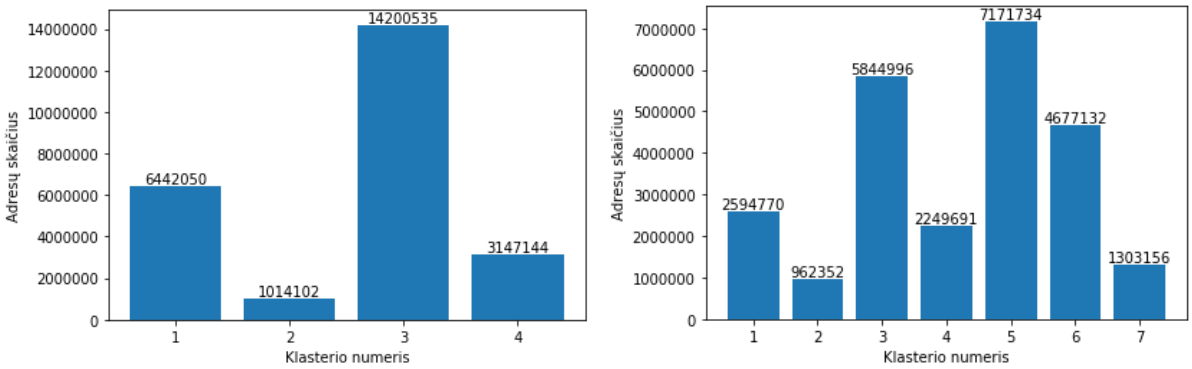
8 lentelė. Bitkoino septynių klasterių k-vidurkių modelių ansamblių palyginimas

	Klasteriai priskiriami išskirtims				**1 % kiekvieno klasterio adresų yra išskirtys**			
Balai	**Viso išskirčių (% nuo visų adresų)**	**„Bitcoin Talk“**	**Ponzi**	**„CryptoScamDB“**	**Viso išskirčių(% nuo visų adresų)**	**„Bitcoin Talk“**	**Ponzi**	**„CryptoScamDB“**
1	3367854 (14 %)	7	20	22	375421 (1,5 %)	7	9	5
2	3209451 (13 %)	6	18	22	233273 (0,9 %)	7	9	4
3	3207460 (13 %)	6	18	21	224655 (0,9 %)	7	8	4
4	3205297 (13 %)	6	18	21	207123 (0,8 %)	7	7	4
5	2943318 (12 %)	4	17	21	96380 (0,4 %)	4	5	2

Šiuo atveju modeliai, kurie išskirtims priskiria visus mažiausių klasterių adresus, būtų nelabai naudingi. Pirmiausia dėl to, kad išskirtims priskiria gana didelę adresų dalį, kuri viršija 10 % ar net 30 %. Nors jie ir aptinka daugiau apgaulės atvejų, tačiau žiūrint vienam adresui susijusiam su apgaule tenka didesnis išskirčių kiekis. Atsižvelgiant į šiuos šio metodo trūkumus, praktikoje būtų geriau naudoti metodą, kai išskirtims yra priskiriama dalis toliausiai nuo centroido nutolusių adresų. Tokiu atveju, daugiau apgaulės atvejų yra aptinkama naudojant keturių klasterių k-vidurkių modelių ansamblį.

3.4. K-vidurkių modelio pritaikymas bitkoino duomenims

K-vidurkių modelių ansamblio taikymas nėra įprastas ir norėtųsi rezultatus pasilyginti su įprastu k-vidurkių metodu, kai visi duomenys yra klasterizuojami iš karto. Turimi techniniai ištekliai to neleidžia padaryti naudojant Python biblioteką *SKLearn*. Todėl k-vidurkių modelis bus sukurtas naudojant „BigQuery ML“, kuris leidžia kurti mašininio mokymosi modelius „BigQuery“ naudojant standartines SQL užklausas. Duomenys buvo klasterizuojami į keturis ir septynis klasterius, kad būtų galima palyginti rezultatus tarpusavyje. Gautas adresų pasidalinimas į klasterius yra pateiktas 16 paveiksle. Gauti klasteriai yra pasiskirstę tolydžiau ir nėra tokių didelių skirtumų tarp klasterių, kaip modeliuose gautuose naudojant *SKLearn* biblioteką.



16 pav. Bitkoino adresų pasidalinimas į klasterius naudojant „BigQuery ML“

9 lentelėje pateikta išskirčių nustatymo metodų palyginimas, kai yra klasterizuojami visi duomenys. Metodas, kai išskirtim priskiriami visas mažiausias klasteris, parodė gana prastus rezultatus. Geresni apgaulių identifikavimo rezultatai yra nebent naudojant septynių klasterių modelį ir išskirtims priskiriant septintą klasterį, kuris yra ne pats mažiausias. Tuomet yra identifikuojama daugiau „BitcoinTalk“ duomenų rinkinyje esančių apgaulių, nei naudojant k-vidurkių modelių ansamblį. Kituose apgaulių rinkiniuose yra identifikuojama mažiau apgaulių. Vena iš pagrindinių prastesnio identifikavimo priežasčių yra dėl mažesnio adresų skaičiaus priskyrimo išskirtims. Naudojant vieną k-vidurkių modelį išskirtims priskiriama du kartus mažiau adresų. Jei iš kiekvieno klasterio atrenkama dalis adresų ir jie priskiriami išskirtims, tuomet modelis su keturiais klasteriais duoda geresnius rezultatus nei modelis su septyniais klasteriais. Tokios pat išvados buvo gautos ir naudojant modelių ansamblį. Lyginant tarpusavyje modelių ansamblį ir vieną modelį sunku padaryti vienareikšmes išvadas. Todėl, kad su vienais duomenų rinkiniais yra gaunami geresni rezultatai su vienu modeliu, o kitais atvejais su modelių ansambliu.

9 lentelė. Bitkoino k-vidurkių modelių palyginimas

	Klasteriai priskiriami išskirtims				**1 % kiekvieno klasterio adresų yra išskirtys**			
Klasteris	**Viso išskirčių (% nuo visų adresų)**	**„Bitcoin Talk“**	**Ponzi**	**„Crypto ScamDB “**	**Viso išskirčių(% nuo visų adresų)**	**„Bitcoin Talk“**	**Ponzi**	**„Crypto ScamD B“**
4 (2)	1014102 (4,1 %)	1	2	0	248 034 (1 %)	10	15	5
7 (2)	962352 (3,9 %)	0	2	0	248 034 (1 %)	10	12	4
7 (7)	1303156 (5,3 %)	9	9	0				

Vertinant bendrus visų duomenų rinkinių rezultatus su septyniais klasteriais gaunami geresni rezultatai naudojant vieną modelį. Taip pat ir lyginant keturių klasterių modelius ir imant panašų išskirčių skaičių (apie 1 % nuo visų adresų), truputį geresnius bendrus rezultatus demonstruoja vienas modelis (aptiko iš viso 30 atvejų), nei modelių ansamblis (aptiko iš viso 29 atvejus). Taigi vertinant bendrai geriausius rezultatus, galima teigti, kad k-vidurkių modelių ansamblis nenusileidžia vienam k-vidurkių modeliui.

3.5. Izoliavimo miško modelių ansamblio pritaikymas bitkoino duomenims

Antrasis metodas naudojamas išskirtims nustatyti yra izoliavimo miškas. Kaip ir sudarant k-vidurkių modelius, taip ir šiuo atveju nepakako resursų, norint sukurti vieną izoliavimo miško modelį su visai duomenis. Todėl buvo naudojami tie patys atsitiktinai paskirstyti penki duomenų rinkiniai ir su jais konstruojamas penkių modelių ansamblis.

10 lentelėje yra patrikti rezultatai gauti naudojant izoliavimo miško modelių ansamblį. Kaip ir k-vidurkių atveju buvo sudaryti penki izoliavimo miško modeliai, kurie atskirai nustatinėdavo išskirtis. Išskirčių nustatymui buvo naudojami keli skirtingi slenksčiai. Naudojant keturių klasterių k-vidurkių modelių ansamblį, kai reikalaujama, kad visi penki modeliai būtų patvirtinę išskirtį gauname, kad išskirtims yra priskiriama 47 543 adresų ir iš viso aptinkama 16 apgaulės atvejų (7 lentelė). Izoliavimo miško atveju yra aptinkama 14 apgaulės atvejų, kai išskirtimis laikoma 44 886 adresai, ir 16 apgaulių, kai išskirtimis laikomi 52 086 adresai. Sumažinus išskirtinių adresų skaičių iki 31 249 aptiktų apgaulių skaičius sumažėja ne taip ženkliai iki 13 (10 lentelė). Taigi k-vidurkių ir izoliavimo miško rezultatai, kai išskirtimis laikoma maža dalis (apie 0,2 %) adresų yra ganėtinai panašūs. Izoliavimo miško modeliai labai gerai veikia aptinkant apgaulės atvejus iš „BitcoinTalk“ duomenų rinkinio. Kituose apgaulių duomenų rinkiniuose, kai yra naudojamas 0,2 % slenkstis, aptinkama mažai arba iš viso neaptinkama apgaulių. Tačiau izoliavimo miške padidinus išskirčių slenkstį iki 1 %, aptinkama daugiau Ponzi schemų ir apgaulių iš „CryptoScamDB“ duomenų rinkinio. Jei išskirtimis laikomas 1 % adresų, tuomet vienas keturių klasterių k-vidurkių modelis aptinka 30 apgaulės atvejų (9 lentelė), keturių klasterių k-vidurkių modelių ansamblis aptinka 29 apgaules (7 lentelė), o izoliavimo miškas aptinka taip pat 29 apgaulės atvejus (10 lentelė). Taigi bendras aptinkamų apgaulių skaičius yra gana panašus. Tačiau izoliavimo miškas stipriai lenkia visus kitus modelius aptinkant „BitcoinTalk“ duomenų rinkinio apgaules (aptinka 15 atvejų lyginant su 10 ar 7 atvejais naudojant k-vidurkių modelius). Tokie ne vienodi apgaulių aptikimo kiekiai skirtingose duomenų rinkiniuose yra duomenų rinkinių specifikos, kurie buvo apžvelgti ankstesniame skyriuje. Kadangi vieni modeliai geriau aptinka apgaules iš vieno duomenų rinkinio, o kiti iš kito, būtų galima apjungti skirtingus modelius į ansamblį ir pasižiūrėti ar jų bendradarbiavimas duotų dar geresnius rezultatus.

10 lentelė. Bitkoino izoliavimo miško modelių ansamblių palyginimas

	Išskirtimis laikoma 0,2 % visų adresų				**Išskirtimis laikoma 1 % visų adresų**			
Balai	**Viso išskirčių (% nuo visų adresų)**	**„Bitcoin Talk“**	**Ponzi**	**„CryptoScamDB“**	**Viso išskirčių(% nuo visų adresų)**	**„Bitcoin Talk“**	**Ponzi**	**„CryptoScamDB“**
1	61210 (0,25 %)	14	3	0	270266 (1,09 %)	15	13	4
2	52086 (0,21 %)	14	2	0	242433 (0,98 %)	15	12	2
3	44886 (0,18 %)	12	2	0	225165 (0,91 %)	15	12	2
4	38557 (0,16 %)	12	1	0	209110 (0,84 %)	15	11	2
5	31249 (0,13 %)	12	1	0	190027 (0,77 %)	15	10	1

Vertinant bendrai rezultatus gautus naudojant izoliavimo miško modelių ansamblį, jei labai neišsiskiria iš kitų rezultatų gautų naudojant k-vidurkių modelius. Tačiau šis modelių ansamblis sugeba labai gerai aptikti apgaulės atvejus iš „BitcoinTalk“ duomenų rinkinio.

3.6. Modelių kurtų su bitkoino duomenimis pritaikymas eterio duomenims

Kitas svarbus klausimas yra ar sukurtas modelis ant vienos kriptografinės valiutos taip pat galėtų sėkmingai identifikuoti neįpratus adresus ir kitoje kriptografinėje valiutoje. Taigi bitkoino kriptografinei valiutai sukurti k-vidurkių modelių ansamblis, k-vidurkių modelis ir izoliavimo miško modelių ansamblis buvo pritaikyti eterio kriptografinės valiutos duomenims. Gauti rezultatai yra pateikti 11 ir 12 lentelėje. Gautų rezultatų palyginti su bitkoino kriptografinės valiutos

duomenimis neišeina, nes yra skirtingi apgaulių duomenų rinkiniai ir apgaulių kiekiai tuose duomenų rinkiniuose. Tačiau lyginant tarpusavyje skirtingus k-vidurkių modelius, gaunamos panašios tendencijos. Priskiriant visą klasterį išimtims gaunama, kad aptinkami didesni kiekiai apgaulių, tačiau ir adresų kiekis priskiriamas išskirtims yra gana didelis. Yra viena išimtis su keturių klasterių k-vidurkių modelių ansambliu, nes keliuose modeliuose į klasterius priskirtus išskirtims patenka sąlyginai mažai adresų. Todėl reikalaujant daugiau balų yra atrenkamas nedideli kiekai adresų, kurie yra laikomi išskirtimis. Be to, šie klasteriai labai gerai identifikuoja Ponzi schemas. Naudojant metodą, kuriame 1 % labiausiai nuo centroido nutolusių taškų yra priskiriama išskirtimi, rezultatai yra prastesni, ir aiškesnio skirtumo, ar keturių ar septynių klasterių vidurkių modelių ansamblis aptinka daugiau apgaulių, nėra. Naudojant vieną k-vidurkių modelį aptikti tiek apgaulių kaip su k-vidurkių modelių ansambliu nepavyksta. Tačiau svarbiausia, kad bitkoino kriptografinės valiutos duomenimis sukurti modeliai leidžia aptikti apgaules ir eterio kriptografinės valiutos tinkle.

11 lentelė. Bitkoino k-vidurkių modelių palyginimas eterio duomenimis

	Klasteriai priskiriami išskirtims			**1 % kiekvieno klasterio adresų yra išskirtys**		
Balai	**Viso išskirčių (% nuo visų adresų)**	**Ponzi**	**„CryptoScamDB“**	**Viso išskirčių(% nuo visų adresų)**	**Ponzi**	**„CryptoScamDB“**
Keturių klasterių k-vidurkių modelių ansamblis						
1	1903328 (16,7 %)	92	29	201160 (1,8 %)	5	9
2	1901850 (16,7 %)	92	29	169630 (1,5 %)	4	7
3	206412 (1,8 %)	56	1	118636 (1,0 %)	2	7
4	178743 (1,6 %)	55	1	54875 (0,5 %)	1	6
5	178028 (1,6 %)	55	1	26320 (0,2 %)	1	5
Septynių klasterių k-vidurkių modelių ansamblis						
1	1817967 (15,9 %)	32	31	175689 (1,5 %)	4	8
2	1671815 (14,6 %)	32	31	121893 (1,1 %)	2	7
3	1669288 (14,6 %)	32	31	113231 (1,0 %)	2	7
4	1667065 (14,6 %)	32	31	107215 (0,9 %)	1	6
5	1560976 (13,7 %)	29	29	52594 (0,5 %)	1	6
Keturių klasterių k-vidurkių modelis						
	212826 (1,9 %)	0	4	114121 (1 %)	0	7
Septynių klasterių k-vidurkių modelis						
	78787 (0,7 %)	14	0	114119 (1 %)	0	8

12 lentelėje yra pateikti rezultatai gauti, taikant bitkoino duomenims sukurtą izoliavimo miško modelių ansamblį, eterio kriptografinės valiutos duomenimis. Kadangi tai yra kitas duomenų rinkinys, tai nėra išlaikomos proporcijos tarp slenksčio naudojamo nustatyti išskirtims. Izoliavimo miško modeliai buvo sukurti taip, kad 0,2 % ar 1 % išskirčių būtų tarp bitkoino kriptografinės valiutos adresų. Tačiau daugiau eterio kriptografinės valiutos adresų atitinka tuos nustatytus kriterijus, dėl to ir gaunama daugiau išskirčių. Tačiau lyginant su k-vidurkių modeliais izoliavimo miškas parodė geresnį apgaulių aptikimo rezultatą, atsižvelgiant į adresų kiekį laikomą išskirtimis. Neatsižvelgiant į keturių klasterių k-vidurkių modelių ansamblį, kuris labai gerai identifikavo Ponzi schemos apgaulių atvejus. K-vidurkių modelių atveju kai išskirtimis laikoma 1,5 % adresų, tai

aptinkama 11 apgaulių (11 lentelė), o izoliavimo miško atveju, kai išskirtimis laikoma 1,6 % adresų iš viso aptinkama 23 apgaulės (12 lentelė).

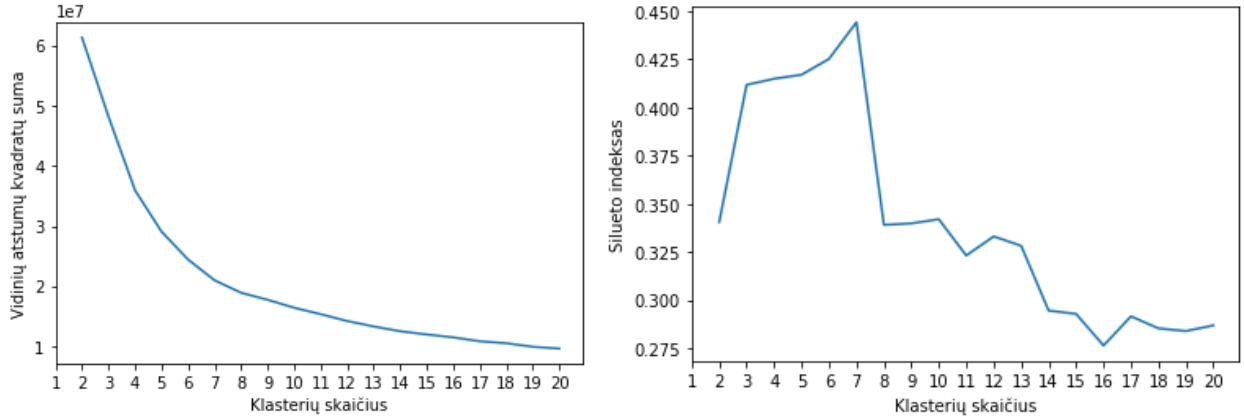
12 lentelė. Bitkoino izoliavimo miško modelių ansamblių palyginimas eterio duomenimis

	Išskirtimis laikoma 0,2 % visų adresų			**Išskirtimis laikoma 1 % visų adresų**		
Balai	**Viso išskirčių (% nuo visų adresų)**	**Ponzi**	**„CryptoScamDB“**	**Viso išskirčių(% nuo visų adresų)**	**Ponzi**	**„CryptoScamDB“**
1	305917 (2,7 %)	14	19	969068 (8,5 %)	28	61
2	275755 (2,4 %)	13	17	884902 (7,8 %)	28	57
3	255996 (2,2 %)	13	16	824625 (7,2 %)	27	56
4	229254 (2,0 %)	12	14	769509 (6,7 %)	25	42
5	188077 (1,6 %)	11	12	676426 (5,9 %)	25	48

Gauti rezultatai rodo, kad modeliai sukurti vienai kriptografinei valiutai gali būti sėkmingai taikomi ir kitai kriptografinei valiutai. Šiuo atveju bitkoino kriptografinei valiutai sukurti modeliai aptiko apgaules ir eterio kriptografinės valiutos tinkle. Geriausiai veikė izoliavimo miško modelių ansamblis.

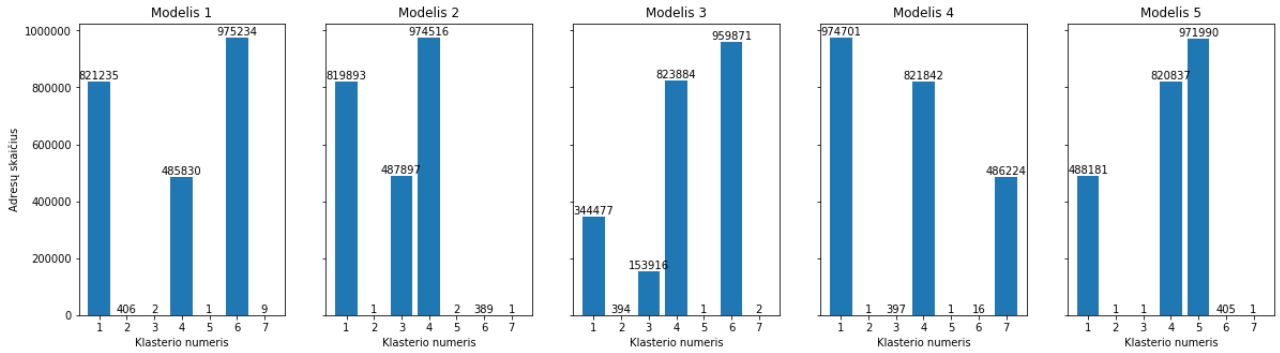
3.7. Modelių sukurtų su skirtingais duomenimis palyginimas eterio blokų grandinėje

Praeitame skyriuje gauti rezultatai rodo, kad vienai kriptografinei valiutai sukurtas modelis gali būti sėkmingai pritaikytas kitai kriptografinei valiutai. Tačiau reikia įsivertinti gautus rezultatus, kiek jie yra geri. Tą galima padaryti sukuriant identiškus modelius su eterio kriptografinės valiutos duomenimis ir pažiūrint kaip jiems seksis aptikti apgaules.



17 pav. Klasterių skaičiaus nustatymas eterio duomenims

Kuriant modelius eterio kriptografinei valiutai buvo atliekami identiški žingsniai, kaip ir kuriant bitkoino kriptografinės valiutos modelius. Pirmiausia buvo nustatyta, kad k-vidurkių metodui tinkamiausias klasterių skaičius yra 7 (17 pav.). Naudojant bitkoino duomenis tinkamiausias klasterių skaičius buvo 4, taip pat dar buvo nagrinėjami 7 klasteriai (13 pav.). Kaip ir bitkoino duomenų atveju (14 pav.), taip ir eterio duomenų atveju (18 pav.) yra trys ar keturi didesni klasteriai, o kiti klasteriai palyginus su didžiaisiais yra ganėtinai maži.



18 pav. Adresų pasidalinimas į klasterius eterio duomenis

13 lentelėje yra pateiktas rezultatų palyginimas tarp dviejų skirtingų išskirčių nustatymo metodų, kai modeliams sudaryti yra naudojamas k-vidurkių metodas. Šių duomenų vienareikšmiškai palyginti su 11 lentelės duomenimis, kurioje sukčiavimo aptikimui buvo naudojami modeliai sukurti pagal bitkoino kriptografinės valiutos duomenis, neišeina.

13 lentelė. Eterio k-vidurkių modelių palyginimas

	Klasteriai priskiriami išskirtims			**1 % kiekvieno klasterio adresų yra išskirtys**		
Balai	**Viso išskirčių (% nuo visų adresų)**	**Ponzi**	**„CryptoScamDB“**	**Viso išskirčių(% nuo visų adresų)**	**Ponzi**	**„CryptoScamDB“**
Septynių klasterių k-vidurkių modelių ansamblis						
1	2611999 (22,9 %)	24	64	143659 (1,3 %)	64	5
2	2442155 (21,4 %)	21	53	123082 (1,1 %)	62	4
3	2438180 (21,4 %)	21	52	115261 (1,0 %)	61	4
4	2433660 (21,3 %)	21	52	110883 (1,0 %)	61	4
5	2317768 (20,3 %)	21	49	77738 (0,7 %)	61	4
Septynių klasterių k-vidurkių modelis						
3	639599 (5,6 %)	0	20	114118 (1,0 %)	17	5
2	911617 (8,0 %)	23	21			

Kai išskirtims priskiriamas 1 % toliausiai nuo centroido nutolusių adresų, geresnius apgaulių aptikimo rezultatus tiek vieno modelio, tiek modelių ansamblio atveju parodo modeliai kurti su eterio duomenimis (11 ir 13 lentelės). Aišku duomenų rinkinyje „CryptoScamDB“ yra nustatomi šiek tiek mažesni apgaulių atvejai, tačiau iš Ponzi schemos duomenų rinkinio žymiai daugiau apgaulių identifikuojama su modeliais kurtais konkrečiai eterio duomenims. Metodo, kai išskirtims priskiriami mažiausi klasteriai, modelių ansamblio atveju geresni rezultatai yra gaunami naudojant modelius sudarytus su bitkoino duomenimis. Šis metodas duoda tokius gerus rezultatus, nes trims modeliams iš ansamblio pavyko labai gera identifikuoti Ponzi schemas ir kas svarbiausia, kad į tuos klasterius pateko sąlyginai mažai adresų. Tai leidžia gana stipriai sumažinti adresų kiekį priskiriamą išskirtims. Žiūrint į bendrus apgaulių aptikimo rezultatus ir atsižvelgiant į adresų kiekį priskiriamą išskirtims, tai geriau atrodo modelių ansamblis kurtas naudojant bitkoino duomenis. Tačiau žiūrint duomenų rinkinius atskirai, daugiau apgaulės atvejų iš „CryptoScamDB“ aptinka modelių ansamblis kurtas su eterio duomenimis. Vieno k-vidurkių modelio duomenų palyginti negalima, nes labai išsiskiria išskirtim priskiriamų adresų kiekiai. Tai iš dalie atsitinka dėl tos pačios priežastie, kuri buvo minėta anksčiau. Į kai kuriuos klasterius, kurie buvo sudaryti naudojant bitkoino

duomenis, patenka mažiau eterio adresų. Todėl išskirtims priskiriamų adresų kiekis naudojant bitkoino duomenims kurtą k-vidurkių modelį šį kartą yra žymiai mažesnis. K-vidurkių modelis kurtas su bitkoino duomenimis atranda mažiau išskirčių, bet ir aptinka mažiau apgaulės atvejų. Taigi vienareikšmiškai pasakyti, kad visais atvejais sukurtas modeli naudojant k-vidurkių metodą duos geresnius rezultatus neišeina. Tačiau jei žiūrėsime į bendrai daugiausiai apgaulių aptinkančius rezultatus, kai nedidelis kiekis adresų yra priskiriamas išskirtims, tai naudojant bitkoino duomenis daugiausia buvo identifikuota 57 adresai, kai išskirtimis buvo laikoma 1,8 % adresų (11 lentelė), o naudojant eterio duomenis pavyko identifikuoti 69 atvejus, kai išskirtimis buvo laikoma 1,3 % adresų. Taigi apibendrinant rezultatus galima sakyti, kad žiūrint bendrus apgaulių aptikimo skaičius ir naudojant modelius sukurtus su tos pačios kriptografinės valiutos duomenis, yra identifikuojama daugiau apgaulių.

14 lentelė. Eterio izoliavimo miško modelių ansamblių palyginimas

	Išskirtimis laikoma 1,0 % visų adresų			**Išskirtimis laikoma 1,5 % visų adresų**		
Balai	**Viso išskirčių (% nuo visų adresų)**	**Ponzi**	**„CryptoScamDB“**	**Viso išskirčių(% nuo visų adresų)**	**Ponzi**	**„CryptoScamDB“**
1	136828 (1,2 %)	17	8	201819 (1,8 %)	22	14
2	123730 (1,1 %)	14	4	183158 (1,6 %)	22	12
3	113376 (1,0 %)	14	4	170633 (1,5 %)	22	11
4	102730 (0,9 %)	13	3	159482 (1,4 %)	20	9
5	93181 (0,8 %)	12	3	141986 (1,2 %)	19	8

14 lentelėje yra pateiktas rezultatų palyginimas tarp dviejų skirtingų išskirčių nustatymo slenksčių, kai modeliams sudaryti yra naudojamas izoliavimo miško modelių ansamblis sukurtas naudojant eterio duomenis. Rezultatų palyginti pagal naudojamus išskirtims nustatyti slenksčius neišeina, nes naudojant bitkoino duomenims sukurtus modelius slenksčiai buvo nustatyti pagal bitkoino duomenis, o eterio atveju gaunasi didesnis adresų kiekis nei nustatytas slenkstis. Todėl palyginimui reikia pasirinkti atvejus kai išskirtimis yra laikomas panašus kiekis adresų. Modelių ansamblis naudojant bitkoino duomenis identifikavo 23 adresus susijusius su apgaulėmis, kai išskirtimis buvo laikoma 1,6 % adresų (12 lentelė), o modelių ansamblis su eterio duomenimis identifikavo 34 adresus, kai išskirtimis buvo laikoma 1,6 % adresų (14 lentelė). Taigi izoliavimo miško modelių ansamblis, kurtas naudojant eterio duomenis, identifikavo daugiau apgaulių.

Nors bitkoino kriptografinei valiutai kurti modeliai gali sėkmingai identifikuoti apgaulės atvejus eterio kriptografinei valiutai, tačiau su eterio duomenimis sukurti modeliai identifikuoja daugiau apgaulės atvejų.

3.8. K-vidurkių ir izoliavimo miško modelių ansamblis

Prieš tai nagrinėjant modelių rezultatus buvo pastebėta, kad k-vidurkių ansamblis geriau aptinka apgaules iš „CryptoScamDB“ duomenų rinkinio, o izoliavimo miško modelių ansamblis iš „BitcoinTalk“ duomenų rinkinio. Gali būti, kad apjungus šiuos du modelius į bendrą ansamblį bus gauti dar geresni apgaulių identifikavimo rezultatai.

15 lentelė. Bitkoino k-vidurkių ir izoliavimo miško modelių ansamblio rezultatai

Balai	Viso išskirčių (% nuo visų adresų)	„BitcoinTalk“	Ponzi	„CryptoScamDB“
1	611341 (2,46 %)	15	16	12
2	598497 (2,41 %)	15	16	11
3	441718 (1,78 %)	15	16	9
4	297781 (1,20 %)	15	13	4
5	296391 (1,19 %)	15	13	4
6	81190 (0,33 %)	9	11	3
7	76211 (0,31 %)	9	10	3
8	50146 (0,20 %)	7	8	2
9	20028 (0,08 %)	6	5	0
10	14874 (0,06 %)	4	5	0

15 lentelėje pateikti išskirčių nustatymo rezultatai, kai ansamblis yra sudarytas iš penkių k-vidurkių modelių ir penkių izoliavimo miško modelių. Daugiau apgaulių aptikti naudojant šį modelį nepavyko. Pagrindinė priežastis, kad izoliavimo miškas ir k-vidurkių metodais nustatytos išskirtys yra gana skirtingos. Naudojant vien tik izoliavimo miško modelių ansamblį iš viso aptikta buvo 32 apgaulės, kai išskirtimis buvo laikoma 1,09 % adresų (10 lentelė). Su naujai sukurtu ansambliu tas pats kiekis aptinkamas, kai išskirtimis laikomas 1,19 % adresų (15 lentelė). Naudojant vien tik k-vidurkių modelių ansambliu yra aptinkamos 37 apgaulės, kai išskirtimis laikoma 1,7 % adresų (7 lentelė). Su naujai sukurtu modelių ansamblių aptinkama 40 apgaulės atvejų, kai išskirtimis laikoma 1,78 % adresų. Taigi tokio didelio pagerėjimo apjungus du skirtingus modelius nesigauna. Šis modelis aptinka daugiau apgaulės atvejų tik tuomet, kai yra didinimas išskirčių kiekis. Taip pat šis ansamblis gali būti naudingas norint sumažinti išskirtimis laikomų adresų kiekį.

3.9. Rezultatų apibendrinimas

Gauti rezultatai rodo, kad šiame darbe pavyko pagerinti apgaulių aptikimą, nes prieš tai atliktuose panašaus pobūdžio tyrimuose buvo aptinkama iki 5 apgaulių iš 30 „BitcoinTalk“ duomenų rinkinyje aprašytų atvejų [42, 43, 44]. Nors šiame tyrime, šis duomenų rinkinys buvo sumažintas iki 16 atvejų, tačiau geriausiu atveju pavykdavo identifikuoti net 15 atvejų iš 16. Tokiam geram rezultatui įtakos turėjo tai, kad buvo nagrinėjamos visi blokų grandinės sandoriai, o ne pasirinkto laikotarpio sandoriai. Taip pat adresų, kuriuose aktyviai nebuvo vykdoma veikla, pašalinimas iš duomenų rinkinio leido sumažinti duomenų kiekį, neprarandant daugelio adresų susijusių su sukčiavimu. Tačiau dalis žinomų apgaulės atvejų pakliuvo tarp pašalintų adresų ir, kaip buvo minėta anksčiau, jiems nustatyti reiktų kurti grafų struktūras, kurios atspindėtų tolimesnį lėšų kelią. Su kitais duomenų rinkiniais tokio įspūdingo aptikimo rezultato pasiekti nepavyko, nors Ponzi schemos duomenų rinkinyje pavyko identifikuoti apie 63 % žinomų apgaulių eterio tinkle (13 lentelė 64 atvejai iš 102). Mažiausiai atvejų buvo identifikuota iš „CryptoScamDB“ duomenų rinkinio, nes jis daugiausiai įtraukė tokius adresus, kurie užsiima mažesnėmis apgaulėmis. Norint aptikti smulkesnes apgaules reiktų bandyti išskirti kitokius požymius arba kuriant modelius naudoti kitokius metodus. Tačiau gauti rezultatai rodo, kad šiame darbe sukurti modeliai leidžia aptikti adresus susijusius su sukčiavimu ir taip apsisaugoti nuo bendradarbiavimo su jų savininkais.

Reikėtų paminėti, kad vieno ar kito modelio taikymas turi savo niuansų. Pavyzdžiui, izoliavimo miško modelių ansamblis gali būti taikomas ir vienam adresui, norint aptikrinti ar jis yra išksirtis. Norint taikyti k-vidurkių metodą, jei išskirtims yra priskiriami mažiausi klasteriai, taip pat galima išskirtis nustatyti vienam adresui. Tačiau jei išskirtims yra priskiriama 1 % toliausiai nuo centroido nutolusių adresų, tuomet išskirtims nustatinėti reiktų turėti didesnį adresų kiekį. Taip pat rezultatai, gauti naudojant šį metodą, gali skirtis priklausomai nuo to koks pradinis duomenų rinkinys yra parenkamas.

16 lentelė. Modelių tarpusavio palyginimas, kai išskirtimis laikomas apie 1 % adresų

Modelis	Aptikta bitkoino apgaulių	Aptikta eterio apgaulių (su bitkoino modeliais)	Aptikta eterio apgaulių
Keturių klasterių k-vidurkių ansamblis	29 (14,1 %)	9 (1,3 %)	-
Septynių klasterių k-vidurkių ansamblis	20 (9,7 %)	7 (1,0 %)	65 (9,6 %)
Vienas keturių klasterių k-vidurkių modelis	30 (14,6 %)	7 (1,0 %)	-
Vienas septynių klasterių k-vidurkių modelis	26 (12,6 %)	14 (2,0 %)	22 (3,3 %)
Izoliavimo miško modelių ansamblis	29 (14,1 %)	23 (3,4 %)[1]	18 (2,7 %)
K-vidurkių ir izoliavimo miško modelių ansamblis	32 (15,5 %)[2]	-	

[1] šiuo atveju išskirčių buvo 1,6 % nuo visų adresų, nes tai mažiausias gautas išskirčių skaičius su šiuo modeli

[2] šiuo atveju išskirčių buvo 1,19 % nuo visų adresų, nes tai mažiausias gautas išskirčių skaičius su šiuo modeliu

16 lentelėje yra pateikti apibendrinti duomenys, kaip pavyko identifikuoti apgaulės atvejus bitkoino ir eterio tinkle. Norint visus modelius palyginti imti atvejai, kai išskirtimis yra laikoma 1 % arba mažiau adresų nuo visų adresų. Keletui modelių nebuvo tokių situacijų, tai jiems parinktas 1 % artimiausias variantas. Iš bitkoino modelių būtų sunku kažkurį vieną modelį išskirti, nes tiek k-vidurkių modelių ansamblis, tiek vienas k-vidurkių modelis, tiek izoliavimo miškas parodė labai panašius rezultatus ir sugebėjo aptikti 29 – 30 apgaulės atvejų, kas sudarytų 14,1 – 14,6 % nuo visų bendro apgaulių skaičiaus duomenų rinkiniuose. K-vidurkių ir izoliavimo miško modelių ansamblis sugebėjo aptikti 32 apgaules, bet jame išskirčių buvo daugiau – 1,19 %. Tokį patį rezultatą sugebėjo pasiekti ir izoliavimo miško modelių ansamblis su mažesne dalimi išskirčių (10 lentelė). Eterio apgaulių aptikimui naudojant bitkoino modelius geriausi rezultatai buvo pasiekti su izoliavimo miško modelių ansambliu, tačiau išskirtimis buvo laikoma 1,6 % adresų. Atsižvelgiant į nustatyta 1 % išskirčių ribą, tai geriausias pasiektas rezultatas buvo su vienu septynių klasterių k-vidurkių modeliu, kuris aptiko 14 apgaulių. Eterio apgaulių aptikimui, naudojant modelius sukurtus su eterio duomenis, geriausi rezultatai buvo gauti su septynių klasterių k-vidurkių modelių ansambliu. Modelis aptikto 65 apgaules, kurios sudaro 9,6 % nuo visų apgaulių esančių duomenų rinkiniuose. Taigi vienareikšmiškai geriausio modelio bitkoino atveju išskirti negalime, nes visi modeliai parodė labai artimus rezultatus. Eterio atveju geriausiai veikė septynių klasterių k-vidurkių modelių ansamblis.

Išvados

1. Atliktus literatūros analizę buvo išsiaiškinta, kad blokų grandinės technologija pritaikoma finansuose, medicinoje, švietimo srityje, chemijos pramonėje ir kt. Tačiau, blokų grandinės technologija labiausiai išvystyta ir daugiausiai naudojama finansų rinkose, ypač kriptografinėms valiutoms. Visgi, esant daug privalumų, su jomis pasitaiko ir tokios nelegalios veiklos, kaip Ponzi schemos, šantažuojantys laiškai, išpirkos reikalaujanti programinė įranga, kriptografinių valiutų biržos nulaužimai, kas užima didžiausią dalį neteisėtos veiklos kriptografinių valiutų blokų grandinėje ir šių sukčiavimų padaryta žala 2019 metais siekė 4,3 milijardo dolerių.
2. Nagrinėjant kitų autorių tyrimus buvo nustatyta, kad bitkoino blokų grandinėje aptinkant sukčiavimus yra naudojamas mokymasis be mokytojo ir dažniausiai pritaikomas k-vidurkių metodas. Sukuriant modelius, jis laikomas tarsi etalonu, su kuriuo yra lyginami kitais modeliais gauti rezultatai. Tačiau, eterio blokų grandinėje apgaulėms identifikuoti naudojamas mokymasis su mokytoju ir dažniausiai modeliams sukurti pritaikomas atsitiktinio miško metodas. Didžiausias iššūkis, su kuriais susiduria tyrėjai, yra didelis duomenų kiekis. Norint sumažinti duomenų kiekį, yra naudojamas trumpesnis laikotarpis ir sumažinamas požymių, naudojamų modeliui kurti, kiekis. Taip pat autoriai mini poreikį išlygiagretinti metodų taikymo ar modelių kūrimo procesą.
3. Atlikus literatūros ir kriptografinės valiutos sandorių duomenų analizę, darbe buvo pasiūlyta metodika, kuri atrenka duomenis ne pagal laikotarpį ar požymių kiekį, o atsisakant adresų (bankinės sąskaitos atitikmuo kriptografinių valiutų blokų grandinėje) turinčių mažiausiai informacijos, t. y., iš duomenų rinkinio pašalinant adresus neturinčius bent trijų įeinančių ir trijų išeinančių sandorių. Taip pat šiame tyrime sukčiavimui aptikti buvo išbandyta viena iš galimybių išlygiagretinti k-vidurkių modelio taikymą – sukuriant k-vidurkių modelių ansamblį. Duomenų atrinkimas ir gauti sukčiavimo aptikimo rezultatai rodo, kad k-vidurkių ansamblis labai panašiai (aptiko 29 atvejus) aptinka apgaules kaip vienas k-vidurkių metodas (aptiko 30 atvejų).
4. Sukūrus modelius ir palyginus jų apgaulių aptikimo rezultatus paaiškėjo, kad bitkoino blokų grandinėje skirtingi modeliai aptinka panašius sukčiavimo atvejų skaičius. Tiek k-vidurkių modelių ansamblis, tiek vienas k-vidurkių modelis, tiek izoliavimo miško modelių ansamblis aptiko 29–30 apgaulių. Eterio blokų grandinėje k-vidurkių modelių ansamblis aptiko 65 apgaules, kai vienas k-vidurkių modelis aptiko 22 apgaules, o izoliavimo miško modelių ansamblis – 18. Su kitų autorių tyrimuose gautais rezultatais galima palyginti tik iš „BitcoinTalk“ duomenų rinkinio aptiktas apgaules, kuomet šiame darbe sukurtas izoliavimo miško modelių ansamblis aptiko 15 apgaulės atvejų, kai su šiuo rinkiniu kitų autorių tyrimuose daugiausiai yra aptinkamos tik 5 apgaulės.
5. Modelius, sukurtus naudojant bitkoino sandorių duomenis, galima sėkmingai pritaikyti aptinkant apgaulės atvejus eterio blokų grandinėje. Tai rodo, kad dalis apgaulių savo prigimtimi yra panašios, neatsižvelgiant, kokioje blokų grandinėje jos yra vykdomos. Tačiau modeliai, sukurti naudojant eterio sandorių duomenis, tos pačios rūšies kriptografinės valiutos grandinėje aptiko 65 apgaules, o modeliai, sukurti naudojant bitkoino sandorių duomenis, eterio blokų grandinėje aptiko tik 23 apgaules.

Literatūros sąrašas

1. POLONIEX. Bitcoin exchange. *Poloniex* [interaktyvus]. N.d. [žiūrėta 2020-05-26]. Prieiga per: https://poloniex.com/exchange#usdc_btc
2. CoinMarketCap. Top 100 Cryptocurrencies by Market Capitalization. *CoinMarketCap* [interaktyvus]. N.d. [žiūrėta 2020-05-26]. Prieiga per: https://coinmarketcap.com/
3. YLI-HUUMO, Jesse, et al. Where is current research on blockchain technology? —a systematic review. *PloS one* [interaktyvus]. 2016, 11(10): e0163477 [žiūrėta 2020-05-26]. doi: 10.1371/journal.pone.0163477
4. TERLATO, Peter. $673 million stolen in crypto hacks in 2018. *Finder* [interaktyvus]. 2018 [žiūrėta 2020-05-26]. Prieiga per: https://www.finder.com/673-million-stolen-in-crypto-hacks-in-2018
5. CHAINANALYSIS. The 2020 state of crypto crime. *Chainanalysis* [interaktyvus]. 2020 [žiūrėta 2020-05-26]. Prieiga per: https://go.chainalysis.com/rs/503-FAP-074/images/2020-Crypto-Crime-Report.pdf
6. UNDERWOOD, Sarah. Blockchain beyond bitcoin. *Communications of the ACM* [interaktyvus]. 2016, 59(11), 15-17 [žiūrėta 2020-05-26]. doi: 10.1145/2994581
7. SWAN, Melanie. *Blockchain: Blueprint for a new economy*. Sebastopol: O'Reilly Media, Inc., 2015. ISBN 9781491920497.
8. UMEH, Jude. Blockchain double bubble or double trouble? *Itnow* [interaktyvus]. 2016, 58(1), 58-61 [žiūrėta 2020-05-26]. doi: 10.1093/itnow/bww026
9. CASEY, Michael, et al. *The impact of blockchain technology on finance: a catalyst for change*. Geneva: International Center for Monetary and Banking Studies (ICMB), 2018. ISBN 9781912179152.
10. AZARIA, Asaph, et al. Medrec: Using blockchain for medical data access and permission management. In: *2016 2nd International Conference on Open and Big Data (OBD)*. IEEE, 2016. pp. 25-30.
11. HOY, Matthew B. An introduction to the blockchain and its implications for libraries and medicine. *Medical reference services quarterly* [interaktyvus]. 2017, 36(3): 273-279 [žiūrėta 2020-05-26]. doi: 10.1080/02763869.2017.1332261
12. METTLER, Matthias. Blockchain technology in healthcare: The revolution starts here. In: *2016 IEEE 18th international conference on e-health networking, applications and services (Healthcom)*. IEEE, 2016. pp. 1-3.
13. ROMAN-BELMONTE, Juan M.; DE LA CORTE-RODRIGUEZ, Hortensia; RODRIGUEZ-MERCHAN, E. Carlos. How blockchain technology can change medicine. *Postgraduate medicine* [interaktyvus]. 2018, 130(4): 420-427 [žiūrėta 2020-05-26]. doi: 10.1080/00325481.2018.1472996
14. ZHANG, Peng, et al. Blockchain technology use cases in healthcare. *Advances in computers. Elsevier* [interaktyvus]. 2018, 111, 1-41 [žiūrėta 2020-05-26]. doi: 10.1016/bs.adcom.2018.03.006
15. SHARPLES, Mike; DOMINGUE, John. The blockchain and kudos: A distributed system for educational record, reputation and reward. In: *European conference on technology enhanced learning*. Springer, Cham, 2016. p. 490-496.

16. GRECH, Alexander; CAMILLERI, Anthony F. *Blockchain in education.* Luxembourg: Publications Office of the European Union, 2017. ISBN 9789279734977.
17. SIKORSKI, Janusz J.; HAUGHTON, Joy; KRAFT, Markus. Blockchain technology in the chemical industry: Machine-to-machine electricity market. *Applied Energy* [interaktyvus]. 2017, 195: 234-246 [žiūrėta 2020-05-26]. doi: 10.1016/j.apenergy.2017.03.039
18. ZYSKIND, Guy, et al. Decentralizing privacy: Using blockchain to protect personal data. In: *2015 IEEE Security and Privacy Workshops*. IEEE, 2015. pp. 180-184.
19. DUJAK, Davor; SAJTER, Domagoj. Blockchain applications in supply chain. In: Kawa A., Maryniak A. (eds) *SMART supply network*. Cham: Springer, 2019, pp. 21-46. ISBN 9783319916675.
20. NAKAMOTO, Satoshi. A peer-to-peer electronic cash system. Bitcoin. *Bitcoin* [interaktyvus]. 2008 [žiūrėta 2020-05-26]. Prieiga per: https://bitcoin.org/bitcoin.pdf
21. ZHENG, Zibin, et al. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services* [interaktyvus]. 2018, 14(4): 352-375 [žiūrėta 2020-05-26]. doi: 10.1504/IJWGS.2018.095647
22. ŽILINSKAS, Evaldas; ALZBUTAS, Robertas. Importance and forecasting of bitcoin energy consumption. *17th International Conference of Young Scientists on Energy Issues, Kaunas, Lietuva, 2020: abstract.*
23. KING, Sunny. Primecoin: Cryptocurrency with prime number proof-of-work. *Primecoin* [interaktyvus]. 2013 [žiūrėta 2020-05-26]. Priega per: https://primecoin.io/bin/primecoin-paper.pdf
24. DZIEMBOWSKI, Stefan, et al. Proofs of space. In: *Annual Cryptology Conference*. Springer, Berlin, Heidelberg, 2015. p. 585-605.
25. VASIN, Pavel. Blackcoin's proof-of-stake protocol v2. *Blackcoin* [interaktyvus]. 2014 [žiūrėta 2020-05-26]. Prieiga per: https://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper.pdf
26. RENNOCK, Michael J.W.; COHN, Alan.; BUTCHER, Jared R. Blockchain technology and regulatory investigations. *The Journal* [interaktyvus]. 2018, 1(7), 35-44 [žiūrėta 2020-05-26]. Prieiga per: https://www.steptoe.com/images/content/1/7/v3/171269/LIT-FebMar18-Feature-Blockchain.pdf
27. LIN, Iuon-Chang; LIAO, Tzu-Chun. A survey of blockchain security issues and challenges. *International Journal of Network Security* [interaktyvus]. 2017, 19(5): 653-659 [žiūrėta 2020-05-26]. doi: 10.6633/IJNS.201709.19(5).01
28. CHOHAN, Usman W. Cryptocurrencies: A brief thematic review. *SSRN* [interaktyvus]. 2017 [žiūrėta 2020-05-26]. doi: 10.2139/ssrn.3024330
29. ANTONOPOULOS, Andreas M. *Mastering bitcoin: Programming the open blockchain*. Sebastopol: O'Reilly Media, Inc., 2014. ISBN 9781491954386.
30. CHUEN, David Lee Kuo (ed.). *Handbook of digital currency: Bitcoin, innovation, financial instruments, and big data*. London: Academic Press, 2015. ISBN 9780128021170.
31. Bitcoinblockhalf. Bitcoin Block Reward Halving Countdown. *Bitcoinblockhalf* [interaktyvus]. N.d. [žiūrėta 2020-05-26]. Prieiga per: https://www.bitcoinblockhalf.com/
32. TAYLOR, Michael Bedford. The evolution of bitcoin hardware. *Computer* [interaktyvus]. 2017, 50(9): 58-66 [žiūrėta 2020-05-26]. doi: 10.1109/MC.2017.3571056

33. Bitcoin. Bitcoin Improvement Proposals. *GitHub* [interaktyvus]. N.d. [žiūrėta 2020-05-26]. Prieiga per: https://github.com/bitcoin/bips
34. CHAKRAVARTY, Manuel MT, et al. The extended UTXO model. In: *4th Workshop on Trusted Smart Contracts*. 2020.
35. ANTONOPOULOS, Andreas M.; WOOD, Gavin. *Mastering ethereum: building smart contracts and dapps*. Sebastopol: O'reilly Media, 2018. ISBN 9781491971949.
36. BUTERIN, Vitalik. A Prehistory of the Ethereum Protocol. *Vitalik Buterin's website* [interaktyvus]. 2017 [žiūrėta 2020-05-26]. Prieiga per: https://vitalik.ca/general/2017/09/14/prehistory.html
37. DANNEN, Chris. *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners*. Berkeley: Apress, 2017. ISBN 9781484225349.
38. Ethereum. The Ethereum Improvement Proposal repository. *Ethereum* [interaktyvus]. N.d. [žiūrėta 2020-05-26]. Prieiga per: https://eips.ethereum.org/
39. WOOD, Gavin, et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper* [interaktyvus]. 2014 [žiūrėta 2020-05-26]. Prieiga per: http://gavwood.com/Paper.pdf
40. VASEK, Marie; MOORE, Tyler. There's no free lunch, even using Bitcoin: Tracking the popularity and profits of virtual currency scams. In: *International conference on financial cryptography and data security*. Springer, Berlin, Heidelberg, 2015. p. 44-61.
41. MOORE, Tyler; HAN, Jie; CLAYTON, Richard. The postmodern Ponzi scheme: Empirical analysis of high-yield investment programs. In: I*nternational Conference on financial cryptography and data security. Springer*. Berlin, Heidelberg, 2012. p. 41-56.
42. PHAM, Thai; LEE, Steven. Anomaly detection in bitcoin network using unsupervised learning methods. *arXiv* [interaktyvus]. 2016 [žiūrėta 2020-05-26]. Prieiga per: https://arxiv.org/abs/1611.03941v2
43. PHAM, Thai; LEE, Steven. Anomaly detection in the bitcoin system-a network perspective. *arXiv* [interaktyvus]. 2016 [žiūrėta 2020-05-26]. Prieiga per: https://arxiv.org/abs/1611.03942
44. MONAMO, Patrick; MARIVATE, Vukosi; TWALA, Bheki. *Unsupervised learning for robust Bitcoin fraud detection*. In: 2016 Information Security for South Africa (ISSA). IEEE, 2016. p. 129-134.
45. ZAMBRE, Deepak; SHAH, Ajey. Analysis of bitcoin network dataset for fraud. *Stanford CS 224W Project Final Report - Group 30* [interaktyvus]. 2013 [žiūrėta 2020-05-26]. Prieiga per: https://pdfs.semanticscholar.org/255d/540d10d710a81fe4aa035d9fd884c7e9c5cb.pdf
46. BARTOLETTI, Massimo; PES, Barbara; SERUSI, Sergio. Data mining for detecting Bitcoin Ponzi schemes. In: *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 2018. p. 75-84.
47. JUNG, Eunjin, et al. Data Mining-based Ethereum Fraud Detection. In: *2019 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2019. p. 266-273.
48. CHEN, Weili, et al. Exploiting blockchain data to detect smart Ponzi schemes on Ethereum. *IEEE Access* [interaktyvus]. 2019, 7: 37575-37586 [žiūrėta 2020-05-26]. doi: 10.1109/ACCESS.2019.2905769
49. BITCOINCORE. About. *BitcoinCore* [interaktyvus]. N.d. [žiūrėta 2020-05-26]. Prieiga per: https://bitcoincore.org/en/about/

50. ETHHUB. Running an Ethereum Node. *EthHub* [interaktyvus]. N.d. [žiūrėta 2020-05-26]. Prieiga per: https://docs.ethhub.io/using-ethereum/running-an-ethereum-node/
51. ZHENG, Peilin; ZHENG, Zibin; DAI, Hong-ning. XBlock-ETH: Extracting and Exploring Blockchain Data From Ethereum. *arXiv* [interaktyvus]. 2019 [žiūrėta 2020-05-26]. Prieiga per: https://arxiv.org/abs/1911.00169
52. LADIMOLNAR. Bitcoin Database Generator. *GitHub* [interaktyvus]. 2017 [žiūrėta 2020-05-26]. Priega per: https://github.com/ladimolnar/BitcoinDatabaseGenerator
53. IN3RSHA. Import the Bitcoin blockchain in to a Neo4j graph database. *GitHub* [interaktyvus]. 2019 [žiūrėta 2020-05-26]. Priega per: https://github.com/in3rsha/bitcoin-to-neo4j
54. PETKANIČ, Peter. *Bitcoin Blockchain Analysis* [interaktyvus]. Brno, 2018 [žiūrėta 2020-05-26]. Prieiga per: https://is.muni.cz/th/v2dsl
55. TIGERGRAPH. Bitcoin to TigerGraph. *GitHub* [interaktyvus]. 2019 [žiūrėta 2020-05-26]. Priega per: https://github.com/tigergraph/bitcoin-to-tigergraph
56. DAY, Allen. Introducing six new cryptocurrencies in BigQuery Public Datasets—and how to analyze them. *Google Cloud* [interaktyvus]. February 5, 2019 [žiūrėta 2020-05-26]. Prieiga per: https://cloud.google.com/blog/products/data-analytics/introducing-six-new-cryptocurrencies-in-bigquery-public-datasets-and-how-to-analyze-them
57. GOOGLE. BigQuery pricing. Google Cloud [interaktyvus]. N.d. [žiūrėta 2020-05-26]. Prieiga per: https://cloud.google.com/bigquery/pricing
58. DAY, Allen. Bitcoin mining pool address signatures and statistics of their behavior over time. *GitHub* [interaktyvus]. 2019 [žiūrėta 2020-05-26]. Prieiga per: https://gist.github.com/allenday/16cf63fb6b3ed59b78903b2d414fe75b
59. GANJI, Venkata Ratnam; MANNEM, Siva Naga Prasad. Credit card fraud detection using anti-k nearest neighbor algorithm. *International Journal on Computer Science and Engineering*, 2012, 4.6: 1035-1039.
60. MALINI, N.; PUSHPA, M. Analysis on credit card fraud identification techniques based on KNN and outlier detection. In: *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*. IEEE, 2017. p. 255-258.
61. AGGARWAL, Charu C.; YU, Philip S. Outlier detection for high dimensional data. In: *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*. 2001. p. 37-46.
62. MAHESHWARI, Diksha. Payment Card Fraud Detection with Data Mining: A Review. In: *ICDSMLA 2019*. Springer, Singapore, 2020. p. 1579-1589.
63. AGGARWAL, Charu C. *Outlier analysis*. Springer, Cham, 2015. ISBN 9783319475783.
64. THORNTON, Dallas, et al. Outlier-based Health Insurance Fraud Detection for US Medicaid Data. In: *ICEIS (2)*. 2014. p. 684-694.
65. NIAN, Ke, et al. Auto insurance fraud detection using unsupervised spectral ranking for anomaly. *The Journal of Finance and Data Science* [interaktyvus]. 2016, 2.1: 58-75 [žiūrėta 2020-05-26]. doi: 10.1016/j.jfds.2016.03.001
66. VANHOEYVELD, Jellis; MARTENS, David; PEETERS, Bruno. Value-added tax fraud detection with scalable anomaly detection techniques. *Applied Soft Computing* [interaktyvus]. 2020, 86: 105895 [žiūrėta 2020-05-26]. doi: 10.1016/j.asoc.2019.105895

67. DOMINGUES, Rémi, et al. A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition* [interaktyvus]. 2018, 74: 406-421 [žiūrėta 2020-05-26]. doi: 10.1016/j.patcog.2017.09.037
68. ŽILINSKAS, Evaldas. Fintech rizikos ir bitkoino kriptografinės valiutos neįprastų transakcijų mašininis identifikavimas. *Fizinių Ir Technologijos Mokslų Tarpdalykiniai Tyrimai: 10-oji Jaunųjų Mokslininkų Konferencija: Pranešimų Santraukos*. 2020. p. 28
69. HAN, Jiawei; PEI, Jian; KAMBER, Micheline. *Data mining: concepts and techniques*. Waltham: Elsevier, 2011. ISBN 9789380931913.
70. STREHL, Alexander; GHOSH, Joydeep. Cluster ensembles---a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research* [interaktyvus]. 2002, 583-617 [žiūrėta 2020-05-26]. Prieiga per: https://dl.acm.org/doi/10.1162/153244303321897735
71. GHOSH, Joydeep; ACHARYA, Ayan. Cluster ensembles. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* [interaktyvus]. 2011, 1.4: 305-315 [žiūrėta 2020-05-26]. doi: 10.1002/widm.32
72. DUDOIT, Sandrine; FRIDLYAND, Jane. Bagging to improve the accuracy of a clustering procedure. *Bioinformatics* [interaktyvus]. 2003, 19.9: 1090-1099 [žiūrėta 2020-05-26]. doi: 10.1093/bioinformatics/btg038
73. HDBSCAN. Benchmarking Performance and Scaling of Python Clustering Algorithms. *Read the Docs* [interaktyvus]. N.d. [žiūrėta 2020-05-26]. Prieiga per: https://hdbscan.readthedocs.io/en/latest/performance_and_scalability.html
74. ŽILINSKAS, Evaldas. Neįprastų transakcijų mašininis identifikavimas blokų grandinėje. *Matematika ir matematikos dėstymas, Kaunas, Lietuva, balandžio 26 d., 2019: pranešimas.*
75. LIU, Fei Tony; TING, Kai Ming; ZHOU, Zhi-Hua. Isolation forest. In: *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008. p. 413-422.
76. CHEN, Wo-Ruo, et al. Representative subset selection and outlier detection via isolation forest. *Analytical methods* [interaktyvus]. 2016, 8.39: 7225-7231 [žiūrėta 2020-05-26]. doi: 10.1039/C6AY01574C
77. BARTOLETTI, Massimo, et al. Dissecting Ponzi schemes on Ethereum: identification, analysis, and impact. *Future Generation Computer Systems* [interaktyvus]. 2020, 102: 259-277 [žiūrėta 2020-05-26]. doi: 10.1016/j.future.2019.08.014
78. MEHAR, Muhammad Izhar, et al. Understanding a revolutionary and flawed grand experiment in blockchain: the DAO attack. *Journal of Cases on Information Technology (JCIT)* [interaktyvus]. 2019, 21.1: 19-32 [žiūrėta 2020-05-26]. doi: 10.4018/JCIT.2019010102

Priedai

1 priedas. Duomenų rinkinio *crypto_bitcoin* lentelės *blocks* struktūra

Lauko pavadinimas	**Tipas**	**Lauko apibūdinimas**
hash	Simbolių eilutė	Bloko maiša
size	Sveikasis skaičius	Bloko duomenų dydis baitais
stripped_size	Sveikasis skaičius	Bloko duomenų dydis, neįskaitant liudijimo (angl. *witness*) duomenų
weight	Sveikasis skaičius	Bloko svoris (3 * bazinis dydis + visas dydis)
number	Sveikasis skaičius	Bloko numeris
version	Sveikasis skaičius	Bloko antraštėje nurodyta protokolo versija
merkle_root	Simbolių eilutė	Merkle medžio šakninis mazgas, kuriame lapai yra sandorių maišos
timestamp	Laiko žyma	Bloko antraštėje nurodyta bloko sukūrimo data
timestamp_month	Data	Mėnuo, kurį buvo sukurtas blokas
nonce	Simbolių eilutė	Bloko antraštėje nurodytas sprendimo sunkumas
bits	Simbolių eilutė	Bloko antraštėje nurodytas sunkumo slenkstis
coinbase_param	Simbolių eilutė	Duomenys, nurodyti šio bloko pirmame (išgavimo) sandoryje
transaction_count	Sveikasis skaičius	Sandorių skaičius bloke

2 priedas. Duomenų rinkinio *crypto_bitcoin* lentelės *transactions* struktūra

Lauko pavadinimas	**Tipas**	**Lauko apibūdinimas**
hash	Simbolių eilutė	Sandorio maiša
size	Sveikasis skaičius	Sandorio dydis baitais
virtual_size	Sveikasis skaičius	Virtualus sandorio dydis (skiriasi liudijimo sandoriams)
version	Sveikasis skaičius	Bloke, kuriam priklauso sandoris, nurodyta protokolo versija
lock_time	Sveikasis skaičius	Ankstyviausias laikas, kai šis sandoris gali būti pridėta naujausiame bloke
block_hash	Simbolių eilutė	Bloko, kuriam priklauso sandoris, maiša
block_number	Sveikasis skaičius	Bloko, kuriam priklauso sandoris, numeris
block_timestamp	Laiko žyma	Bloko, kuriam priklauso sandoris, laiko žyma
block_timestamp_month	Data	Bloko, kuriam priklauso sandoris, mėnuo
input_count	Sveikasis skaičius	Įvesčių skaičius sandoryje
output_count	Sveikasis skaičius	Išvesčių skaičius sandoryje
input_value	Dešimtainis skaičius	Įvesčių vertė sandoryje
output_value	Dešimtainis skaičius	Išvesčių vertė sandoryje
is_coinbase	Loginis	Ar sandoris yra pirmas (išgavimo)
fee	Dešimtainis skaičius	Už šį sandorį sumokėtas mokestis
inputs	Pasikartojantis įrašas	Sandorio įvestys

Lauko pavadinimas	Tipas	Lauko apibūdinimas
inputs.index	Sveikasis skaičius	Sandorio įvesties numeris (numeruojama nuo 0)
inputs.spent_transaction_hash	Simbolių eilutė	Sandorio, kuris yra išvestis, kurią šį įvestis išleidžia, maiša
inputs.spent_output_index	Sveikasis skaičius	Išvesties, kurią ši įvestis išleidžia, indeksas
inputs.script_asm	Simbolių eilutė	Bitkoino skriptų kalbos operacijų kodai simbolinis atvaizdavimas
inputs.script_hex	Simbolių eilutė	Bitkoino skriptų kalbos operacijų kodo šešioliktainis atvaizdavimas
inputs.sequence	Sveikasis skaičius	Skaičius skirtas leisti atnaujinti nepatvirtintus užrakintus sandorius; šiuo metu nenaudojamas, išskyrus uždrausti užrakinimą sandoryje
inputs.required_signatures	Sveikasis skaičius	Parašų skaičius, reikalingas autorizuoti išlestai išvesčiai
inputs.type	Simbolių eilutė	Išleistos išvesties adreso tipas
inputs.addresses	Simbolių eilutė	Adresas, kuriam priklauso išleista išvestis
inputs.value	Dešimtainis skaičius	Vertė bazine valiuta, pridedama prie išleistos išvesties
outputs	Pasikartojantis įrašas	Sandorio išvestys
outputs.index	Sveikasis skaičius	Išvesties indeksas sandoryje, kuris bus naudojamas vėlesniuose sandoriuose, nurodant konkrečią išvestį (numeruojamas nuo 0)
outputs.script_asm	Simbolių eilutė	Bitkoino skriptų kalbos operacijų kodo simbolinis atvaizdavimas
outputs.script_hex	Simbolių eilutė	Bitkoino skriptų kalbos operacijų kodo šešioliktainis atvaizdavimas
outputs.required_signatures	Sveikasis skaičius	Parašų skaičius, reikalingas autorizuoti išleidžiant šitą išvestį
outputs.type	Simbolių eilutė	Išvesties adreso tipas
outputs.addresses	Simbolių eilutė	Adresas, kuriam priklauso šita išvestis
outputs.value	Dešimtainis skaičius	Vertė bazine valiuta, pridedama prie šios išvesties

3 priedas. Duomenų rinkinio *crypto_ethereum* lentelės *blocks* struktūra

Lauko pavadinimas	Tipas	Lauko apibūdinimas
timestamp	Laiko žyma	Laikas, kada blokas buvo
number	Sveikasis skaičius	Bloko numeris
hash	Simbolių eilutė	Bloko maiša
parent_hash	Simbolių eilutė	Viršblokio maiša
nonce	Simbolių eilutė	Atlikto darbo įrodymo maiša
sha3_uncles	Simbolių eilutė	Dėdžių duomenų bloke SHA3 maiša
logs_bloom	Simbolių eilutė	Bloko žurnalų Bloom filtras
transactions_root	Simbolių eilutė	Bloko sandorių medžio šaknis
state_root	Simbolių eilutė	Bloko būsenų medžio šaknis
receipts_root	Simbolių eilutė	Bloko kvitų medžio šaknis

Lauko pavadinimas	Tipas	Lauko apibūdinimas
miner	Simbolių eilutė	Naudos gavėjo adresas, kuriam atiduodamas bloko atlygis
difficulty	Dešimtainis skaičius	Bloko sudėtingumas
total_difficulty	Dešimtainis skaičius	Grandinės iki šio bloko bendras sudėtingumas
size	Sveikasis skaičius	Bloko dydis baitais
extra_data	Simbolių eilutė	Bloko papildomi duomenys
gas_limit	Sveikasis skaičius	Maksimalus bloko degalų limitas
gas_used	Sveikasis skaičius	Kiek degalų išnaudojo sandoriai šiame bloke
transaction_count	Sveikasis skaičius	Sandorių skaičius bloke

4 priedas. Duomenų rinkinio *crypto_ethereum* lentelės *transactions* struktūra

Lauko pavadinimas	Tipas	Lauko apibūdinimas
hash	Simbolių eilutė	Sandorio maiša
nonce	Sveikasis skaičius	Kiek siuntėjas prieš tai yra padaręs sandorių
transaction_index	Sveikasis skaičius	Sandorio pozicijos indeksas bloke
from_address	Simbolių eilutė	Siuntėjo adresas
to_address	Simbolių eilutė	Gavėjo adresas
value	Dešimtainis skaičius	Pervedama suma vėjais
gas	Sveikasis skaičius	Siuntėjo suteikti degalai
gas_price	Sveikasis skaičius	Siuntėjo nurodyta degalų kaina vėjais
input	Simbolių eilutė	Duomenys siunčiami kartu su šiuo sandoriu
receipt_cumulative_gas_used	Sveikasis skaičius	Bendras sunaudotas degalų kiekis, kai šis sandoris buvo atlikta bloke
receipt_gas_used	Sveikasis skaičius	Šio sandorio sunaudotas degalų kiekis
receipt_contract_address	Simbolių eilutė	Sukurtas sutarties adresas, jei buvo sutarties sudarymo sandoris (priešingu atveju *null*)
receipt_root	Simbolių eilutė	*stateroot* 32 baitai po sandorio
receipt_status	Sveikasis skaičius	1 – sėkmingas, 0 – nesėkmingas
block_timestamp	Laiko žyma	Bloko, kuriam priklauso sandoris, laiko žyma
block_number	Sveikasis skaičius	Bloko, kuriam priklauso sandoris, numeris
block_hash	Simbolių eilutė	Bloko, kuriam priklauso sandoris, maiša

5 priedas. Duomenų rinkinio *crypto_ethereum* lentelės *traces* struktūra

Lauko pavadinimas	Tipas	Lauko apibūdinimas
transaction_hash	Simbolių eilutė	Sandorio maiša
transaction_index	Sveikasis skaičius	Sandorio pozicijos indeksas bloke
from_address	Simbolių eilutė	Siuntėjo adresas
to_address	Simbolių eilutė	Gavėjo adresas
value	Dešimtainis	Pervedama suma vėjais

Lauko pavadinimas	**Tipas**	**Lauko apibūdinimas**
	skaičius	
input	Simbolių eilutė	Duomenys siunčiami kartu su šiuo sandoriu
output	Simbolių eilutė	Žinutės įvykdymo išvestis
trace_type	Simbolių eilutė	Pėdsako tipas (viena iš reikšmių *call, create, suicide, reward, genesis, daofork*)
call_type	Simbolių eilutė	Vykdymo tipas (viena iš reikšmių *call, callcode, delegatecall, staticcall*)
reward_type	Simbolių eilutė	Atlygio tipas (viena iš reikšmių *block, uncle*)
gas	Sveikasis skaičius	Žinutės vykdymui suteikiami degalai
gas_used	Sveikasis skaičius	Žinutės vykdymui sunaudoti degalai
subtraces	Sveikasis skaičius	Popėdsakių skaičius
trace_address	Simbolių eilutė	Kableliais atskirtas pėdsakų adresų esančių vykdymo medyje sąrašas
error	Simbolių eilutė	Klaida jei žinutės vykdymas neapvyko.
status	Sveikasis skaičius	1 – jei sėkmingai žinutė įvykdyta, priešingu atveju 0.
block_timestamp	Laiko žyma	Bloko, kuriam priklauso pėdsakas, laiko žyma
block_number	Sveikasis skaičius	Bloko, kuriam priklauso pėdsakas, numeris
block_hash	Simbolių eilutė	Bloko, kuriam priklauso pėdsakas, maiša
trace_id	Simbolių eilutė	Unikali simbolių eilutė identifikuojanti pėdsaką.