Kauno technologijos universitetas

Matematikos ir gamtos mokslų fakultetas

KRIPTOVALIUTŲ IR INTERNETINĖS INFORMACIJOS SĄRYŠIŲ TYRIMAS TAIKANT DUOMENŲ GAVYBOS METODUS

Baigiamasis magistro projektas

Giedrius Petrošius
Projekto autorius

Prof. dr. Robertas Alzbutas
Vadovas
Doc. dr. Alina Stundžienė
Vadovė

Kaunas, 2018

Kauno technologijos universitetas

Matematikos ir gamtos mokslų fakultetas

KRIPTOVALIUTŲ IR INTERNETINĖS INFORMACIJOS SĄRYŠIŲ TYRIMAS TAIKANT DUOMENŲ GAVYBOS METODUS

Baigiamasis magistro projektas

Didžiųjų verslo duomenų analitika (621G12002)

Giedrius Petrošius
Projekto autorius

Prof. dr. Robertas Alzbutas
Vadovas
Doc. dr. Alina Stundžienė
Vadovė

Doc. dr. Kristina Poškuvienė
Recenzentė
Doc. dr. Jurgita Bruneckienė
Recenzentė

Kaunas, 2018

Kauno technologijos universitetas
Matematikos ir gamtos mokslų fakultetas
Giedrius Petrošius

Kriptovaliutų ir internetinės informacijos sąryšių tyrimas taikant duomenų gavybos metodus

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Giedriaus Petrošiaus, baigiamasis projektas tema „Kriptovaliutų ir internetinės informacijos sąryšių tyrimas taikant duomenų gavybos metodus“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

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

_____ _____

(vardą ir pavardę įrašyti ranka) (parašas)

Turinys

Paveikslų sąrašas

Lentelių sąrašas

Petrošius, Giedrius. Kriptovaliutų ir internetinės informacijos sąryšių tyrimas taikant duomenų gavybos metodus. Magistro baigiamasis projektas / vadovai Prof. doc. Robertas Alzbutas, Doc. dr. Alina Stundžienė; Kauno technologijos universitetas, Matematikos ir gamtos mokslų fakultetas.
Studijų kryptis ir sritis: Fiziniai mokslai, Matematika (01 P)
Reikšminiai žodžiai: Kriptovaliuta, LSTM, dinaminis temų modeliavimas.
Kaunas, 2018. 45 p.

Santrauka

Bitkoinas populiari skaitmeninė valiuta naudojama visame pasaulyje internetiniams mokėjimams atlikti. Bitkoinas ir kitos kriptovaliutos tapo investavimo instrumentu, taip pat juo prekiaujama kaip ir įprastomis valiutomis.

Tyrimo tikslas yra taikant duomenų gavybos metodus iš prieinamos internetinės informacijos išgauti vertingus dėsningumus kriptovaliutų rinkoje. Šiam uždaviniui pasiekti buvo pritaikytas dinaminis temų modeliavimas, naudojant 1,37 milijono komentarų, rasti pasislėpusias temas (klasterius) ir jų evoliucijas 4,3 metų laikotarpyje (nuo 2014m sausio iki 2018m. kovo).

Pasinaudojus internetine informacija tyrime buvo prognozuojama bitkoino kaina LSTM neuroniniais tinklais, kurių geriausias pasiektas tikslumas yra 112,65 RMSE ir MAPE – 0,65%. Remiantis LTSM modeliu buvo sukurta automatizuota prekiavimo strategija, kurios pirkimo ir pardavimo sprendimai buvo 94 % teisingi.

Petrošius, Giedrius. Relation between cryptocurrency and online information analysis applying data mining methods. Master's Final Degree Project / supervisor assoc. Prof. Dr Robertas Alzbutas, Assoc. Prof. Dr Alina Stundžienė; Faculty of Mathematics and Natural Science, Kaunas University of Technology.
Study field and area: Natural Sciences, Mathematics (01 P).
Keywords: Cryptocurrency, LSTM, dynamic topic modeling.
Kaunas, 2018. 45 pages.

Summary

Bitcoin is a trending digital currency that is used worldwide to make online payments. Bitcoin and other cryptocurrencies have consequently become an investment vehicle in itself and are traded in a way similar to other open currencies.

The aim of this thesis to use data mining techniques on online information in order to find cryptocurrency market patterns. To accomplish thesis goal dynamic topic modeling was applied to 1,37 million comments in order to find hidden topics evolutions over a period of 4,3 years from January 2014 to March 2018.

The research is also concerned with predicting the price of Bitcoin using online information. The task was achieved with Long-Short Memory (LSTM) neural network which highest accuracy is 112,65 RMSE and MAPE – 0,65%. Based on LSTM model Bitcoin automated trading strategy was developed which decisions sell and bitcoins was 94% accurate.

SANTRUMPOS

API – programinė sąsaja

DJIIA - Dow Jones indeksas

DTM - dinaminis temų modeliavimas

ERH - efektyvioji rinkos hipotezė

LDA - Latentinis Dirichleto pasiskirstymo modelis

LSTM – ilgos-trumpos atminties rekurentiniai neuroniniai tinklai

MAPE – vidutinė procentinė absoliutinė paklaida

RMSE - vidutinė kvadratinė paklaida

TM - temų modeliavimas

ĮVADAS

Pirmoji kriptovaliuta – Bitkoinas buvo pristatytas 2008m. kaip internetinė mokėjimo sistema išsprendžianti dvigubo mokėjimo problemą ir pašalinanti trečiosios šalies dalyvavimą transakcijų patvirtinimui. Kriptovaliutos greitai pasaulyje išpopuliarėjo savo kainos augimu, kuris dažniausiai fiksuojamas ne procentinė išraiška, o kartais. Daugelis žmonių susižavėję šiuo augimu diversifikuoja savo investicinius portfelius, kiti juos „kasa“, o treti bando išnaudoti jų didelius svyravimus – aktyviai prekiauja kriptovaliutų rinkoje.

Dauguma internetinių vartotojų priima sprendimus pirkti ar parduoti kriptovaliutas pasirodžius naujai internetiniai informacijai. Vieniems užtenka vienos spaudos naujienos, kitiems - ekspertų išreikštos nuomones socialiniuose tinkluose, o dalis žmonių bando sprendimus priimti kuo racionaliau diskutuodami ir ieškodami atsakymų su kriptovaliutomis susijusiuose internetiniuose forumuose ar kitose kriptovaliutų bendruomenėse.

Procesas apžvelgti internetinius duomenis, norint įvertinti skaitmeninių valiutų rinkos būseną, yra labai imlus laikui. Paspartinti šį procesą yra naudojami duomenų gavybos metodai, kurie surenka internetinę informaciją, ją apdoroja ir bando juose atrasti įžvalgas ar reikšmingus dėsningumus. Dėl to šis procesas ne tik sutaupo laiką, bet leidžia atrasti ir įvertinti su kriptovaliutomis susijusias rizikas ir sukurti pranašesnę investavimo ar prekiavimo strategiją.

Tikslas. Sudaryti kriptovaliutos ir internetinės informacijos sąryšius nusakantį modelį ir prognozuoti kriptovaliutos kainą naudojant duomenų gavybos ir mašininio mokymo metodus.

Uždaviniai:

- Atlikti kriptovaliutos ir internetinės informacijos išgavimą ir apdorojimą, bei duomenų charakteristikų išskyrimą ir vertinimą;
- Nustatyti internetinės informacijos (forumo komentarų) atskirose laiko intervaluose vyraujančias temas (klasterius) ir jų evoliuciją laike;
- Sudaryti kriptovaliutos ir internetinės informacijos sąryšius nusakantį modelį taikant neuroninį tinklą ir parinkti modelio parametrus;
- Taikant sudarytą modelį prognozuoti kriptovaliutos kainą, sudaryti kriptovaliutų prekiavimo strategiją ir ją pademonstruoti.

1. LITERATŪROS APŽVALGA

Pirmykštėje bendruomenės santvarkoje žmogus negalėjo visko pats pasigaminti, taigi iškilo būtinybė keistis rezultatais. Bėgant laikui formavosi ypatingos prekės poreikis (pinigų), atliekančios prekių mainuose visuotinio ekvivalento vaidmenį. Pirmųjų pinigų vaidmenį atlikdavo dažniausiai vartojamos ir parduodamos prekės, jos būdavo tokios, koks buvo tautos verslas: galvijai, odos, arbata, tabakas, ginklai ir kt. Palengvinti paslaugų ir gaminių prekybą graikų filosofas Aristotelis iškėlė keturis kriterijus [1] „geriems pinigams“ apibūdinti:

1) **tvarumas**: pinigai turi būti atsparūs dėvėjimuisi, t. y., išlaikyti savo formą, medžiagą bėgant laikui;

2) **portatyvumas**: pinigai turi būti lengvai gabenami ir jų vertė turi būti aukšta, atsižvelgiant į jų dydį ir svorį;

3) **dalumas**: pinigai turi būti lengvai išskaidomi dalimis ir atgal sujungiami be jokios įtakos jų fundamentaliosioms savybėms;

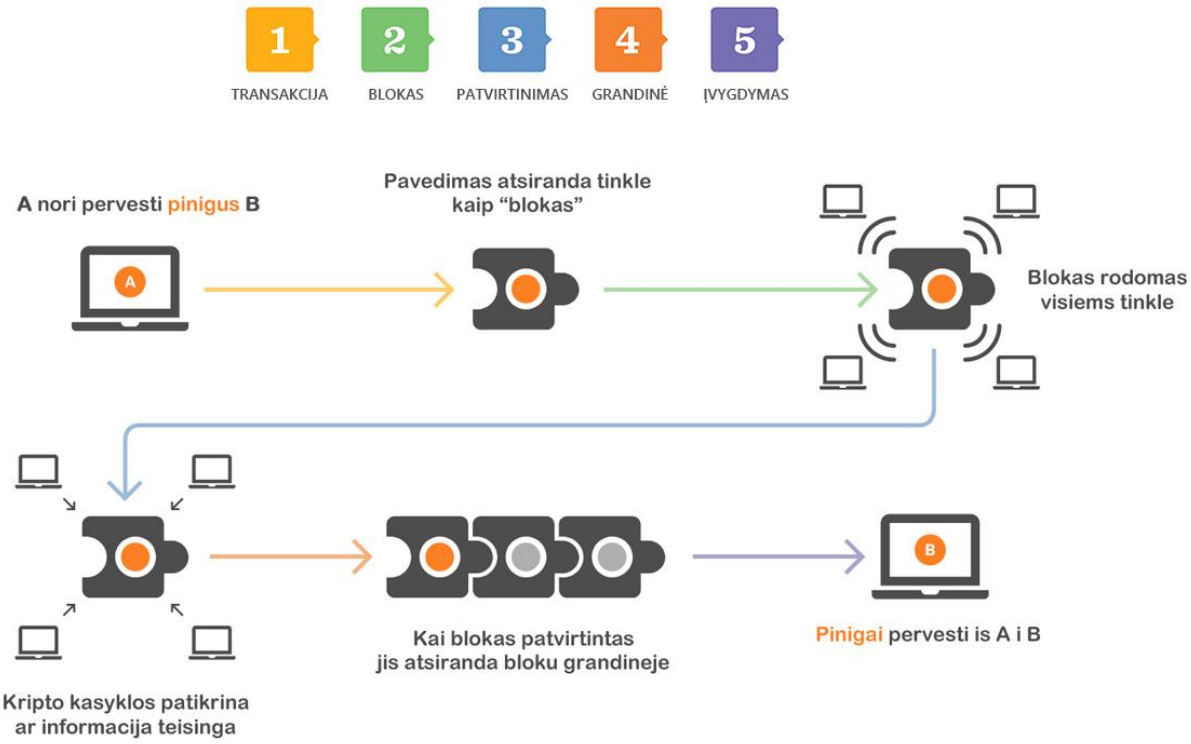
4) **vidinė vertė**: pinigai turi būti nepriklausomi nuo kitų objektų ir būti vertingi patys iš savęs.

Tuomet visuotiniu mainų ekvivalentu buvo pasiūlytas auksas, kuris puikiai atitiko Aristotelio kriterijus, tačiau ekonomikai augant mainų ekvivalento paklausa didėjo, valstybės buvo priverstos sukurti labiau prieinamą mainų vertės matą, kurį būtų galima reguliuoti ir kontroliuoti. Tai buvo dekretinių pinigų gimimas. Dekretiniai pinigai ankstyvoje vystymosi raidoje buvo padengti tam tikru tauriojo metalo kiekiu, tačiau didėjant pinigų paklausai vyriausybės ar centriniai bankai vis daugiau jų spausdino, kol galiausiai visų esamų pinigų rinkoje vertė viršijo fizinių ekvivalentų vertę. Todėl dabar dekretiniai pinigai yra pinigai, nes taip tvirtina vyriausybė.

1.1. Bitkoino samprata

Pirmąją kriptovaliutą bitkoiną 2008m. straipsnyje pristatė S. Nakamotas [2]. Jame jis bitkoiną apibrėžė kaip „elektroninė pinigų mokėjimo sistema, paremta kriptografija vietoje pasitikėjimo“. Bitkoino mokėjimo sistema veikia blokų grandinių (angl. *blockchain)* technologija, kuri nuo kitų inovatyvių mokėjimo sistemų skiriasi tuo, kad išsprendžia dvigubo išleidimo problemą, taip pašalindama būtinybę dalyvauti trečiajai šaliai (kuri yra paremta pasitikėjimu) transakcijoms patvirtinti. Blokų grandinių technologija – decentralizuota vieša transakcijų saugojimo sistema. Ji transakcijas užšifruoja blokų grandinėmis, toks veikimo principas užtikrina informacijos saugumą ir vientisumą sistemoje. Šia technologija pagrįsta mokėjimo sistema yra decentralizuota, todėl nėra trečiųjų šalių skirtų įrašams saugoti, įrašai saugomi pačiame tinkle. Ši technologija vadinama

revoliucine, viena reikšmingiausių technologinių inovacija nuo pat interneto atsiradimo. Ją galima pritaikyti praktiškai visose srityse ir pasak IBM vadovo G. Rometį „blokų grandinių technologija pakeis operacijas kaip internetas pakeitė informacijos pasiekiamumą“. Sistemos transakcijos, paremtos blokų grandinių technologija (žr. 1 pav.), skiriasi penkiomis savybėmis nuo įprastai atliekamų piniginių pavedimų [3]:



1 pav. Blokų grandinių technologijos veikimo schema [4]

1. **negrįžtamos (*angl.* irreversible):** po transakcijos patvirtinimo, jos atšaukti neįmanoma, nei vienas sistemos naudotojas neturi teisių sugrąžinti įvykdytos transakcijos valiutos, vienintelis spendimas – įvykdyti naują atgalinę transakciją;

2. **pseudonimiškos (*angl.* pseudonymous):** visos transakcijos yra atviros ir yra galimybė pasižiūrėti kur, kada ir kas keliauja, bet nei transakcija, nei vartotojo paskyra nėra susieta su tikrąją asmens tapatybe;

3. **greitos ir globalios:** transakcijos užklausos į tinklą patenka iškart, bet jų patvirtinimas užtrunka kelias dešimtis minučių (priklausomai nuo transakcijų skaičiaus). Kadangi transakcijos vyksta globaliame tinkle, transakcijos laikas nepriklauso nuo vartotojų lokalizacijos;

4. **saugios:** kriptovaliutos vertė yra užrakinama kriptografinio protokolo viešuoju raktu ir tik valiutos savininkas gali įvykdyti transakciją pasirašydamas privačiuoju raktu. Kriptografinis

protokolas paremtas didelių pirminių skaičių matematiniais skaičiavimais, todėl kibernetiniams nusikaltėliams apdoroti kriptografinius skaičiavimus, norint atspėti privatųjį raktą naudojant įprastus kompiuterius, ko gero truktų daugiau laiko, negu egzistuoja mūsų visata;

5. **nereikalaujančios leidimų (*angl.* permissionless):** naudotis kriptovaliutomis nereikia jokių leidimų ar reikalavimų, tereikia atsisiųsti nemokamą programinę įrangą ir ją įdiegti, tuomet skaitmeninę valiutą bus galima gauti ir išsiųsti kitiems. Taip pat niekas negali uždrausti naudotis sistema ir disponuoti kriptovaliutomis.

Techniškai Bitkoinas (tikrinis daiktavardis) yra protokolas ir programinė įranga, veikianti „vartotojas vartotojui“ (angl. *peer-to-peer*) kompiuterių tinklų modelyje, tačiau bitkoinas (bendrinis daiktavardis) taip pat ir sistemos vertės matas – valiuta. Nors Nakamoto tikslas buvo Bitkoiną pristatyti kaip mokėjimo sistemą, tačiau technologijos potencialas viršijo lūkesčius ir daugelį startuolių įkvėpė kurti naujas paslaugas naudojant Bitkoino technologinę koncepciją. Šiuo metu jau yra 1574 aktyvių kriptovaliutų [5] pagrįsta šia technologija.

Apibendrinant Bitkoino mokėjimo sistema nuo tradicinių valiutų atsiskaitymų skiriasi keturiais fundamentaliais aspektais:

1. Bitkoinas nėra valdomos centrinės institucijos, o decentralizuoto kompiuterių tinklo prie kurio gali prisijungti ir išeiti bet kuris internetinis vartotojas;
2. Transakcijos tarp dviejų šalių gali būti įvykdomas tinkle be tarpininką atstovaujančios finansinės institucijos;
3. Kriptovaliutos vertę nustato rinkos paklausa, o ne subjektyvi centrinio banko politika;
4. Bitkoino pasiūla didėja iš anksto nustatyta mažėjančia tendencija iki 2140 m., tuomet rinkoje bus 21 milijonas bitkoinų.

1.2. Kriptovaliutos – turto klasė

Kriptovaliutų rinka - visiškai skaitmeninė internetinė rinka, kuri finansiniu požiūriu turi potencialą sujungti pasaulyje vyraujančias ekonomikas. Priešingai negu įprastos valiutos, kriptovaliutos neturi fizinės rinkos lokacijos, bei plačiai naudojamų rodiklių apibūdinančių jų ekonominę būseną. 2013 metais vieno bitkoino vieneto kaina pakilo nuo 12 iki 1242,2 JAV dolerių. Tokia kainos raida atkreipė pasaulio dėmesį ir jį smarkiai išskyrė iš kitų tradicinių valiutų. Žmonės pradėjo žiūrėti į kriptovaliutą kaip į investavimo įrankį, o dvidešimties finansų ministrų ir centrinių bankų valdytojų grupė G20 nesenai klasifikavo bitkoiną - turto klase [6].

Bitkoinas, kaip skaitmeninė turto klasė, turi didelį privalumą, kad jis nekoreliuoja su kitomis turto klasėmis [7]. Ši unikali savybė sukuria įdomią alternatyvą diversifikuoti turimą investicijų portfelį. Kitas kriptovaliutos privalumas yra labai aukšti kainos vyravimai. Visi investuotojai uždirba iš turto klasių kainos kitimo, o bitkoinas investicinį portfelį paverčia agresyvia pelno ir rizikos finansine strategija.

Nepaisant didelio bitkoino kainos augimo, egzistuoja daug rizikų. Visų pirma, kaip anksčiau minėta, bitkoinas nėra valdomas centrinių banko ar valstybės, todėl kriptovaliuta neturi jokių saugumo garantijų t.y. jeigu buvote apgauti ir praradote savo skaitmenines lėšas, valstybė nesiims jokių veiksmų. Taip pat jeigu prarasite savo privačius raktus ar atliksite transakciją ne ten kur norite – virtualių pinigų atgauti nepavyks. S.Sukumulja ir C. Sikora nurodo dar kelias rizikas susijusias su bitkoino prekiavimu [8]:

Reguliacijos rizika: tai viena didžiausia egzistuojanti rizika bitkoinui kaip skaitmeninai valiutai ir alternatyviai investicijai, pvz., Kinijos centrinis bankas uždraudė kriptovaliutų biržą teigdama, kad kriptovaliutas sunku kontroliuoti, jos yra naudojamos nusikalstamai veikai ir jos gali padaryti žalos kitoms pasaulinėms investicijoms. Įvykus tokiems valstybinio lygio sprendimams bitkoino kaina stipriai krenta;

Bitkoino plėtojimosi problema (angl. *Bitcoin scalability problem*): didėjant vartotojų skaičiui sistemoje atliekama vis daugiau transakcijų. Šiuo metu grandinės bloke atsiradusias transakcijas apdoroti įprastai užtenka apie 20 - 40 minučių. Iškilus būtinybei pagreitinti sistemos darbą, galimas bitkoino skilimas į kelias valiutas [9] ar kitoks techninis sprendimas, kas į rinką įneštų naujų neapibrėžtumų;

Kiberatakos: kai į didelės bitkoino kompanijas ir keityklas yra įsilaužta ir pasisavinti jų turimų kriptovaliutų privatieji raktai – valiutos kainos krenta;

Kitos valiutos įtaka: kaip minėta anksčiau yra daug kitų alternatyvių kriptovaliutų. Jei kitas skaitmeninis žetonas bus pigesnis, inovatyvus ir sudomins žmones, yra tikimybė, kad žmonės pradės „bėgti“ iš bitkoino rinkos į naująją;

1.3. Finansinių aktyvų prognozavimo teorija

Kriptovaliutos yra nauja turto klasė. Kaip ir kiekvienoje finansų rinkoje finansininkai ir investuotojai bando įvertinti, prognozuoti aktyvo vertę t.y. išspręsti pagrindinį finansų teorijos uždavinį – aprašyti kainos susidarymo mechanizmą. Kadangi bitkoino rinka vis dar yra ankstyvoje raidoje palyginus su kitais finansiniais aktyvais, todėl apžvelgti kainos susidarymo mechanizmai yra

koncentruoti į akcijų rinką. Vertinant akcijos kainų prognozavimą ilgą laiką buvo remiamasi dvejomis teorijomis – atsitiktinio vaikščiojimo teorija (angl. *Random walk*) ir efektyviąja rinkos hipoteze (angl. *Eficient Market Hypothesis*).

Atsitiktinio vaikščiojimo teorija teigia, kad akcijos kainos kitimas yra atsitiktinis procesas, turintis tą patį kainos pokyčių pasiskirstymą. Kitimas neturi jokių dėsningumų t.y. nepriklauso nuo rinkos, ankstesnės akcijos kainos dinamikos, jos trendo ar istorijos. Atsitiktinio vaikščiojimo teorijos idėja yra, kad akcijos kaina kinta visiškai nepriklausomai ir jos neįmanoma prognozuoti.

Efektyvioji rinkos hipotezė (ERH) – Fama [10] struktūrizuota idėja teigianti, kad vertybinių popierių kainas ir jų pokyčius rinkoje atspindi visa esama informacija apie jas. Efektyvumas šiame kontekste charakterizuoja aktyvo kainos koregavimąsi rinkoje atsiradus naujai informacijai. Kad rinka būtų efektyvi, turi galioti keletas prielaidų: rinkoje turi būti daug investuotojų, taip pat jie turi būti racionalūs (teisingai įsisavina informaciją, todėl racionaliai įkainoja vertybinius popierius). Fama [10] išskyrė tris egzistuojančias ERH formas:

- **silpna forma**: dabarties kainos atspindi visą praeities informaciją, šioje formoje vertybinių popierių fundamentali analizė investuotojui suteikia galimybę gauti didesnę grąžą negu rinkos vidurkis trumpuoju laikotarpiu, bet jokių dėsningumo rinkoje nėra ir ilguoju laikotarpiu ši analizė neveikia, techninė analizė šioje formoje neveikia;

- **pusiau stipri forma**: dabarties kainos atspindi visą dabarties momentu prieinamą viešąją informaciją. Manoma, kad šioje formoje kaina visuomet yra paklausos ir pasiūlos pusiausvyros taške. Fundamentali ir techninė analizė investuotojams jokio pranašumo šioje formoje nesuteikia;

- **stipri forma**: dabarties kainos atspindi visą dabarties momentu prieinamą informaciją – viešą ir privačią. Šioje formoje nei vienas investuotojas negali įgauti pranašumo prekiaudamas rinkoje.

Taigi pasak ERH rinka yra visada efektyvi ir joks investuotojas negali „laimėti prieš rinką“ (nebent jis turi privačią informaciją) ir gauti didelių investicinių grąžų naudojantis pasenusia informacija ar kopijuojant ankstesnius kainų kitimo modelius [11], o praktikoje geri investuotojų rezultatai priskiriami sėkmei. Tačiau tokie įvykiai kaip 1920m. Wall Street bumas kuris baigėsi 1929m. akcijų rinkos griūtimi, 1987m. Juodasis pirmadienis ir paskutinioji finansinė krizė, kai DJIIA indeksas (angl. *Dow Jones industrial avareage index)* prarado 54% vertės, negali būti paaiškinami racionalia rinkos elgsena, kuri apibrėžiama EMH. Po paskutinės finansinės krizės keletas asmenų pasisakė, kad EMH reikėtų atsisakyti, nes tikėjimas, kad rinka save „prisižiūrės“ ir nesuformuos

finansinių burbulų yra naivus [12]. Įdomu, jog J. Kviginsas savo straipsnyje [13] rašo, kad bitkoino burbulas yra aiškus EMH paneigimas.

Vienas naujausių požiūrių į finansų rinkas bei ekonomika pasiūlė lietuvių kilmės Nobelio premijos laureatas Robertas J. Šileris. Jis išvystė naratyvų ekonomikos teorija [14], kurioje vietoj įprasto neoklasikinio racionalaus agento yra naratyvais besivadovaujantis ne itin racionalus asmuo. Kuris gali pirkti pervertintus aktyvus arba pardavinėti nuvertintus, nes jo aplinkoje populiarūs naratyvai teigia, kad taip elgtis logiška. Naratyvai yra istorijų apibendrinimai, jie aiškūs, lengvai suprantami, tačiau tuo pat metu turi stiprią emocinę vertę.

Naratyvų teorija (angl. *narrative economics*) teigia, kad investuotojai nėra linkę gilintis į faktus, visą informaciją apie tikrus aktyvus, o būtent į naratyvus. Tarkime žiniasklaida suformuoja kokį stiprų naratyvą, o skaitytojai juo patiki, tuomet jie naratyvą per įvarius kanalus pradeda platinti didesniam ratui žmonių. Tokį naratyvų plitimą Šileris prilygina su virusų plitimu ir epidemija, kuris puikiai paaiškina burbulus ir krizes. Burbulo metu teigiamu naratyvu užsikrėtę agentai pradeda pirkti aktyvus, taip naratyvui, lyg epidemijai, padeda plisti visuomenėje. Epidemijai pasiekus piką, aktyvo kaina labai išsipučia, užsikrėtę žmonės po mažu pradeda sveikti ir pradeda žiūrėti į faktus – įgauna racionalumo, tai ir sustabdo tolimesnį kainos augimą. O vėliau, kuo daugiau žmonių pasveiksta virusinį naratyvą visuomenėje pakeičia kitas masinė panika.

1.4. Bitkoino kainą įtakojantis veiksniai

Išaugus bitkoino populiarumui daugelis pradėjo juo domėtis, tirti jo charakteristikas. Virtualaus aktyvo susidomėjimą išreiškė ne tik potencialūs investuotojai, bet ir akademinės srities atstovai, kurie pradėjo tyrinėt kas įtakoja bitkoino kainos kitimą.

Pasak L. Kristoufeko [15] bitkoino kaina negali būti paaiškinama standartinėmis ekonomikos teorijomis, tokiomis kaip ateities pinigų srautu, perkamosios galios ar palūkanų normos paritetu, nes įprastų valiutų paklausos ir pasiūlos dėsniai skiriasi keliomis ypatybėmis lyginant su bitkoino rinka. Visų pirma, kaip anksčiau minėta, bitkoinas nėra reguliuojamas vyriausybių ir centrinių bankų, taipogi jo kaina yra atskirta nuo tikrosios ekonomikos, todėl nėra jokių makroekonomikos principų įtakos. P. Ciaianas, M. Rajcaniova, ir A. Kancas savo tyrime [16] taip pat teigia, kad jokie makroekonomikos faktoriai neformuoja bitkoino kainos. Abu darbai teigia, kad tik vienas dalykas įtakoja kriptovaliutos kainą – investicijos patrauklumas. D. Yermackas [17] savo straipsnyje nurodo, kad bitkoino kaina turi stiprią koreliaciją su prekybos charakteristikomis arba jos pasiūla ir paklausa. Taip pat ji turi silpną koreliacija su klasikinėmis valiutomis ir jokie makroekonomikos įvykiai jos neveikia.

D. Van Wijkas [18], priešingai negu ankstesni tyrinėtojai, pabrėžia globalinės makroekonomikos raidos rolę Bitkoino kainos formavimui. Jo nuomone šią raidą atspindi akcijų biržos indeksai, valiutų kursai ir naftos kainos, pvz., akcijų indeksai atspindi didelių kompanijų rezultatus jų rinkos šalyse, o tai reprezentuoja bendrą makroekonomikos situaciją ir finansinės raidos būseną. Darbe Wijkas pateikia, kad Dow Jones indeksas, euro-dolerio kursas ir naftos kaina reikšmingai įtakoja Bitkoino kainą ilguoju laikotarpiu. Darbuose ([19], [20]) atrasta makroekonomikos indikatoriaus DJIIA neigiamas efektas bitkoino kainai.

Kiti akademiniai tyrinėtojai ([21],[22]) tvirtina, jog bitkoino kainos dinamika priklauso nuo rinkos spekuliacijos. Kuomet paklausa pakyla, kaina reaguodama taip pat kyla dėl ribotos išleidimo paklausos. Tyrime [23] teigiama, kad Bitkoino kaina nulemta rinkos principais, tokias kaip pasiūlos ir paklausos dėsniu, ilgo laikotarpio perspektyvoje. Kainos svyravimai rinkoje yra taip pat nulemti naujų dalyvių atėjimu į rinką ir jų skleidžiama informacija potencialiems rinkoms dalyviams (angl. *word-of-mouth*). Rinkos dalyvių augimas, neatsižvelgiant ar jie investuoja į bitkoiną ar tiesiog naudojasi mokėjimo sistema, kelia bitkoino kainą [24].

Dar vienas faktorius įtakojantis bitkoino kaina yra auksas. Y. Zhu [19] straipsnyje teigia, kad aukso kaina neigiamai veikia bitkoino kainą, bet ilguoju laikotarpiu ši sąveika nėra žymi, nors rinkos savo elgesiu yra panašios. Kitame tyrime [25] atrasta stipri bitkoino kainos koreliacija su aukso kaina.

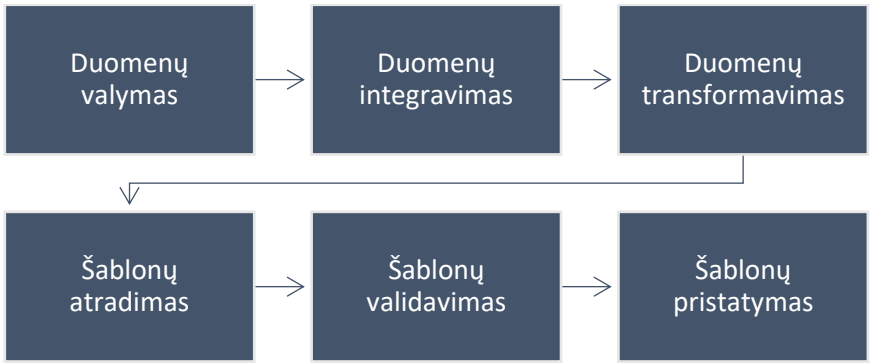
Taip pat daugelis akademinio pasaulio atstovų tyrė bitkoino kainos sąryšius su internetine informacija, apie tai plačiau sekančiame skyrelyje.

1.5. Duomenų gavyba

Bitkoinas, kaip anksčiau minėta, yra nereguliuojamas ir jo kaina rinkoje labai svyruojajanti, dėlto ją labai sunku įvertinti, bet kuriuo momentu. Kita vertus internete yra begalė informacijos šaltinių nestruktūrizuoto teksto kurį parašo kriptovaliutų rinkos dalyviai. Tokie duomenys atskleidžia vartotojų nuomones, išmintį, kas gali būti labai galingas indikatorius signalizuojantis svarbių įvykių įtaką kriptovaliutoms.

Kiekvienas metais duomenų sugeneruojama vis daugiau – jų augimas eksponentinis. Pasak IBM ataskaitos [26] 2017 m. 90% duomenų buvo sugeneruoti per paskutinius dvejus metus, o viename IBM straipsnyje [27] teigiama, kad 80 % visų esančių duomenų yra nestruktūrizuoti. Investuotojams norint racionaliai priimti spendimus perkant ar parduodant kriptovaliutą, o nesivadovaujant atsitiktiniais minėtais Šilerio naratyvais, reikėtų apžvelgti daug šių duomenų, kas dažnai yra net fiziškai neįmanoma. Vienas iš esančių sprendimų yra naudoti duomenų gavybą.

Mokslas iš didelio kiekio duomenų išskiriantis naudingus šablonus (angl. *pattern)* vadinamas duomenų gavybą (angl. *data mining)*. Duomenų gavyba yra plati ir tarpdisciplininė sritis apimanti dirbtinį intelektą, mašininį mokymąsi (angl. *machine learning*) ir natūralios kalbos apdorojimą (angl. *natural language processing).* Šis procesas tipiškai susideda iš šešių dalių [28]: duomenų valymo, duomenų integravimo, duomenų transformavimo, šablonų atradimo, šablono gilesnės analizės ir jo validavimo, bei atrastų šablonų pristatymo (žr. 2 pav.). Šablonas yra naudingas, jei jo dėsningumas pasireiškia ir testuojamuose duomenyse, bei yra lengvai suprantamas žmogui. Aptarsime duomenų gavybos panaudojimą bitkoino kontekste.



2 pav. Duomenų gavybos veikimo schema

1.5.1. Sentimentų analizė

Žmonės kurie turi bendrų interesų linkę komentuoti susijusiose forumo temose. Bitkoinas dažniausias prekiaujamas vartotojams padarant pirkimo/pardavimo sprendimą remiantis prieinama informacija internete. Kita vertus įmanoma stebėti

Sentimentų analizė yra mokslas tiriantis išreikštas žmogaus nuomones, įsitikinimus ir emocijas tekstuose. Dažniausiai ši analizė, dar vadinama nuomonių gavyba, atliekama poliškumo klasifikacija – tekstui ar sakiniui priskirią teigiamą, neutralią ar neigiamą žymę. Teksto poliškumui nustatyti naudojami du metodai. Pirmuoju metodu pagal nustatytas taisykles ir sudarytus žodynus analizuojami teksto žodžių emocinis stiprumas, antruoju – pažymėti tekstai atitinkamu sentimentu modeliuojami mašininio mokymu pagrįstomis technikomis.

Ko gero pirmasis tyrėjas bandęs rasti sentimentus susijusius su bitkoinu yra L. Kristoufekas, akademikas analizavo sąryšį tarp bitkoino kainos ir kriptovaliutų domėjimusi internete [15], kurį atspindi internautų „Google“ tendencijų užklausų skaičius apie bitkoiną ir aplankytų vartotojų skaičius Vikipedijos bitkoino puslapyje. Tyrime rasta stipri koreliacija tarp kainos ir abiejų domėjimosi indikatorių, taip pat dvikryptis kainos dinamikos ir „Google“ užklausų priežastingumas t.y. ne tik užklausos įtakoja kainos kitimą, bet ir kainos kitimas įtakoja bitkoino užklausų skaičių. Tyrime taip pat išskirtos keletas kainos dinamikos taisyklių: jeigu bitkoino kaina yra aukščiau jos

laiko eilutės trendo ir domėjimasis juo tebeauga, tai kaina taip pat toliau kils; jeigu kaina yra žemiau trendo ir žmonių interesų indikatorius auga, tai bitkoino kaina kris.

J. Kaminskis [29] tyrė socialinio tinklo „Twitter“ žinučių susijusių su bitkoinu sentimentus. Žinutes pagal penkiolikos žodžių sudarytą leksikoną buvo klasifikuojamos į pozityvias, negatyvias arba susijusios su nežinia (angl. *uncertainty*), kurią atspindėjo žodžiai: viliuosi, bijau, jaudinuosi. Kaminskis ištyrė, kad neigiamos ir nežinią atspindinčio žinutės turi vidutinę teigiamą koreliaciją su bitkoino transakcijų skaičiumi ir neigiamą su kriptovaliutos kaina. Panašiai kitame publikuotame straipsnyje [30] buvo analizuojama „Twitter“ pranešimų sentimentai. Tyrime įvertinus žinučių teigiamus, neigiamus ir neutralias sentimentus ir panaudojus Bernoulli Naive Bayes klasifikatorių buvo teisingai klasifikuota 59% valandinių bitkoino kainos kitimų. „Twitter“ tekstinių pranešimų sentimentus taip pat tyrinėjo E. Stenkvista ir J. Lonas [31], taikydami VADER modelį tyrėjai ieškojo sentimentų pokyčių 2,27 milijonuose žinutėse. Remiantis šiais pokyčiais 30 minučių intervaluose bitkoino kainos kitimas buvo prognozuotas 79% tikslumu.

Keliuose tyrimuose buvo koncentruotasi į internetinio forumo komentarų sentimentus. Viename straipsnyje [32] tyrėjų grupė ieškojo komentarų emocijų sąryšių su bitkoinu ir internetiniuose žaidimuose esančiomis skaitmeninėmis valiutomis. Tyrėjai analizuodami teksto semantiką sudarė tokių emocijų kaip džiaugsmas, pyktis, baimė, staigmena, liūdesys, pasišlykštėjimas, tikėjimas, anticipacija ir neutralumas indikatorius. Siurprizo, tikėjimo ir baimės emocijos tyrime buvo atrinktos kaip įtakojančios bitkoino kainos svyravimus, o geriausias SVM (angl. *support vector machines*) modelis naudojantis šiuos sentimentus prognozavo bitkoino kainos kitimus 76,1% tikslumu. Kitame darbe [33] bitkoino komentarus VADER modeliu klasifikavo į penkias grupes: labai pozityvus, pozityvus, neutralus, neigiamas ir labai neigiamas. Sentimentais AODE (angl. *averaged one-dependence estimators*) (žr. formulę (1)) modeliu buvo prognozuojamas bitkoino kainos ir transakcijų kitimas.

$$\hat{P}(y|x_1,\dots,x_n) = \frac{\sum_{i:1\le i\le n \wedge F(x_i)\ge m} \hat{P}(y,x_i)\prod_{j=1}^{n}\hat{P}\left(x_j|y,x_i\right)}{\sum_{y'\in Y}\sum_{i:1\le i\le n \wedge F(x_i)\ge m} \hat{P}(y',x_i)\prod_{j=1}^{n}\hat{P}\left(x_j|y',x_i\right)} \quad (1)$$

Tyrime rastas stiprus sąryšis tarp teigiamą sentimentą atspindinčių komentarų ir bitkoino kainos, o geriausias sukonstruotas modelis prognozavo bitkoino svyravimus 79,5% tikslumu.

1.5.2. Temų modeliavimas

Vienas iš dažniausiai taikomų duomenų gavybos metodų yra temų modeliavimas (toliau TM) (angl. *topic modelling*). Metodas apibrėžia vyraujančias temas, joms aktualiais žodžiai, didelės apimties dokumentuose (angl. *corpus*). Šios technikos privalumas yra tai, kad suteikia galimybę

apžvelgti paplitusias temas dokumentuose be dokumentų skaitymo, kas yra daug laiko reikalaujantis procesas.

Keli tyrėjai sukūrė įvykių atpažinimo algoritmą [34], kuris duotame dideliuose tekstiniuose duomenyse panaudoja TM ir sukuria temas atspindinčias įvykius. Dokumento autoriai tyrimui atlikti panaudojo naujienų agentūros „Rueters“ naujienas ir socialinio tinklo „Twitter“ žinutes. Sukurtas algoritmas pirmiausiai sudaro kasdienines LDA (angl. *Latent Dirichlet allocation*) modelio temas (klasterius), po šio žingsnio apskaičiuoja sudarytų dieninių temų kosinusinius panašumus (angl. *cosine similarity*) ir remdamasis šiais panašumais „Bump-detection“ algoritmu identifikuoja įvykius. Darbe pateikiamas pavyzdys, kaip algoritmas atpažina 2014m. bitkoino keityklos „Mt. Gox“ bankrutavimą, kuris įvyko dėl apie 850 000 dingusių bitkoinų (kas dabar daugiau nei 6 mlrd. JAV dolerių). Pasak tuometinio keityklos savininko Marko Kapelo [35] bitkoinai buvo pavogti įsilaužėlių. „Mt. Gox“ tuo metu valdė apie 70% bitkoino biržos ir toks įvykis nusmukdė bitkoino kainą nuo 17,01 iki 0,01 JAV dolerio [36] .

Panašus algoritmas [36] buvo sukurtas identifikuoti su kriptovaliutomis susijusius įvykius, tokius kaip naują trendą skaitmeninėje valiutoje, ekonominius neramumus ar sukčiavimą kriptovaliutų rinkoje. Algoritmas pagrįstas dinaminiu temų modeliu (toliau DTM) , kuris buvo apmokomas naudojantis bitcointalk.org forumo komentarais. Modelis yra LDA algoritmo atmaina, kuriame atsiranda laiko dimensija, todėl duomenyse paslėptoms temoms leidžia evoliucionuoti ir kisti laike. Sukurtas įvykių atpažinimo algoritmas buvo testuojamas su 37 didžiausiais bitkoino įsilaužimais ar apgaulėmis [37], kurių metu pasisavinta piniginė vertė buvo didesnė negu 1 000 bitkoinų. Programos iš minėtų įvykių neatspindėjo septynių nusikalstamų veikų DTM sumodeliuotose temose.

Dar vienas įdomus darbas buvo pristatytas Pietų Korėjos akademikų [38]. Tyrėjai pasitelkdami duomenų gavybą internetinio forumo bitcointalk.org duomenyse prognozavo bitkoino kainos ir transakcijų svyravimus. Vienas iš darbo tikslų buvo iš komentarų išgauti raktinius žodžius, kurie padėtų identifikuoti reikšmingus komentarus. Šiai užduočiai buvo panaudotas TM metodas neneigiamos matricos faktorizavimas (angl. *non-negative matrice factorisation*). Gautus komentarų temų reitingus, forumo komentarų statistikas (forumo temų skaičius, komentarų skaičius, komentarų, peržiūrėjimo skaičius), bei „Google“ tendencijos ir Vikipedijos indikatorius sudėjo į neuroninį tinklą bitkoino kainos ir transakcijų svyravimus prognozuoti. Geriausias apmokytas neuroninis tinklas prognozavo daugiau negu 80 % tikslumu.

2. TYRIMO METODAI IR PROGRAMINĖ ĮRANGA

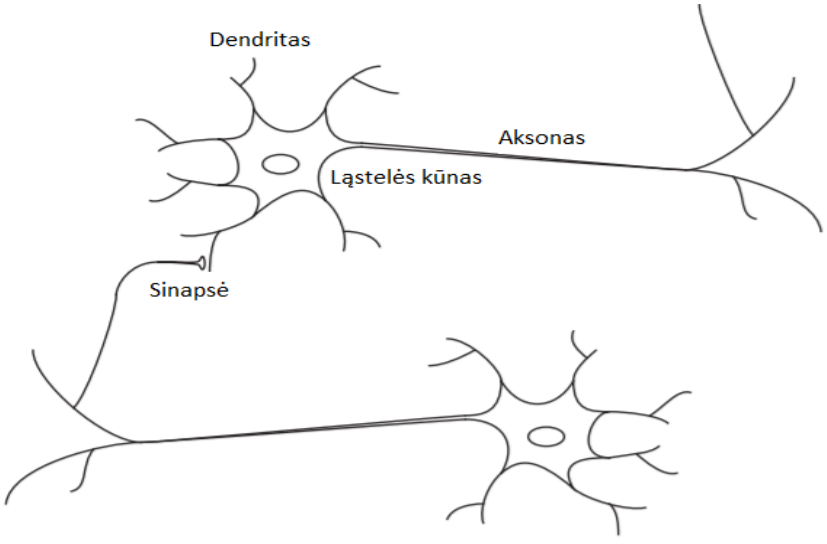
2.1. Neuroniniai tinklai

Šiame skyrelyje apžvelgsime kas yra neuroniniai tinklai ir jų veikimo principą. Jie buvo modeliuojami remiantis žmogaus nervine sistema, todėl pateiksime ir biologinės nervų sistemos neurono veikimo modelį.

2.1.1. Biologinis neuronas

Žmogaus nervų sistema – labai sudėtingas neuronų tinklas, atsakingas už vidaus organų veiklą ir emocijas. Jos pagrindinis struktūrinis ir funkcinis vienetas yra neuronas (nervinė ląstelė). Neuronai yra specifinės ląstelės, galinčios generuoti elektrocheminį signalą, dėl kurių galime galvoti, identifikuoti objektus ir pritaikyti asmeninę patirtį atliekant tam tikras užduotis. Neuronas savo funkcijoms atlikti turi tris pagrindines komponentes (žr. 3 pav.):

- dendritus – išsišakojusią įėjimo struktūrą, kurie priima elektrocheminį signalą iš kito neurono ir perneša jį į ląstelės kūną – soma;
- somą – pagrindinę ląstelės dalį, kurioje yra išsidėstę branduolys ir kitos ląstelės organėlės. Somoje yra apdorojami gauti signalai ir generuojami nauji;
- aksoną – išsišakojusią išėjimo struktūrą, kuri iš neurono kūno informaciją savo ataugėlėmis – sinapsėmis, kurios sujungtos su kito neurono dendritais, perduoda kitiems neuronams.

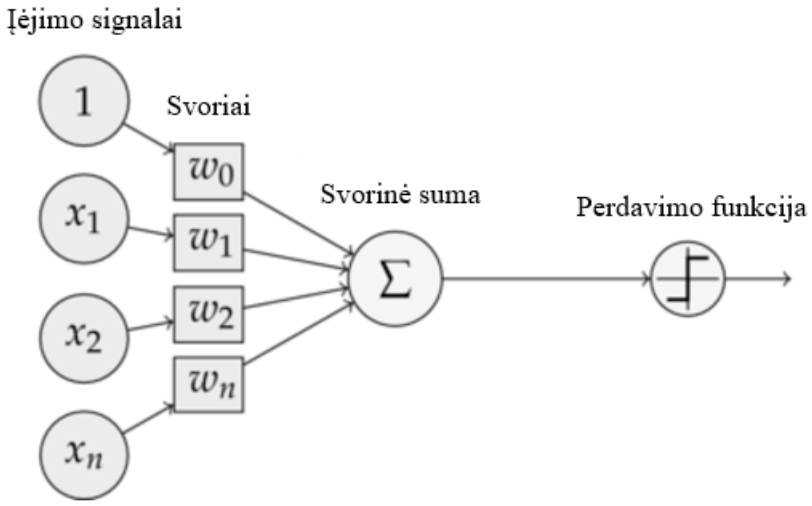


3 pav. Neurono struktūra [39]

Dėl neuronuose vykstančių kompleksiškų cheminių procesų, signalas besijungiančių neuronų gali būti neperduotas, t. y., nesužadinamas neuronas, į kurį siunčiamas signalas. Neurono tikimybė būti sužadintam priklauso nuo sinapsių perdavimo efektyvumo, kuris veikia signalo stiprumą. Jei dendritais gautas signalas viršija tam tikrą lygį (sužadinimo slenkstį), neuronas sužadinimas ir jis duoda elektrinį impulsą kitiems neuronams. Manoma, kad sinapsių efektyvumas keičiasi vykstant mokymosi procesui.

2.1.2. Dirbtinis neuronas

Dirbtiniai neuroniniai tinklai nėra tokie kompleksiški kaip žmogaus nervinė sistema, bet buvo modeliuojami siekiant išlaikyti analogišką funkcionalumą. Neuronas dirbtiniuose tinkluose, analogiškai kaip ir biologinėje sistemoje, turi keletą įėjimo signalų. Tai gali būti neuroninio tinklo įėjimo signalai arba kitų dirbtinių neuronų išėjimo signalai. Kiekviena įėjimo jungtis turi priskirtą stiprumo koeficientą – svorį, kuris atitinka biologinių neuronų sinapsių funkcionalumą, jei jis teigiamas – neuronas turi žadinamąjį efektą, jei jis neigimas – slopinamąjį efektą. Visi neuronai dirbtiniuose neuronų tinkluose turi savo slenksčio vertę, kurios skirtumas su svorine įėjimo signalų suma apibrėžia neurono sužadinimo būseną. Su gautu sužadinimo signalu ir perdavimo funkcija apskaičiuojama neurono išėjimo vertė.



4 pav. Dirbtinio neurono schema

3 pav. $x_1, x_2, \dots, x_n$ žymi neurono įėjimo signalus, $w_1, w_2, \dots, w_n$ pažymėti svoriai. Schemoje w_0 atitinka neurono slenksčio vertę, o svorinė suma s apskaičiuojama pagal formulę:

$$s = w_0 + \sum_{i}^{n} w_i x_i \tag{2}$$

Apskaičiavę svorinę sumą s, skaičiuojamą neurono perdavimo funkciją $f(s)$, pateiksime labiausiai paplitusias funkcijas.

Tiesinė perdavimo funkcija: (c – konstanta)

$$y = s \cdot c \tag{3}$$

Šuolio funkcija: ši funkcija yra biologinio neurono atitiktis, jei suminė suma s lygi arba didesnė už nulį, neuronas sužadinimas – funkcija grąžina 1, priešingu atveju – 0.

$$y = \begin{cases} 1, jei\ s \geq 0 \\ -1, kitu\ atveju \end{cases} \tag{4}$$

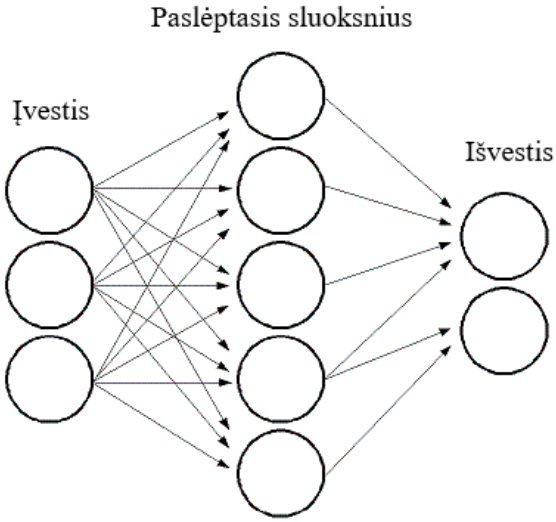
Loginio sigmoido funkcija: ši tolydi funkcija taip pat atspindi biologinio neurono funkcionalumą. Funkcijos reikšmių sritis [0;1] ($f(s)$ artėja prie 0, kai $s \to -\infty$, $f(s)$ artėja prie 1, kai $s \to \infty$)

$$y = \frac{1}{1 + e^{-s}} \tag{5}$$

Hiperbolinio tangento funkcija: panaši į loginio sigmoido funkciją, tik jos reikšmių sritis [-1;1]

$$y = \frac{e^{s} - e^{-s}}{e^{s} + e^{-s}} \tag{6}$$

Norint imituoti žmogaus nervų sistemą aptartas dirbtinis neuronas jungiamas su kitais neuronais, taip vieno neurono išėjimo signalas tampa kitų neuronų įėjimo signalu. Tokiu principu sukonstruotas tinklas gali aproksimuoti netiesines funkcijas. Praktikoje neuronai dažniausiai grupuojami į sluoksnius. Pirmasis sluoksnis būna duomenų įvesties, paskutinis – išvesties, o sluoksniai tarp jų – paslėptieji (angl. *hidden layers*) (žr. 5 pav.).



5 pav. Dirbtinio neurono tinklo schema

2.1.3. Ilgos-trumpos atminties rekurentiniai neuroniniai tinklai

LSTM (angl. *long-short term memory*) yra S. Hochreiterio ir J. Schmidhuberio [40] pasiūlyti rekurentiniai neuroniniai tinklai gebantys išmokti ilgo laikotarpio priklausomybės. Ši savybė prisiminti praeities dėsningumus ir pritaikyti reikiamu metu LSTM neuroninius tinklus išskiria iš kitų modelių, todėl jie yra plačiai taikomas daugelyje sričių: šnekamosios kalbos atpažinime, tekstų generavime, nuotraukų antraščių formavime, kalbų modeliavime ir vertime ir kt.

LSTM veikimo principas pagrįstas matrica C_t (žr. 6 pav.), dar vadinama ląstele, kuri padeda išsaugoti minėtąsias priklausomybes. Ląstelė C_t laikui bėgant kinta pagal formulę

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t, \tag{7}$$

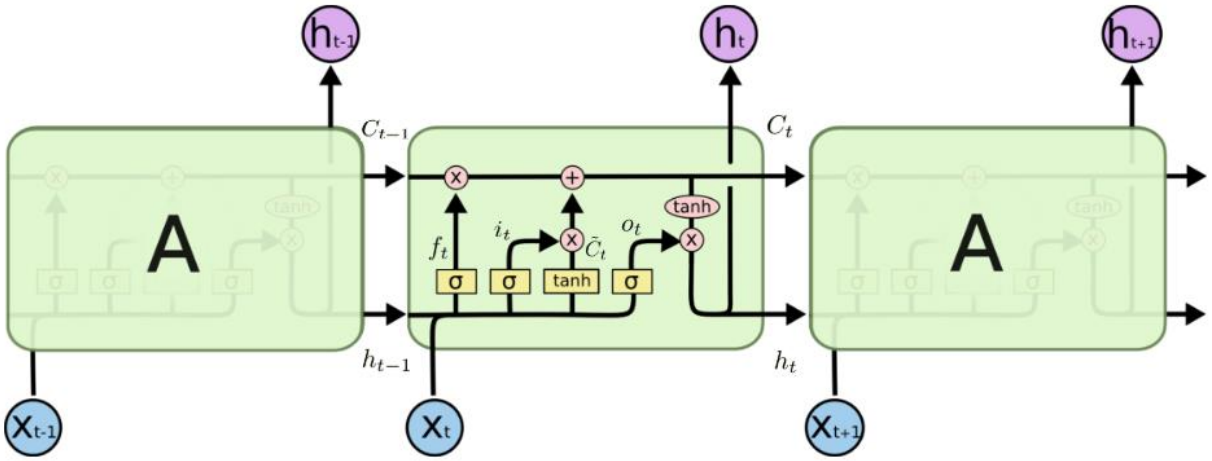
čia f_t yra matrica dar vadinama „pamiršimo vartais“, kuri iš ląstelės pašalina nereikalingą informacija ir yra apskaičiuojama pagal formulę

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{8}$$

(7) formulėje $i_t * \widetilde{C_t}$ yra ląstelės C_t atnaujinimo matrica, kurios komponentės išreiškiamos formulėmis

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{9}$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{10}$$



6 pav. LSTM neuroninių tinklų schema [41]

Tuomet LSTM išeities vertė h_t apskaičiuojama formulėmis

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{11}$$

$$h_t = o_t * tanh(C_t) \tag{12}$$

Formulėse (8)(9)(11) σ žymima anksčiau aprašyta loginio sigmoido funkcija (5), o (10)(12) formulėse žymima $tanh$ yra hyperbolinio tangento funkcija (6).

2.2. Dinaminis temų modeliavimas

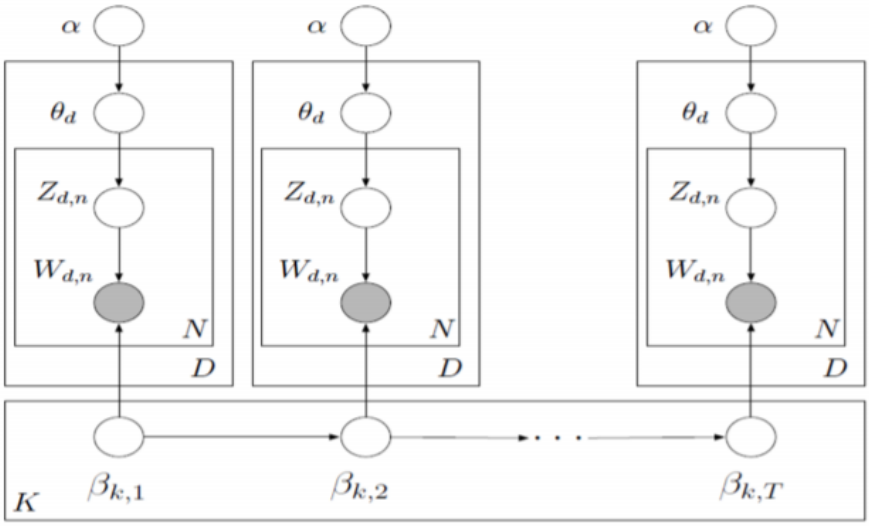
DTM yra LDA modelio plėtinys, kuris priešingai negu kiti TM metodai įtraukia laiko dimensiją t.y. modelis laviruoja tarp skirtingais laiko pjūviams priklausantiems dokumentams ir ieško kaip žodžiai laikui bėgant kinta temose.

Pateiksime [42] aprašytą DTM temų generavimo procesą:

1. Įtraukti temą $\beta_t|\beta_{t-1} \sim \mathcal{N}(\beta_{t-1}, \sigma^2 I)$
2. Įtraukti $\alpha_t|\alpha_{t-1} \sim \mathcal{N}(\alpha_{t-1}, \delta^2 I)$
3. Kiekvienam dokumentui:
    a) Įtraukti $\eta \sim \mathcal{N}(\alpha_t, a^2 I)$
    b) Kiekvienam žodžiui dokumente:
        i. Įtraukti temos priskyrimą $Z \sim Mult(\pi(\eta))$
        ii. Įtraukti žodį $W_{t,d,n} \sim Mult(\pi(\beta_{t,z}))$

Čia π funkcija išreiškiama formule:

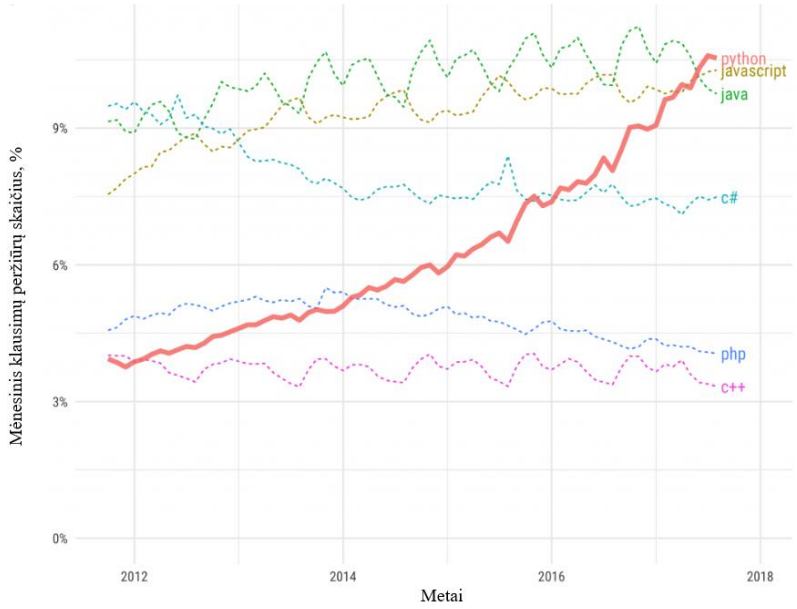
$$\pi\left(\beta_{k,t}\right)_w = \frac{e^{\beta_{k,t,v}}}{\sum_w e^{\beta_{k,t,v}}} \tag{13}$$



7 pav. DTM erdvės diagrama [42]

2.3. Programinė įranga

Šiame tyrime buvo naudojama Python programavimo kalba. Ši aukšto lygio programavimo kalba yra dinaminė ir objektiškai orientuota, kurios sintaksė yra labai lengvai skaitoma ir paprasta. Kalba yra populiarėjanti, jo augimą atskleidžia www.stackoverflow.com užduotų klausimų apie programuotojų iškilusias problemas peržiūrų skaičius [43] (žr. 3 pav.).



8 pav. Stackoverflow klausimų peržiūrų pasiskirstymas laike [43]

Pagal kaggle 2017m. apklausą [44] Python yra labiausiai naudojama duomenų mokslininkų (angl. *data scientists*) programavimo kalba. Daugiausiai korespondentų duomenų mokslininkui rekomenduoja, iš visų programavimo kalbų, išmokti kaip pirmąją. Python tyrimui buvo pasirinkta, nes yra lengva programavimo kalba ir turi naudingų bibliotekų skirtų duomenų analizei:

Numpy: C++ kalba parašyta biblioteka skirta efektyviems skaičiavimais su vektoriais ir n-mačiais masyvais.

NTLK (angl. *Natural Language Toolkit*): nemokamas natūralios kalbos įrankių kompleksas skirtas dirbti su tekstu. Bibliotekoje yra gausu daiktavardžių, būdvardžių, prieveiksmių sinonimų, antonimų ir kitų žodynų, taip pat gausu įrankių tekstui apdoroti tokių kaip tokenizavimas, lematizavimas, žymių tvirtinimas, gramatinis nagrinėjimas ir kt.

Gensim: atvirojo kodo biblioteka skirta temų modeliavimui ir jo analizei.

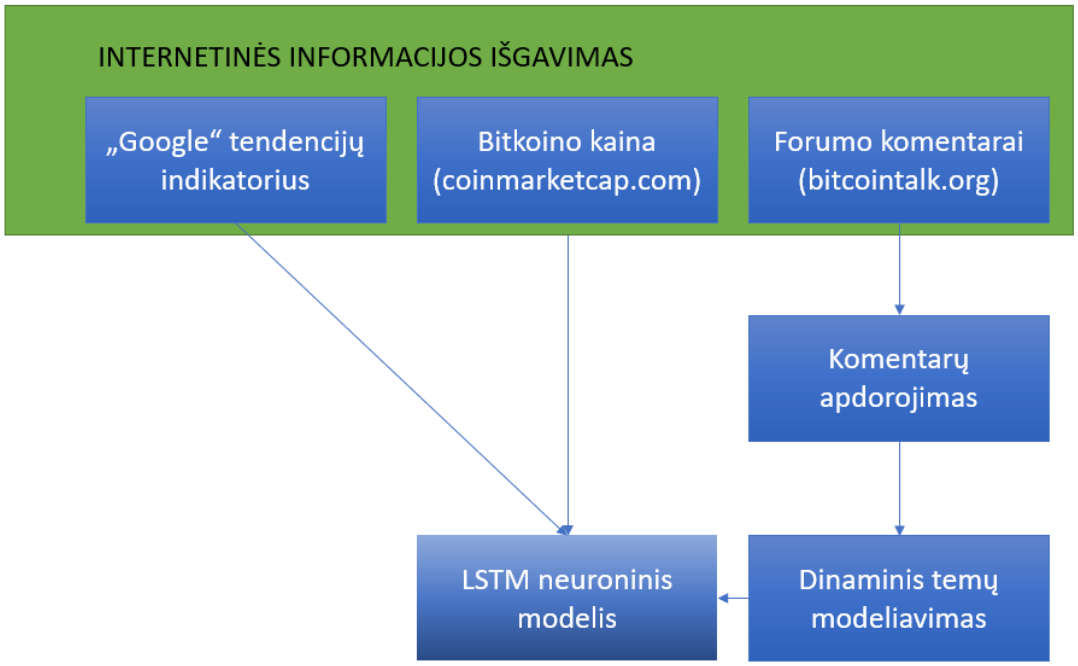
Tensorflow: šiame darbe buvo naudotas mašininio mokymo programinė įranga TensorFlow. TensorFlow yra Google sukurta atviro kodo variklis skirtas atlikti efektyvius skaičiavimus su „tensoriais“ (angl. *tensor*). Tensoriai yra duomenų struktūros, pvz. vienos dimensijos tensorius yra vektorius, dviejų – matrica, trijų – trimatė matrica ir t.t. Ši programinė įranga leidžia atlikti skaičiavimus su grafiniu procesoriumi, kuris naudoja NVIDIA parašytą biblioteką cuDNN, vietoje įprastai naudojamo centrinio procesoriaus. Priklausomai nuo vaizdo plokštės skaičiavimo proceso trukmės atlikimas dažniausiais skiriasi ne procentais, o kartais. Ko gero dėl šios savybės Tensorflow karkasas yra populiariausias tarp visų giliojo mokymo bibliotekų [45] ir pamažu tampa standartu neuroninių tinklų ir dirbtinio intelekto modeliavime.

Dar vienas Tensorflow privalumas yra, jog programuoti su šia programine įranga galima C, C++, Java, Go, Rust, R, Haskell, Julia ir Python programavimo kalbomis. Pasirinkimas yra tikrai didelis, bet yra rekomenduojama naudoti Python programavimo kalbą, nes jei pirmajai buvo pritaikytas Tensorflow ir todėl jos aplinkoje bus didžiausias galimas funkcionalumas.

Tensorflow susilaukus tokiam populiarumui atsirado naujos ar buvo atnaujintos jau egzistuojančios bibliotekos (tokios kaip Keras, Theano, Sonnet ir kt.) leidžiančios patogiau ir greičiau programuoti naudojant Tensorflow karkasą. Šiame tyrime buvo naudojama Keras biblioteka skirta patogiau modeliuoti neuroninius tinklus.

3. TYRIMŲ REZULTATAI IR JŲ APTARIMAS

Tyrimas gali būti suskirstytas į tris pagrindines dalis (žr. 9 pav.): duomenų išgavimą, dinaminį temų modeliavimą ir kainos prognozavimą naudojantis LSTM neuroninius tinklus. Tyrime naudota internetinė informacija išgauta iš trijų šaltinių: „Google“ tendencijų programinė sąsaja (toliau API (angl. *application programming interface*)), kriptovaliutų kainų portalo www.coinmarketplace.com ir internetinio forumo apie kriptovaliutas bitcointalk.org. Didžiausias dėmesys buvo skirtas komentarams, kurie buvo programiškai apdoroti. Apdorotiems komentarams buvo pritaikytas DTM surasti vyraujančias temas (klasterius) atspindinčias internetines diskusijas tirtame laikotarpyje. Pagal surastas temas buvo sukurti rodikliai indikuojantys temos populiarumą kiekvieną dieną. Išgautais duomenis ir DTM temų indikatoriais LSTM neuroniniu tinklu buvo prognozuojamas bitkoino kaina.



9 pav. Tyrimo schema

3.1. Internetinė informacija

Šiais laikais internetas tapo įprasta žinių paieškos ir bendravimo priemonė. Duomenys, kaip anksčiau minėta, auga eksponentiškai, tad dabar sunkesnis uždavinys dažnai tampa juos atsifiltruoti, o nesusirasti. Taip pat ir su informacija apie bitkoiną, jos pilna spaudoje, socialiniuose tinkluose ir kitur. Apžvelgsime tyrime naudotą internetinę informaciją ir jos duomenų šaltinius.

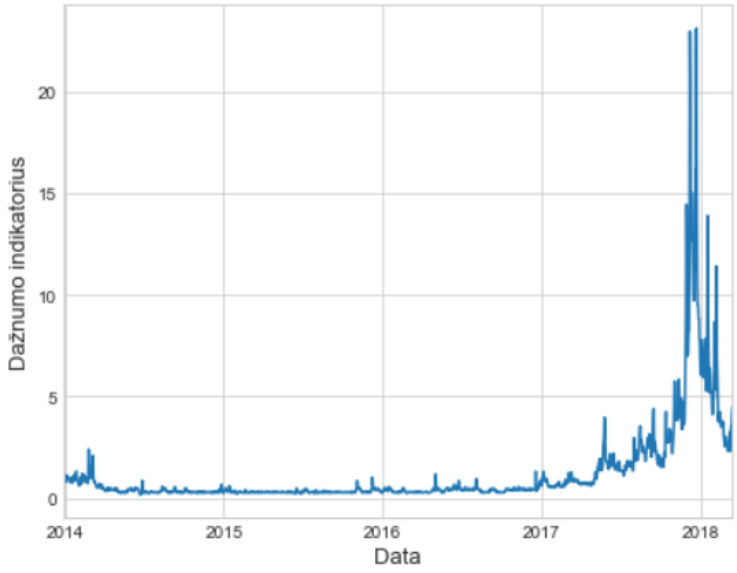
3.1.1. Bitkoino kaina

Iš internetinė svetainės https://coinmarketcap.com *pandas* biblioteka buvo ištraukti duomenys apie bitkoino atidarymo, uždarymo, dienos aukščiausias ir žemiausias kainas, bitkoinų transakcijų

skaičius ir kapitalizacija. Tyrime buvo naudojamos atidarymo (angl. *Open*) ir uždarymo (angl. *Close*) kainos.

3.1.2. Google tendencijos

„Google tendencijos“ (*angl. Google trends)* yra įrankis skirtas „Google“ žiniatinklio paieškoms analizuoti. Įrankis grąžina ar vizualizuoja pageidautinos užklausos santykinį dažnumo indikatorių nurodytame laiko intervale su galimai pasirinktu regionu ar kalba. Tyrime naudojama „bitcoin“ užklausos dieninis dažnumo indikatoriaus istorija 2014m. sausio 1d. - 2018m. kovo 15d. laiko intervale (žr. 10 pav.).



10 pav. „Google“ paieškos „bitcoin“ dažnumo indikatoriaus pasiskirstymas laike

Duomenis gauti naudojant „Google“ tendencijų API, tiesa „Google“ tendencijos grąžina norimos užklauso dieninę laiko eilutę, kai laiko intervalas nedidesnis negu 269 dienos, kitu atveju laiko eilutės reikšmės pateikiamos savaitėmis. Kadangi mūsų analizuojamame laiko intervale indikatorius G_s gražinamas savaitėmis, buvo parašyta funkciją skirta apskaičiuoti dieninės laiko „Google“ t dienos tendencijų indikatorių G_t pagal formulę

$$G_t = G_{s_t} \frac{G_{pt}}{G_{ps}}, \tag{14}$$

čia G_{s_t}- savaitinis „Google“ tendencijų indikatorius (atitinkantis kalendorinę savaitę kurioje yra t diena), G_{pt} – t dienos „Google“ tendencijų indikatorius, G_{ps} - t dienos savaitinis „Google“ tendencijų indikatorius gautas susumavus dieninę G_{pt} laiko eilutę atitinkamoje savaitėje. G_{pt} laiko eilutė yra gauta sujungus gražinamas laikos eilutes iš siųstų užklausų „Google“ tendencijų API. Užklausos buvo formuojamos kas 181 dienų iki duoto intervalo pabaigos. Pirma užklausa buvo

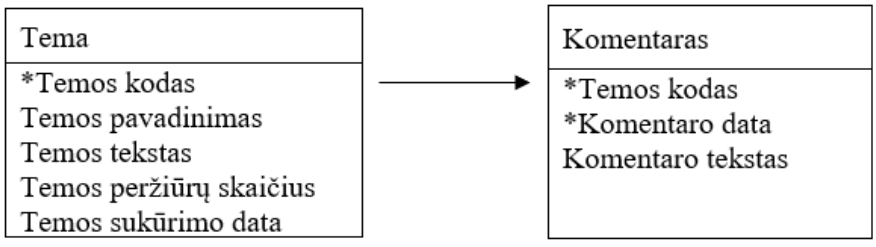
suformuota nuo intervalo pradžios pirmo sekmadienio (nes G_{s_t} savaitinės reikšmės skaičiuojamos nuo sekmadienio). Tokiu būdu ir kitos užklausos prasidėdavo nuo sekmadienio. Toks užklausų formavimas užtikrina, kad sujungus kelias laiko eilutes G_{pt} dieninės reikšmės esančios atitinkamoje savaitė bus tik iš vienos užklausos ir skaičiuojant G_t laiko eilutę rezultatų neiškreips.

3.1.3. Internetinio forumo komentarai

Duomenys gauti iš vieno didžiausio kriptovaliutų internetinio forumo https://bitcointalk.org. Jame internetiniai vartotojai aktyviai įsitraukia į kriptovaliutų diskusijas komentuodami ar kurdami naujas forumo temas. Svetainėje internetinės diskusijos yra sugrupuotos į keturias grupes: „Bitkoinas“, „Ekonomika“, „Kitkas“, „Alternatyvios kriptovaliutos“. Kiekviena grupė taip pat padalinta į tris-penkias sekcijas, pavyzdžiui Bitkoinas turi penkias skiltis: „Bitkoino diskusijos“, „Plėtojimas. ir techninės diskusijos“, „Kasimas“, „Bitkoino techninis aptarnavimas“, „Projekto vystymasis“.

Visi tyrime naudoti komentarai ištraukti iš aktyviausios skilties „Bitkoino diskusijos“ pasinaudojus [38] darbe naudotu programos kodu. Programa duomenys „kasa“ nuo naujausios forumo temos (kurioje parašytas paskutinis komentaras arba nuo nurodyto temų puslapio ir kas keturiasdešimt temų išsaugo duomenys į atskirą *json* formato failą. Faile duomenys yra sugrupuoti temomis. Kiekviena temos grupė turi penkis informacinius punktus: temos sukūrimo data, forumo temos pavadinimas, forumo temos kūrėjo pradinis tekstas, forumo peržiūru skaičius ir temos komentarų (replikų) sąrašas. Komentarų sąraše, kiekvienam komentarui yra priskirta jo parašymo data.

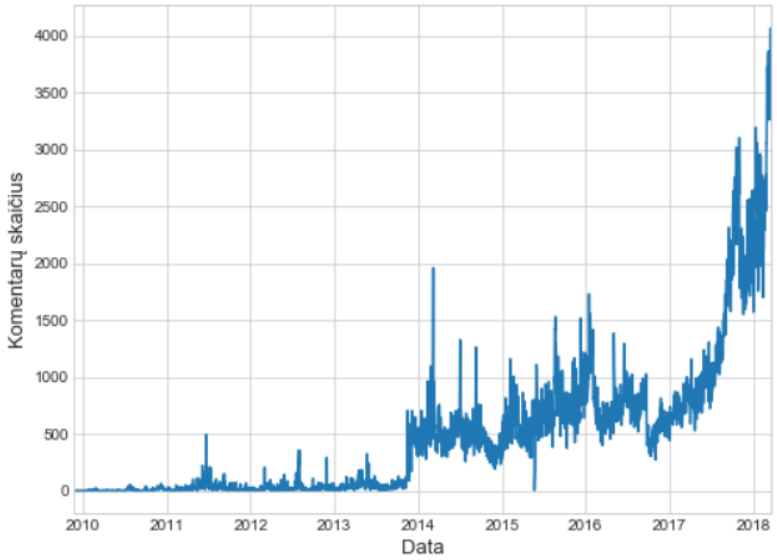
Tyrime naudoti forumo komentarai yra iškasti per keletą kartų, taip pat sujungti su [38] tyrime išgautais duomenimis. Kadangi informacija tarp kasimų yra persidengusi (yra vienodų įrašų), turimų failų sujungimui buvo panaudotas reliacinis duomenų bazės modelio principas. Naudojamas duomenų reliacinis modelis pavaizduotas 11 paveikslėlyje. Duomenų modelyje * pažymėti įrašo indeksai.



11 pav. Komentarų reliacinis modelis

Remiantis reliaciniu modeliu visoms turimoms forumo temoms buvo priskirtas unikalus kodas. Tuomet visi turimi temos duomenys, išskyrus komentarai, buvo transformuojami į *pandas* bibliotekos duomenų lentelę ir išsaugoti viename faile. Forumo komentarai buvo saugomi atskirose *pandas* duomenų matricose, kuriose komentarai buvo indeksuojami pagal temos unikalų kodą. Kiekvieno failo komentarai buvo iteruojami pagal komentaro parašymo datą ir pagal ją komentarų matricos buvo išsaugomos į atskirus duomenų failus (duomenų failo vardai buvo pavadinami komentarų parašymo data). Jeigu išsaugant komentarų matricą duomenų failas jau egzistuodavo, jis buvo papildomas naujais unikaliais komentarais. Tokiu būdu buvo pašalinti visi besikartojantys komentarai.

Aprašyta metodika išviso išgauta 1 430 884 unikalių komentarų 2009 m. lapkričio 22d. - 2018m. kovo 15d. laiko intervale. Jų pasiskirstymas pavaizduotas 12 paveikslėlyje. Paveikslėlyje matyti, kad žymesnis komentarų augimas prasidėjo 2013m. pabaigoje, todėl buvo nuspręsta tyrimui naudoti duomenys nuo 2014m. sausio 1 d., atfiltravus nereikalingus duomenis liko 1 375 223 komentarai.



12 pav. Bitkoino forumo https://bitcointalk.org komentarų pasiskirstymas laike

3.2. Dinaminis temų modeliavimas

Prieš atliekant duomenų dinaminį temų modeliavimą reikia atlikti kelis žingsnius: sudaryti reikšmingų žodžių žodyną ir paruošti duomenis modeliavimui.

3.2.1. Žodynas

Pirmas atliktas žingsnis sudarant žodyną yra reguliariųjų išraiškų panaudojimas. Pavertus visas komentarų raides į mažąsias reguliariosiomis išraiškomis visi internetinės svetainės adresai

buvo pakeisti žodžiu *web_url,* žodis *e-mail* paverstas į *email* ir tik alfabetinės raidėmis prasidedantys žodžiais buvo palikti, tokiu būdu visi skyrybos ženklai ir skaičiai buvo pašalinti iš teksto. Atlikus šiuos žingsnius, komentarai buvo išskaidomi į atskirus žodžius ir juose šalinami bereikšmiai žodžiai (angl. *stop words)*, tai bendri jokios informacijos temų modeliavimui nenešantys žodžiai, pvz., *i, me, we, you, was, were, would, should, other, at, of* ir kt.

Tolimesnis atliktas žingsnis buvo žodžių lematizavimas (angl. *lemmatization)*. Tai labai svarbi teksto apdorojimo dalis, kuri randa analizuojamo žodžio pradinę formą (angl. *lemma*). Tyrime buvo naudojama *ntlk* paketo *WorldNet* lematizatorių, kuriuo galima gauti žodžio daiktavardžio, būdvardžio, veiksmažodžio ir prieveiksmio pradinės formas, pvz. angliškų žodžių *am, are, is* pradinė žodžio forma nurodžius veiksmažodį būtų *be*.

Vėliau buvo panaudotas n-gramų modelis. Modelis duotame tekste suskaičiuoja visas gretimas žodžių poras ir sujungia tas žodžių poras, kurių dažnis viršija nustatytą slenkstį. Pirmąjį kartą pritaikius modelį gauname bigramus, bet procesą galima tęsti ir sujungti bigramus su reikšmingai pasikartojusiu gretimu žodžiu ir gauti trigramus ir t.t. Tyrime buvo naudotas bigramų modelis, 20 dažniausiai pasitaikiusių bigramų pateikti 1 lentelėje.

1 lentelė. Dvidešimt dažniausių bigramų tirtuose komentaruose

Bigramas	**Dažnis**	**Bigramas**	**Dažnis**
private_key	34 431	*paper_wallet*	12946
digital_currency	31 565	*social_medium*	12908
long_term	29 435	*satoshi_nakamoto*	12203
signature_campaign	25 138	*blockchain_technology*	12053
block_size	21 951	*small_amount*	11286
credit_card	20 014	*near_future*	10954
hard_fork	16 829	*blockchain_info*	10386
year_ago	14 445	*payment_method*	9797
alt_coin	13 381	*block_chain*	9578
make_sense	13 198	*market_cap*	9520

Iš gautų bigramų sudarytas žodynas. Jame kiekvienam unikaliam bigramui priskiriama skaitinė reikšmė – raktinis kodas. Žodyne pašalinti žodžiai, kurių dažnumas visose tirtuose komentaruose buvo mažesnis negu penkiolika kartų, po filtravimo liko 38 419 unikalių kodų.

3.2.2. Komentarų paruošimas

Kaip anksčiau minėta, iš viso išgauta 1 375 223 komentarų, bet daugelis iš jų labai trumpi ir teksto temos dažnu atveju neatskleidžia. Dėl to komentarai buvo grupuojami pagal laiką ir forumo

temą, tuomet sugrupuoti komentarai buvo sujungiami į vieną. Taip pat kiekviena forumo tema turėjo įžanginį tekstą, kuris apibūdindavo temos problematiką ar jos aktualijas, šis tekstas taip pat buvo prijungiamas prie pirmo temos parašyto komentaro. Taip iš viso susidarė 128 577 komentarų su skirtingomis 25 515 forumo temomis, toliau šiuos komentarus vadinsime grupuotaisiais.

Kaip ir žodyno sudarymo metu, komentarams buvo pritaikytos tos pačios procedūros apdoroti tekstą. Gavus sugrupuotų komentarų bigramus buvo panaudotas žodynas gauti skaitmeninius žodžių vektorius (angl. *bag of words*). Vektorius, atitinkantis sugrupuotą komentarą, yra sudarytas iš skaitmenų porų (raktas, dažnis), kuriuose raktas yra kodas atitinkantis žodį iš sudaryto žodyno, dažnis – komentare pasikartojęs žodžio kiekis.

3.2.3. Modeliavimas

DTM buvo modeliuotas *gensim* bibltioteka, kurioje reikėjo nurodyti komentarų gautus skaitmeninius žodžių vektorius, temų (klasterių) skaičių ir vektorių su pasirinktais laikotarpių tekstų dažniais. Tyrime pasirinktas DTM temų skaičius - 30, o tiriamasis laikotarpis buvo suskirstytas į septyniolika dalių, kurie atspindėjo metų ketvirčius, jų pasiskirstymas pateiktas 2 lentelėje.

2 lentelė. Sugrupuotų komentarų pasiskirstymas tirtame laikotarpyje

Laikotarpis	**Komentarų skaičius**	**Laikotarpis**	**Komentarų skaičius**
2014 I ketvirtis	6 012	2016 II ketvirtis	6 681
2014 II ketvirtis	5 421	2016 III ketvirtis	6 400
2014 III ketvirtis	5 653	2016 IV ketvirtis	5 232
2014 IV ketvirtis	4 637	2017 I ketvirtis	6 303
2015 I ketvirtis	4 534	2017 II ketvirtis	7 480
2015 II ketvirtis	4 022	2017 III ketvirtis	12 614
2015 III ketvirtis	5 189	2017 IV ketvirtis	18 726
2015 IV ketvirtis	4 988	2018 I ketvirtis	18 490
2016 I ketvirtis	6 195		

3.2.4. Rezultatai

DTM tekstai (komentarai) sugrupuoti į trisdešimt klasterių kiekvieną tiriamą ketvirtį. 3 lentelėje pateikti pirmosios temos dešimt populiariausių žodžių kitimas laike - temos evoliucija. Žodžiai lentelėje pateikti tikimybės mažėjimo tvarka t.y. pirmieji žodžiai labiau indikuoja, kad investavimo vyraujančias pozicijas ir nuomones. Lentelėje matyti, kad 2017 metų visus ketvirčius populiariausi žodžiai yra *bitcoins*, *future* (ateitis) ir *save* (saugoti), kurie gali būti interpretuojami, kaip vartotojų teigiamas sentimentas investuoti į bitkoiną. Šiuo laikotarpiu bitkoino kaina kilo į vis

didesnes aukštumas ir nuo žemiausia kainos iki didžiausios skaitmeninė valiuta paaugo daugiau negu 25 kartus (remiantis coinmarketcap.com duomenimis).

Taip pat matome, kad 2017m. III ketvirtyje tarp dešimt reikšmingiausių temos žodžių atsirado žodis *sell* (parduoti), tuo laiku spauda pradėjo intensyviai rašyti apie daugelio ekspertų nuomonę, kad toks augimas nėra normalus, kad bitkoinas neturi fundamentaliosios vertės ir yra labai išpūstas burbulas. O kai nuo 2017m. gruodžio 17d. bitkoino kaina (apie 19 000 JAV dolerių) pradėjo smukti iki 6 300 JAV dolerių (2018m .vasario 6d.) 2018m. I ketvirčio DTM pirmoje temoje žodis *sell* tapo populiariausiu, kuris indikuoja apie vartotojų baimę ir diskusijas apie bitkoino pardavimą.

3 lentelė. DTM modelio pirmosios temos (klasterio) reikšmingiausi žodžiai

Laikotarpis	**Temos raktiniai žodžiai**
2014 I ketvirtis	*bitcoins, gold, buy, spend, hold, coin, holding, saving, keep, value*
2014 II ketvirtis	*bitcoins, gold, spend, buy, hold, coin, holding, saving, keep, sell*
2014 III ketvirtis	*bitcoins, spend, buy, gold, hold, coin, holding, saving, keep, sell*
2014 IV ketvirtis	*bitcoins, spend, hold, buy, coin, holding, gold, keep, saving, sell*
2015 I ketvirtis	*bitcoins, spend, hold, buy, holding, coin, sell, keep, saving, spending*
2015 II ketvirtis	*bitcoins, hold, spend, buy, holding, sell, coin, keep, saving, spending*
2015 III ketvirtis	*bitcoins, hold, spend, holding, buy, sell, coin, keep, future, investment*
2015 IV ketvirtis	*bitcoins, hold, spend, holding, sell, buy, coin, future, keep, investment*
2016 I ketvirtis	*bitcoins, hold, holding, spend, sell, buy, coin, future, want, investment*
2016 II ketvirtis	*bitcoins, hold, holding, spend, sell, future, buy, coin, want, investment*
2016 III ketvirtis	*bitcoins, hold, spend, future, holding, save, sell, price, want, investment*
2016 IV ketvirtis	*bitcoins, future, save, hold, spend, holding, saving, price, investment*
2017 I ketvirtis	*future, save, bitcoins, saving, hold, spend, holding, need, price, investment*
2017 II ketvirtis	*future, save, bitcoins, saving, hold, need, purpose, fun, want, keep*
2017 III ketvirtis	*future, save, bitcoins, saving, sell, hold, want, plan, keep, need*
2017 IV ketvirtis	*future, bitcoins, sell, want, save, hold, keep, holding, need, saving*
2018 I ketvirtis	*sell, want, bitcoins, future, house, hold, keep, holding, need, profit*

Kita įdomi įžvalga pastebėta dvidešimt devintoje sudarytoje temoje, kuri atspindi diskusijas apie bitkoino kasimą. 4 lentelėje matome, kad 2017 IV ir 2018 I ketvirčių sudarytuose klasteriuose tarp dešimt reikšmingiausių žodžių atsirado žodis *electricity* (elektra). Pasidomėjus elektra ir kasimu šiuo laikotarpiu, radau daug spaudos straipsnių susirūpinusių dėl elektros sunaudojamo kiekio bitkoino sistemos palaikymui. Viename straipsnyje [46] rašoma, kad bitkoino kasėjų atliekami

skaičiavimai sunaudoja vis daugiau elektros ir metų pabaigoje sunaudos 0,5 % viso pasaulio sunaudojamas elektros.

Taip pat žodis minėtose ketvirčiuose galėjo atsirado dėl anksčiau minėto bitkoino kainos smukimo. Bitkoino kainai mažėjant bitkoino kasėjai galimai išreiškė išlaidų didėjimą elektros sąskaita. Pagal [47] straipsnį smulkieji kriptovaliutos kasėjai turi didelę riziką tapti nuostolingais, jei bitkoino kaina nebus aukščiau negu 9 000 JAV dolerių.

4 lentelė. DTM modelio dvidešimt devintosios temos (klasterio) reikšmingiausi žodžiai

Laikotarpis	**Temos raktiniai žodžiai**
2014 I ketvirtis	*mining, miner, pool, network, cost, mine, attack, power, difficulty, block*
2014 II ketvirtis	*mining, miner, pool, network, cost, mine, attack, block, difficulty, power*
2014 III ketvirtis	*mining, miner, pool, network, mine, cost, block, difficulty, attack, power*
2014 IV ketvirtis	*mining, miner, pool, network, mine, cost, block, difficulty, power, attack*
2015 I ketvirtis	*miner, mining, pool, mine, network, cost, block, difficulty, power, electricity*
2015 II ketvirtis	*miner, mining, mine, pool, network, block, cost, difficulty, power, mined*
2015 III ketvirtis	*miner, mining, pool, mine, block, network, cost, power, difficulty, mined*
2015 IV ketvirtis	*miner, mining, pool, mine, block, network, power, cost, difficulty, mined*
2016 I ketvirtis	*miner, mining, pool, mine, block, network, power, cost, difficulty, Chinese*
2016 II ketvirtis	*miner, mining, pool, mine, block, network, power, chinese, cost, reward*
2016 III ketvirtis	*miner, mining, pool, block, mine, network, chinese, power, cost, reward*
2016 IV ketvirtis	*miner, mining, pool, block, segwit, bu, mine, network, power, Chinese*
2017 I ketvirtis	*miner, bu, mining, segwit, pool, block, mine, network, node, power*
2017 II ketvirtis	*miner, mining, segwit, bu, mine, pool, uasf, block, litecoin, bitmain*
2017 III ketvirtis	*mining, miner, mine, segwit, pool, bitmain, power, need, bip, electricity*
2017 IV ketvirtis	*mining, miner, mine, electricity, power, need, hardware, profitable, segwit, difficulty*
2018 I ketvirtis	*mining, miner, mine, electricity, hardware, profitable, power, need, difficulty, much*

Peržiūrėjus visas temas (klasterius) buvo atrinktos aštuonios temos tolimesnei analizei, klasteriai buvo atrinkti pagal juose vyraujančius žodžius. Atrinktų klasterių populiariausi žodžiai pateikti 5 lentelėje.

5 lentelė. DTM atrinkti klasteriai

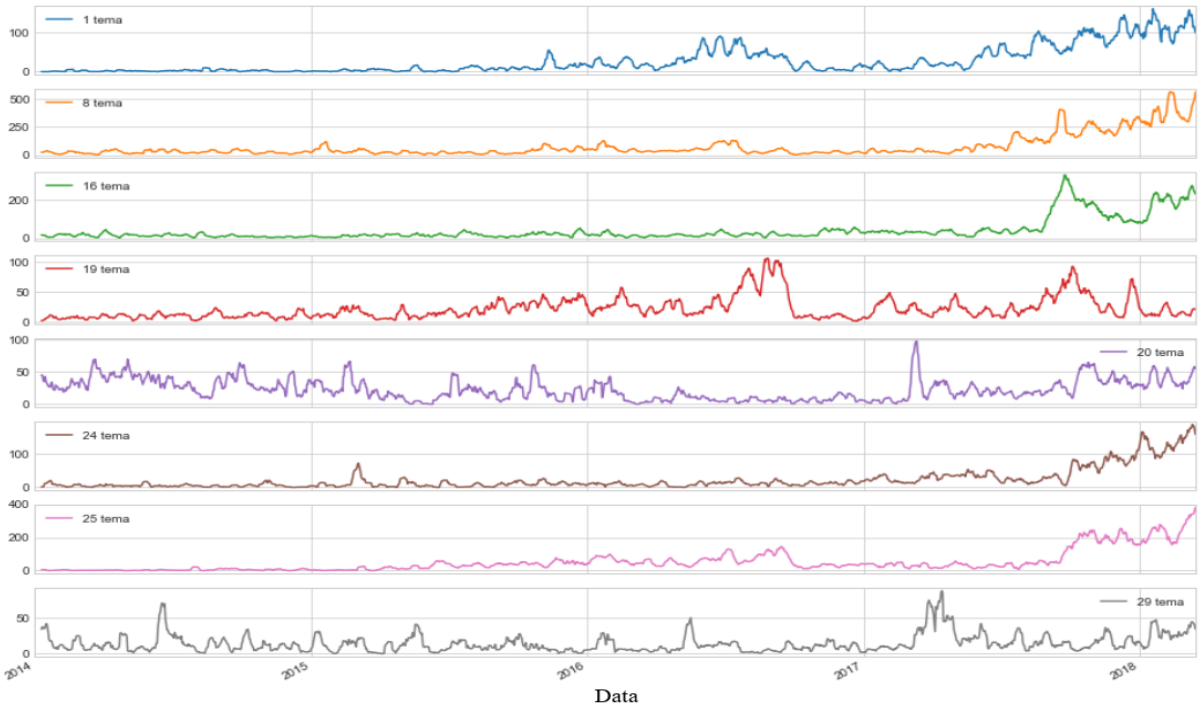
Tema	**Temos raktiniai žodžiai**
1 tema	*bitcoins, future, save, hold, spend, sell, saving, price, investment*
8 tema	*price, btc, buy, sell, market, increase, halving, go, exchange, value*
16 tema	*china, government, country, banned, regulation, control, ban, usa, Russia*
19 tema	*buy, accept, spend, bitcoins, online, store, spend, using, purchase, shop*
20 tema	*gold, value, better, supply, market, price, demand, asset, dollar, buuble*
24 tema	*altcoin, coin, cryptocurrency, etherum, ripple, better, monero, another,Litecoin*
25 tema	*investment, risk, gambling, trading, profit, invest, buy, scam, late, lose*
29 tema	*mining, miner, pool, electricity, hardware, profitable, block, network, cost, power*

Pritaikius DTM modelį kiekvienam grupuotam komentarui priskiriama viena tema. Tuomet pagal formulę

$$d_{i,t} = \sum_{j}^{n_{i,t}} s_{i,t_j} \tag{15}$$

apskaičiuojamas kiekvienos *t* dienos ir *i* temos indikatorius $d_{i.t}$, čia $s_{i.t}$ nurodo iš kiek komentarų buvo apjungtas *i* temos sugrupuotas komentaras, $n_{i.t}$ – skaičius nurodantis kiek *i* temos grupuotų komentarų buvo *t* dieną. Temų indikatoriai toliau buvo suglodintas slenkančio vidurkio metodu (pagal (16) formulę) pasirinkus 10 dienų laikotarpį (žr. 13 pav).

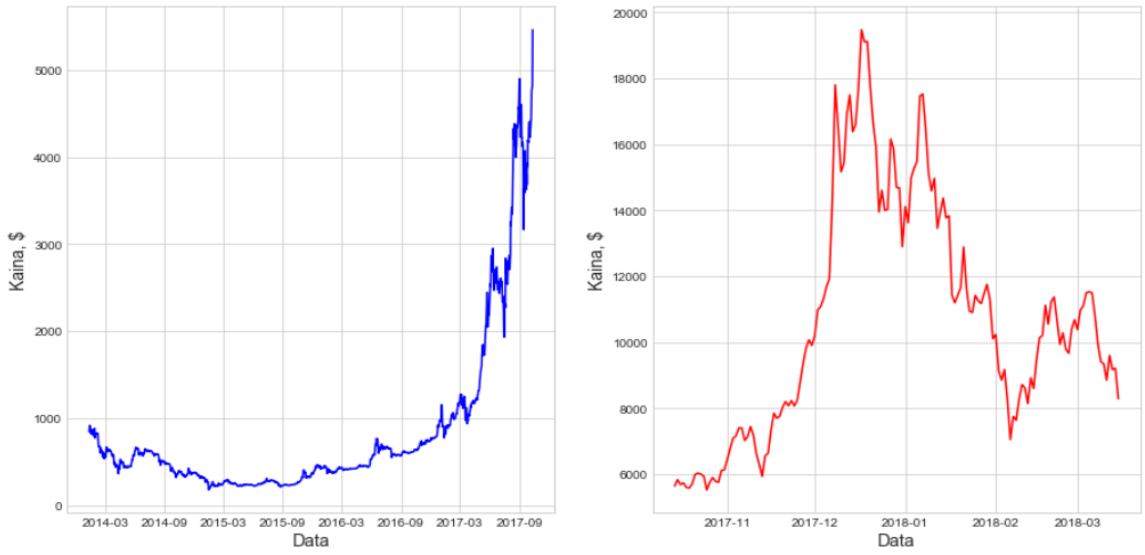
$$\hat{d}_{i,t} = \frac{\sum_{j=0}^{9} d_{i,t-j}}{10} \tag{16}$$



13 pav. DTM analizuojamų temų indikatoriai

3.3. Neuroninių tinklų modeliavimas

Prognozuosime bitkoino „*Open*“ kainą LSTM neuroniniais tinklais. Prognozavimui parinkta vienuolika nepriklausomų kintamųjų: bitkoino kainos „*Open“*, „*Close“*, analizuoti DTM temų indikatoriai ir „Google“ tendencijų indikatorius. Taip pat 90 % pirmųjų laikos eilutės duomenys bus naudojami modelio apmokymui, likusių 10 % -testavimui. 14 paveikslėlyje pateikiame priklausomo kintamojo apmokymo (grafiko kairėje) ir testavimo (grafiko dešnėje) laiko eilutes.



14 pav. Apmokymo ir testavimo laiko eilutės

LSTM neuroninių tinklų modeliavimui duomenis X (žr. (17) formulę) reikia „supjaustyti“ į *p* ilgio

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_n \end{bmatrix} \tag{17}$$

langus (angl. *window)* W, taip kad

$$W_i = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \dots \\ w_p \end{bmatrix} = \begin{bmatrix} x_i \\ x_{i+1} \\ x_{i+2} \\ \dots \\ x_{i+p-1} \end{bmatrix}, \tag{18}$$

čia x_i nepriklausomųjų kintamųjų vektorius i laikotarpyje. Tokiu būdu kiekvieno lango W_i prognozuojamas priklausomas kintamasis bus y_{i+p} .

Kad neuroniniai tinklai greičiau optimizuotų ir konverguotų į minimumą, kiekvieno lango W_i vektorius w_j normalizuosime pagal formulę

$$n_j = \frac{w_j}{w_1} - 1, \tag{19}$$

čia n_j normalizuotas w_j vektorius. Tokiu būdu duomenų languose gausime nepriklausomų kintamųjų procentinius pokyčius lyginant su pirmosiomis langų reikšmėmis.

3.3.1. Eksperimentai

Darysime kelis eksperimentus, kad rastume tiksliausiai bitkoino kainą prognozuojantį LSTM neuroninių tinklų modelį. Eksperimentus kartosime penkis kartus naudojant skirtingus modelio parametrus. Juose naudosime dviejų sluoksnių LSTM modelius, abiejuose sluoksniuose naudojama neuronų „išmetimo“ (angl. *dropout*) reguliarizacijos technika su 50 % tikimybė ignoruoti neuroną. Apmokydami modelį mokymosi imtį skaidysime kas 32 įrašus (angl. *batch size*), o optimizuoti klaidos funkciją naudosime stochastinio gradiento modifikacijos algoritmą *Adam.*

Eksperimentų rezultatus tikrinsime testuojamoje imtyje apskaičiuodami vidutines kvadratines paklaidas (RMSE) pagal formulę

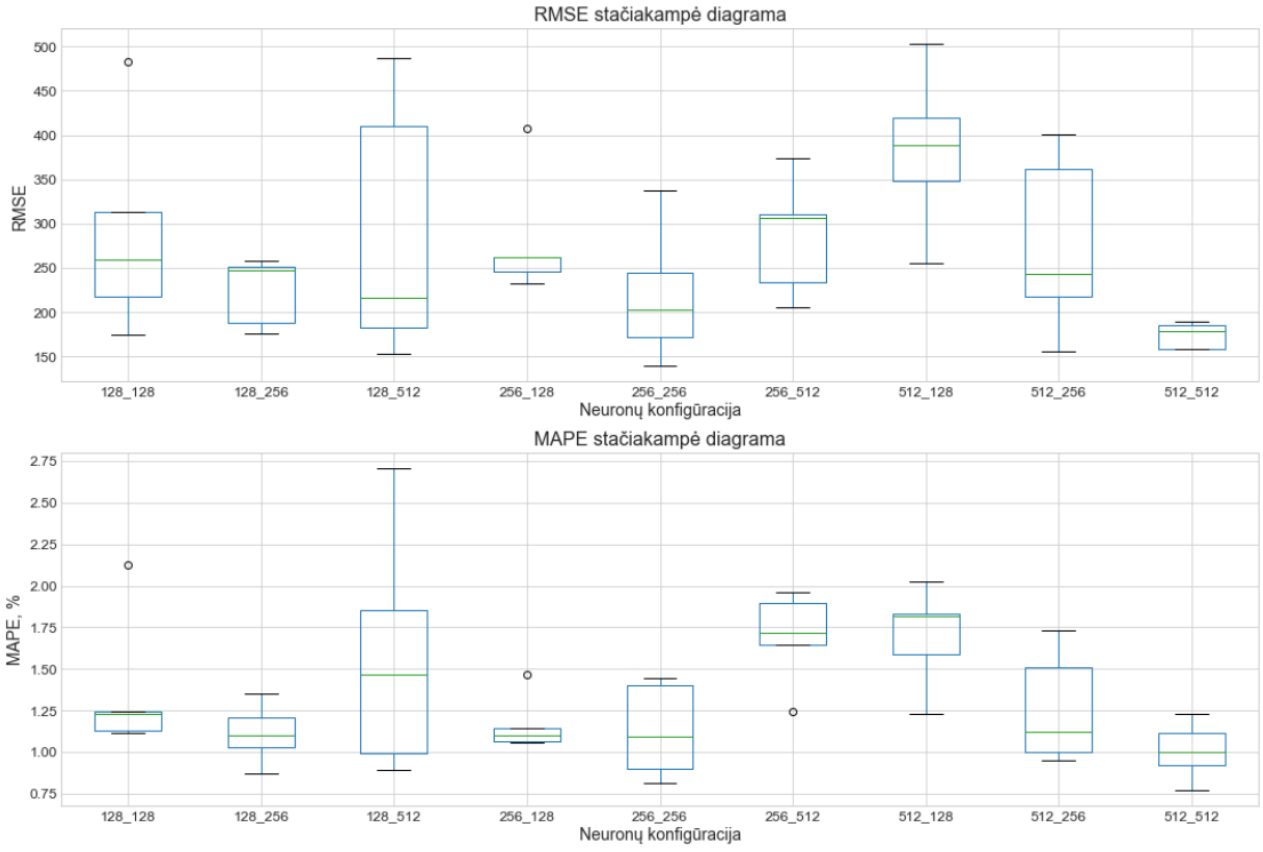
$$RMSE = \sqrt{\frac{\sum_1^n (\hat{y}_i - y_i)^2}{n}} \tag{20}$$

Ir vidutines procentines absoliučias paklaidas (MAPE), kurios apskaičiuojamos pagal formulę

$$MAPE = \frac{1}{n}\sum_{1}^{n}\frac{|\hat{y}_i - y_i|}{y_i}, \quad (21)$$

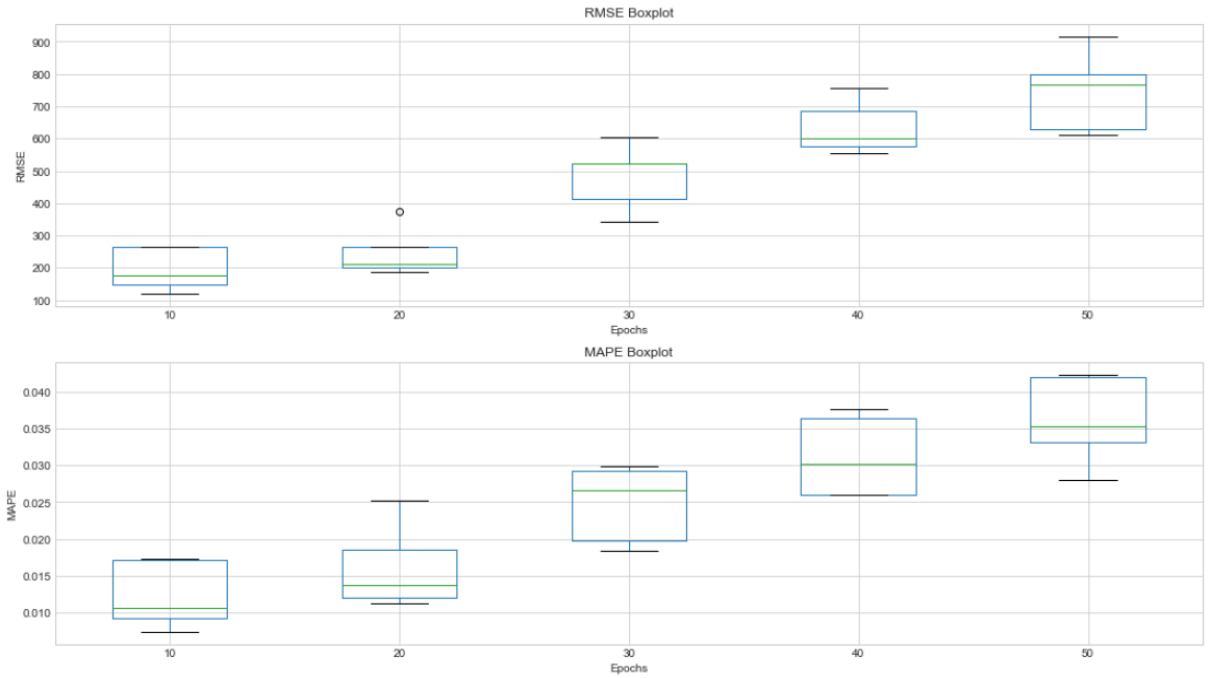
čia $\hat{y}$ prognozuojama bitkoino kaina, y – tikroji bitkoino kaina, n – testuojamos imties dydis.

Pirmuoju eksperimentu bandysime rasti efektyviausia neuronų kompoziciją modelyje, kiekviename sluoksnyje išbandysime 128, 256 ir 512 neuronus t.y. iš viso 9 poras. Kiekvienas neuronų pora buvo apmokoma naudojant 200 epochų, o lango ilgis - 15. Eksperimentų rezultatai pateikti 15 pav., grafike aiškiai matyti, kad geriausią neuronų kombinacija yra 512 ir 512 neuronų, šios konfigūracijos RMSE vidurkis 174,16, o MAPE – 1,01 %.



15 pav. Neuronų konfigūracijos tyrimo RMSE ir MAPE stačiakampė diagramos

Kitame eksperimente naudodami tuos pačius parametrus su (512,512) neuronų konfigūracija tyrinėsime skirtingus langų ilgius. Eksperimentų rezultatai pateikti 16 pav., geriausias rezultatas pasiektas su 5 ilgio langu, kurio eksperimentų RMSE vidurkis 112,65, o MAPE – 0,65 %.



16 pav. Langų ilgio tyrimo RMSE ir MAPE stačiakampės diagramos

3.3.2. Investavimo strategija

Toliau apmokydami LSTM modelį pagal rastus parametrus (neuronų konfigūracija – (512,512), lango ilgis - 5) bandysime prekiauti bitkoinu testuojamuoju laikotarpiu. Sprendimas pirkti bitkoiną ar toliau jį laikyti, jei jis jau yra nupirktas, atliekamas pasinaudojus taisykle

$$\widehat{y_{t+1}} > y_t \tag{22}$$

t.y. jei prognozuojama sekančios dienos bitkoino kaina $\widehat{y_{t+1}}$ yra didesnė už tikrą šiandieninę y_t – bitkoinas perkamas, analogiška taisyklė (23) naudojama pardavimo atveju.

$$\widehat{y_{t+1}} < y_t \tag{23}$$

Eksperimento metu bitkoinas yra nuperkamas pirmąją dieną, taip pat parduodamas paskutinę, jeigu jis yra nupirktas. Atliktų sprendimų tikslumas eksperimente yra net 94 %, pirkimo/pardavimo sprendimų sumaišymo matrica (angl. *confusion matrix*) pateikta 6 lentelėje.

6 lentelė. LSTM modelio pirkimo/pardavimo sprendimų matrica

	Pirkti (Prognozė)	Parduoti (Prognozė)
Pirkti	80	6
Parduoti	3	62

Pasitelkta strategijos atnešė 24 003,58 JAV dolerio pelno, o pirkimo ir laikymo strategija tuo pačiu laikotarpiu būtų atnešusi 2 674,23 JAV dolerio pelną, sprendimai pateikti 17 grafike. Reikia atkreipti dėmesį, kad eksperimente nebuvo pritaikyti realybėje egzistuojantys pirkimo mokesčiai.

17 pav. LSTM modelio bitkoino prekiavimas

Išvados

Iškastiems internetinio forumo https://bitcointalk.org komentarų failams buvo pritaikytas reliacinis duomenų modelis įrašų dublikatams pašalinti ir patogesniam darbui su duomenimis. Išviso iškasta 1,43 milijonas unikalių komentarų 2009 m. lapkričio 22d. - 2018m. kovo 15d. intervale. Peržiūrėjus jų pasiskirstymą laike pastebėta, kad diskusijos internete suaktyvėjo nuo 2013m pabaigos, todėl tyrimo laikotarpis pasirinktas nuo 2014m sausio 1d. iki 2018m. kovo 15d. Didžiausias komentarų aktyvumas pastebėtas nuo 2017m. pabaigos, kai bitkoino kainą pradėjo smarkiai smukti (nuo 19 000 iki 6 300 JAV dolerių) dieninis komentarų kiekis padidėjo 2 kartus per trijų mėnesių laikotarpį, kas indikuoja bitkoino bendruomenės norą įvertinti naująjį bitkoino rinkos ciklą ir jos investicines galimybes.

Programiškai apdorotiems komentarams buvo pritaikytas dinaminis temų modeliavimas. Modeliu buvo sudarytos trisdešimt komentaruose vyraujančių temų(klasterių) evoliucijų. Peržvelgus kelių rastų temų evoliucijas buvo įsitikinta, kad modelio temos yra lengvai suprantamos žmogui ir klasterio raktiniai žodžiai teisingai atspindi temos kontekstą kiekvienu laikotarpiu. Todėl šiuo įrankiu galima atrasti įdomias įžvalgas ir nustatyti potencialias rizikas, mūsų analizuotu atveju buvo rasta elektros sąnaudų rizika.

Remiantis ištraukta internetine informacija ir sudarytais dinaminio temų modeliavimo klasterių dieniniais indeksais buvo konstruojamas trumpos-ilgos atminties rekurentinis (LSTM) neuroninis tinklas skirtas bitkoino kainą prognozuoti. Keli eksperimentai buvo atlikti tiksliausiai prognozuojančiam modeliui surasti. Tiksliausias gautas modelis yra dviejų paslėptųjų sluoksnių neuroninis tinklas su (512,512) neuronų konfigūracija ir 5 lango ilgiu apmokytais duomenimis. Modelio vidutinė kvadratinė paklaida (RMSE) 126,5, o procentinė absoliuti paklaida (MAPE) – 0,65 %.

Pasinaudojus sukonstruotu neuroniniu modeliu sukurta automatizuota prekiavimo strategija, kurioje bitkoinas perkamas, jei sekančios dienos prognozė didesnė už dabartinę kainą, o parduodama jei prognozuojama kainą mažesnė už dabartinė. Strategijos atliktų sprendimų tikslumas testuotame laikotarpyje yra lygus 95 %. Automatizuoti prekybos sprendimai atnešė 24 003,58 JAV dolerių pelną, palyginimui, prikimo ir laikymo strategijos pelnas tuo pačiu laikotarpiu - 2 674,23 JAV dolerių.

Literatūros sąrašas

1. Aristotle's properties of a functional currency [interaktyvus]. 2013. [žiūrėta 2018-03-01]. Prieiga per: http://awildduck.com/?page_id=2941.
2. SATOSHI NAKAMOTO Bitcoin: A Peer-to-Peer Electronic Cash System [interaktyvus]. 2008. Prieiga per : https://bitcoin.org/bitcoin.pdf.
3. What is Cryptocurrency: Everything You Need To Know [Ultimate Guide]. *Blockgeeks* [interaktyvus]. [žiūrėta 2018-04-10]. Prieiga per: https://blockgeeks.com/guides/what-is-cryptocurrency.
4. DOVYDAS Kas yra blockchain technologija. [interaktyvus]. [žiūrėta 2018-04-14]. Prieiga per internetą: http://btc.lt/gidai/kas-yra-blockchain-technologija
5. Cryptocurrency Market Capitalizations | CoinMarketCap [interaktyvus]. [žiūrėta 2018-04-19]. Prieiga per internetą: https://coinmarketcap.com.
6. REIFF, N. G-20 Classifies Bitcoin as an Asset. *Investopedia* [interaktyvus]. 2018. [žiūrėta 2018-04-14]. Prieiga per : https://www.investopedia.com/news/g20-classifies-bitcoin-asset/.
7. BOURI, E. ir kt. Bitcoin for energy commodities before and after the December 2013 crash: diversifier, hedge or safe haven? *Applied Economics*. 2017. p. 1–11.
8. SUKAMULJA, S. - SIKORA, C.O. THE NEW ERA OF FINANCIAL INNOVATION: THE DETERMINANTS OF BITCOIN'S PRICE. In *Journal of Indonesian Economy and Business* . 2018. Vol. 33, no. 1, p. 46.
9. Bitcoin scalability problem. *Wikipedia* [Interaktyvus]. 2018. Prieiga per: https://en.wikipedia.org/w/index.php?title=Bitcoin_scalability_problem&oldid=838608793
10. FAMA, E.F. Efficient Capital Markets: A Review of Theory and Empirical Work. In *The Journal of Finance* . 1970. Vol. 25, no. 2, p. 383.
11. D. KLIMAŠAUSKIENĖ, V. MOŠČINSKIENĖ Lietuvos kapitalo rinkos efektyvumo problema [interaktyvus]. Prieiga per: http://elibrary.lt/resursai/DB/LB/LB_pinigu_studijos/Pinigu_studijos_1998_02_03.pdf.
12. FOX, J. *The myth of the rational market: a history of risk, reward, and delusion on Wall Street*. . 1st ed. Ed. New York: Harper Business, 2009. 382 p. ISBN 978-0-06-059899-0.
13. QUIGGIN, J. The Bitcoin Bubble and a Bad Hypothesis. *The National Interest* [interaktyvus]. [žiūrėta 2018-05-15]. Prieiga per: http://nationalinterest.org/commentary/the-bitcoin-bubble-bad-hypothesis-8353.
14. SHILLER, R. [interaktyvus]. .Cambridge, MA: National Bureau of Economic Research, 2017. [žiūrėta 2018-04-10]. Prieiga per internetą: http://www.nber.org/papers/w23075.pdf.
15. KRISTOUFEK, L. BitCoin meets Google Trends and Wikipedia: Quantifying the relationship between phenomena of the Internet era. *Scientific Reports* [interaktyvus]. 2013. Vol. 3, no. 1. [žiūrėta 2018-05-03]. . Prieiga per internetą: http://www.nature.com/articles/srep03415.
16. CIAIAN, P. ir kt. The economics of BitCoin price formation. *Applied Economics* . 2016. Vol. 48, no. 19, p. 1799–1815.
17. YERMACK, D. Is Bitcoin a Real Currency? An Economic Appraisal. *Handbook of Digital Currency* [interaktyvus]. [s.l.]: Elsevier, 2015. p. 31–43. [žiūrėta 2018-04-12]. ISBN 978-0-12-802117-0 Prieiga per: http://linkinghub.elsevier.com/retrieve/pii/B9780128021170000023.

18. D. VAN WIJK What can be expected from the Bitcoin? [interaktyvus]. 2013. Prieiga per: https://webcache.googleusercontent.com/search?q=cache:26oRzTYnyUsJ:https://thesis.eur.nl/pub/14100/Final-version-Thesis-Dennis-van-Wijk.pdf+&cd=1&hl=lt&ct=clnk&gl=lt.

19. ZHU, Y. ir kt. Analysis on the influence factors of Bitcoin's price based on VEC model. In *Financial Innovation* [interaktyvus]. 2017. Vol. 3, no. 1. [žiūrėta 2018-04-12]. . Prieiga per: http://jfin-swufe.springeropen.com/articles/10.1186/s40854-017-0054-0.

20. WANG, J. ir kt. An Analysis of Bitcoin Price Based on VEC Model [interaktyvus]. [s.l.]: Atlantis Press, 2016. [žiūrėta 2018-04-12]. Prieiga per: http://www.atlantis-press.com/php/paper-details.php?id=25859324.

21. BUCHHOLZ, M., J. DELANEY, J. WARREN, AND J. PARKER. Bits and Bets, Information, Price Volatility, and Demand for BitCoin, Economics 312 [interaktyvus]. 2012. Prieiga per: https://www.reed.edu/economics/parker/s12/312/finalproj/Bitcoin.pdf.

22. J. BOUOIYOUR, R. SELMI - ANNALS OF ECONOMICS & FINANCE What Does Bitcoin Look Like? In . 2015.

23. BOUOIYOUR, J., SELMI, R., TIWARI, A. K., & OLAYENI, O. R. What drives bitcoin price. Economics Bulletin. 2016.

24. POLASIK, M. ir kt. Price Fluctuations and the Use of Bitcoin: An Empirical Inquiry. *SSRN Electronic Journal* [interaktyvus]. 2014. [žiūrėta 2018-04-12]. . Prieiga per: http://www.ssrn.com/abstract=2516754.

25. VASSILIADIS, S., PAPADOPOULOS, P., RANGOUSSI, M., - KONIECZNY, T., & GRALEWSKI, J. Bitcoin value analysis based on crosscorrelations. In *Journal of Internet Banking and Commerce, 22(S7)* . 2017.

26. 10 Key Marketing Trends for 2017 and Ideas for Exceeding Customer Expectations [interaktyvus]. Prieiga per : https://public.dhe.ibm.com/common/ssi/ecm/wr/en/wrl12345usen/watson-customer-engagement-watson-marketing-wr-other-papers-and-reports-wrl12345usen-20170719.pdf.

27. The biggest data challenges that you might not even know you have. *Watson* [interaktyvus]. 2016. [žiūrėta 2018-04-15]. Prieiga per: https://www.ibm.com/blogs/watson/2016/05/biggest-data-challenges-might-not-even-know/.

28. HAN, J. ir kt. *Data mining: concepts and techniques*. San Francisco, Calif: Morgan Kaufmann, 2011. ISBN 978-0-12-381479-1.

29. KAMINSKI, J. Nowcasting the Bitcoin Market with Twitter Signals. In *arXiv:1406.7577 [cs]* [interaktyvus]. 2014. Prieiga per: http://arxiv.org/abs/1406.7577.

30. Referenced in Algorithmic Trading of Cryptocurrency Based on Twitter Sentiment Analysis - Semantic Scholar [interaktyvus]. [žiūrėta 2018-04-18]. Prieiga per: http://cs229.stanford.edu/proj2015/029_report.pdf.

31. STENQVIST, E. - LÖNNÖ, J. *Predicting Bitcoin price fluctuation with Twitter sentiment analysis* 2017.

32. KIM, Y.B. ir kt. Virtual World Currency Value Fluctuation Prediction System Based on User Sentiment Analysis. In ZHOU, W.-X.Sud. *PLOS ONE* . 2015. Vol. 10, no. 8, p. e0132944.

33. KIM, Y.B. ir kt. Predicting Fluctuations in Cryptocurrency Transactions Based on User Comments and Replies. In *PLOS ONE* . 2016. Vol. 11, no. 8, p. e0161197.

34. KEANE, N. ir kt. Using Topic Modeling and Similarity Thresholds to Detect Events. *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation* [interaktyvus]. Denver, Colorado: Association for Computational Linguistics, 2015. p. 34–42. Prieiga per: http://www.aclweb.org/anthology/W15-0805.

35. BANKAI.LT Bitcoin ir kriptovaliutų keityklos. Kur nusipirkti ir parduoti? *Bankai.lt* [interaktyvus]. [žiūrėta 2018-04-16]. Prieiga per: http://www.bankai.lt/kriptovaliutos/informacija/bitcoin-ir-kriptovaliutu-keityklos-kur-nusipirkti-ir-1261.html.

36. MARCO LINTON, ERNIE GIN SWEE TEO, ELISABETH BOMMES, CATHY YI-HSUAN CHEN, WOLFGANG K. HÄRDLE Dynamic Topic Modelling for Cryptocurrency Community Forums [interaktyvus]. Prieiga per: http://sfb649.wiwi.hu-berlin.de/papers/pdf/SFB649DP2016-051.pdf.

37. List of Major Bitcoin Heists, Thefts, Hacks, Scams, and Losses [interaktyvus]. [žiūrėta 2018-0-16]. Prieiga per: https://bitcointalk.org/index.php?topic=576337.0

38. KIM, Y.B. ir kt. When Bitcoin encounters information in an online forum: Using text mining to analyse user opinions and predict value fluctuation. In CHOO, K.-K.R.Sud. *PLOS ONE* . 2017. Vol. 12, no. 5, p. e0177630.

39. HAGAN, M.T. ir kt. *Neural Network Design*. 2 edition. Ed. Wrocław: Martin Hagan, 2014. 800 p. ISBN 978-0-9717321-1-7.

40. HOCHREITER, S. - SCHMIDHUBER, J. Long Short-term Memory. In *Neural computation* . 1997. Vol. 9, p. 1735–80.

41. Understanding LSTM Networks -- colah's blog [interaktyvus]. [žiūrėta 2018-04-21]. Prieiga per: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

42. M. BLEI, D. - D. LAFFERTY, J. Dynamic Topic Models. In *ICML 2006 - Proceedings of the 23rd International Conference on Machine Learning* . 2006. p. 113–120.

43. The Incredible Growth of Python | Stack Overflow. *Stack Overflow Blog* [interaktyvus]. 2017. [žiūrėta 2018-04-19]. Prieiga per: https://stackoverflow.blog/2017/09/06/incredible-growth-python/.

44. The State of ML and Data Science 2017. *Kaggle* [interaktyvus]. [žiūrėta 2018-05-19]. Prieiga per: https://www.kaggle.com/surveys/2017.

45. Ranking Popular Deep Learning Libraries for Data Science [Interaktyvus]. 2017. [žiūrėta 2018-05-19]. Prieiga per: https://blog.thedataincubator.com/2017/10/ranking-popular-deep-learning-libraries-for-data-science/.

46. Bitcoin will use 0.5% of the world's electricity by end of 2018. *The Independent* [interaktyvus]. 2018. [žiūrėta 2018-05-18]. Prieiga per: http://www.independent.co.uk/life-style/gadgets-and-tech/news/bitcoin-mining-energy-use-electricity-cryptocurrency-a8353981.html.

47. Many Bitcoin Miners Are at Risk of Turning Unprofitable. In *Bloomberg.com* [interaktyvus]. 2018. [žiūrėta 2018-05-18]. Prieiga per: https://www.bloomberg.com/news/articles/2018-04-18/bitcoin-miners-facing-a-shakeout-as-profitability-becomes-harder.

Priedai

1 priedas

Pagalbinių funkcijų programos kodas

```
import  time, os, warnings
os.environ["CUDA_DEVICE_ORDER"] = "PCI_BUS_ID"   # see issue #152
os.environ["CUDA_VISIBLE_DEVICES"] = ""
import numpy as np
import pandas as pd
from numpy import newaxis
import tensorflow as tf
from keras.layers.core import Dense, Activation, Dropout
from keras.models import Sequential
from keras.layers.recurrent import LSTM
from keras.models import load_model
import matplotlib.pyplot as plt

os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3' #Hide messy TensorFlow warnings
warnings.filterwarnings("ignore") #Hide messy Numpy warnings

def build_model(layers, dropout = [0.5, 0.5], loss_function = "mse", optimizer =
"adam", activation = "linear"):
    model = Sequential()

    model.add(LSTM(
        input_dim=layers[0],
        output_dim=layers[1],
        return_sequences=True))
    model.add(Dropout(dropout[0]))

    model.add(LSTM(
    layers[2],
        return_sequences=False))
    model.add(Dropout(dropout[0]))

    model.add(Dense(
        output_dim=layers[3]))
    model.add(Activation(activation))

    start = time.time()
```

```
    model.compile(loss=       loss_function,       optimizer=       optimizer,
metrics=['accuracy'])

    print("> Compilation Time : ", time.time() - start)
    return model

def load_network(filename):
    if(os.path.isfile(filename)):
        return load_model(filename)
    else:
        print('ERROR: "' + filename + '" file does not exist ')
        return None


#Help functions
def normalise_windows(window_data):
    normalised_data = []
    for window in window_data:
        #normalised_window = [ np.tanh(((float(p) / float(window[0])) - 1)) for p
in window]
        normalised_window = [ ((float(p) / float(window[0])) - 1) for p in window]
        normalised_data.append(normalised_window)
    return normalised_data

def get_real_value(y_data, raw_data, rows):
    #de-normalise predictions to bitcion prices
    real_values = []
    i = 0
    for value in y_data:
        real_values.append(round((value+1) * raw_data[rows + i],2) )
        #real_values.append(round((np.arctanh(value)+1) * raw_data[rows + i],2) )
        i += 1
    return real_values


def   transform_multivariate_data(data,   y_cols,   window_size,   y_window_size,
normalise_window, col_threshold):
    """
    data - primary data which will be transformed into test and train X,Y datasets
    y_cols - list of y columns name
    normalise_window - boolean variable that indicates if variables have to be
normalisated
```

```
    col_threshold - show how many columns (first in dataframe) has to be
normalisated (if normalise_window TRUE)
    """

    #window_size = window_size + 1
    result = []

    #Convert y-predict columns names to numerical indexes
    y_cols = [list(data).index(col) for col in y_cols]
    data = np.array(data)
    #np.random.shuffle(data)

    for index in range(len(data) - window_size - y_window_size +1) :
        result.append(data[index: index + window_size + y_window_size])

    if normalise_window:
        result = normalise_windows_multiple(result, col_threshold)

    result = np.array(result)
    row = round(0.9 * result.shape[0])

#x_train = result[:int(row), :-y_window_size, :] autoregression case
    x_train = result[:int(row), :-y_window_size, :]
    y_train = result[:int(row), -y_window_size:, y_cols]
    x_test = result[int(row):, :-y_window_size, :]
    y_test = result[int(row):, -y_window_size:, y_cols]

    #y_train = y_train.reshape((y_train.shape[0], 2))
    #y_test = y_test.reshape((y_test.shape[0], 2))
    y_train  =  y_train.reshape((y_train.shape[0],  len(y_cols)*y_window_size),
order = "F")
    y_test = y_test.reshape((y_test.shape[0], len(y_cols)*y_window_size), order =
"F")

    return [x_train, y_train, x_test, y_test, row]

def normalise_windows_multiple(window_data, threshold):
    "threshold - show until which column data should be normalised"
    normalised_data = []
    for window in window_data:
        normalised_window = [ np.append((w[:threshold] / window[0][:threshold] -
1), w[threshold:]) for w in window]
        normalised_data.append(normalised_window)
```

```
    return normalised_data

def calculate_error_multiple(predictions, y):

    error = predictions - y
    rmse = np.power(np.mean(np.power(error,2), axis = 0),0.5)
    mape = np.mean(np.abs(error/y), axis = 0)
    #2018-03-29 correct output
    #return  rmse[0], mape[0]
    return  rmse, mape

def get_real_values_multiple(predictions, window_size, y_window_size, raw_data,
pred_col_num, rows):
    #rawdata = df[['Open','Close']]

    pred_usd = []
    for i in range(len(predictions)):
        pred_usd.append((np.reshape(predictions[i],(y_window_size,
pred_col_num), order = 'F') + 1) * raw_data.values[rows + i])
    #2018-03-29 correct output np.array shape
    #return np.array(pred_usd)
    return np.array(pred_usd).reshape((len(pred_usd),pred_col_num))
```

Iškastų bitcointalk.org komentarų failų transformavimas į reliacinį modelį

```
import json, os, datetime
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.dates as mdates

%matplotlib inline
plt.style.use('seaborn-whitegrid')

#needed folders paths
raw_data_path = os.path.join('..', '..', 'Forum', 'data')
raw_data_path2 = os.path.join('..', '..', 'Forum', 'data2')
raw_data_path3 = os.path.join('..', '..', 'Forum', 'data3')
raw_data_path4 = os.path.join('..', '..', 'Forum', 'data4')
data_path = os.path.join('..', 'data')
topics_path = os.path.join('..', 'topics', 'topic.pickle')

file_names = [os.path.join(raw_data_path, pos_json) for pos_json in
os.listdir(raw_data_path) if pos_json.endswith('.json')]

forum = list()
for i in range(len(file_names)):
    with open(file_names[i] ,encoding="utf-8") as json_data:
        forum += json.load(json_data)['posts']

comments = [(datetime.datetime.strptime(rep["date"][:10], '%Y-%m-%d').date(),
             clean_string(rep["content"]),
             clean_string(dic["topic"]),
             clean_string(dic["content"]),
            datetime.datetime.strptime(dic["date"][:10], '%Y-%m-%d').date(),
            dic["views"])
               for dic in forum  if 'replies' in dic
               for rep in dic['replies'] if rep["content"] != []]

df = pd.DataFrame(comments, columns = ["Date", "Comment", "Topic", "Content",
"Topic date", "Views"])
df = df.sort_values(["Date", "Topic"])
```

```
if os.path.isfile(topics_path):
    topics = pd.read_pickle(topics_path)
else:
    topics = pd.DataFrame(columns= ["Topic", "Content", "Topic date", "Views"])

#insert new topics
indexes = []
i = 0
for topic in df["Topic"]:
    index = topics[topics["Topic"] == topic].index
    if len(index) == 0:
        topics = topics.append(df.iloc[[i]][["Topic", "Content", "Topic date",
"Views"]], ignore_index= True)
        index = len(topics) - 1
    else:
        index = index[0]
    i += 1
    indexes.append(index)
df["Topic No."] = indexes
topics.to_pickle(topics_path)

#write processed data to pickle
for date in df["Date"].unique():
    df2 = df[df["Date"] == date][["Topic No.", "Comment", "Date"]]
    if os.path.isfile(os.path.join(data_path, str(date))):
          df3 = pd.read_pickle(os.path.join(data_path, str(date)))
          df2 = pd.concat([df2, df3], ignore_index=True).drop_duplicates()
    df2.to_pickle(os.path.join(data_path, str(date)))

#read pickle files
df = pd.DataFrame()
for file in os.listdir(data_path):
    help_df = pd.read_pickle(os.path.join(data_path, str(file)))
    help_df["Date"] = datetime.datetime.strptime(file, '%Y-%m-%d').date()
    df = pd.concat([df, help_df], ignore_index=True)
```

Komentarų apdorojumas ir DTM paruošimas

```
import json, os, re, csv, datetime, gzip
import pandas as pd
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import CountVectorizer
from nltk.probability import FreqDist
import pickle
import gensim
import pyLDAvis
import pyLDAvis.gensim
import numpy as np
from gensim import corpora, utils
from gensim.models.wrappers.dtmmodel import DtmModel
from gensim.models import ldaseqmodel

%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('seaborn-whitegrid')

import warnings
warnings.filterwarnings('ignore')

topics_path = os.path.join('..', 'topics', 'topic.pickle')
data_path = os.path.join('..', 'data')
models_path = os.path.join('..', 'models')
project_data_path = os.path.join('..', 'project_data')

#read pickle files
df = pd.DataFrame()
for file in os.listdir(data_path):
    help_df = pd.read_pickle(os.path.join(data_path, str(file)))
    help_df["Date"] = datetime.datetime.strptime(file, '%Y-%m-%d').date()
    df = pd.concat([df, help_df], ignore_index=True)

topics = pd.read_pickle(topics_path)

for i in range(len(topics)):
    index = df[df["Topic No."] == i].index[0]
```

```
    df.at[index, "Comment"] = topics.at[i,"Topic"] + topics.at[i,"Content"] +
df.at[index, "Comment"]
comments = df.groupby(["Date", "Topic No."])["Comment"].apply(lambda x:
''.join(x))

stop_words = stopwords.words("english")
lemmatizer = WordNetLemmatizer()
def preprocessText(text):
    text = text.lower()
    text = re.sub('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\(\),]|(?:%[0-9a-
fA-F][0-9a-fA-F]))+', 'web_url', text)
    text = re.sub('www.(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\(\),]|(?:%[0-9a-fA-F][0-
9a-fA-F]))+', 'web_url', text)
    text = re.sub(r"what's", "what is ", text)
    text = re.sub(r"\'s", " ", text)
    text = re.sub(r"\'ve", " have ", text)
    text = re.sub(r"can't", "cannot ", text)
    text = re.sub(r"n't", " not ", text)
    text = re.sub(r"i'm", "i am ", text)
    text = re.sub(r"\'re", " are ", text)
    text = re.sub(r"\'d", " would ", text)
    text = re.sub(r"\'ll", " will ", text)
    text = re.sub("[^a-zA-Z]", " ", text)
    text = re.sub(r" e g ", " eg ", text)
    text = re.sub(r" b g ", " bg ", text)
    text = re.sub(r"e mail", "email", text)
    words = text.split()
    meaningful_words = [lemmatizer.lemmatize(w) for w in words if not w in
stop_words and len(w) > 1]
    return meaningful_words

tokens = []
for i in range(0,len(comments)):
    if i % 50000 == 0:
        print(datetime.datetime.now().time(),i)
    tokens.append(preprocessText(comments[i]))

bigram_model = Phrases(tokens)
bigram_tokens = []
for token in tokens:
    bigram_tokens.append(bigram_model[token])

dates = comments.index.get_level_values(0)
```

```
freq = dates.value_counts()
freq.index = freq.index.to_period("Q")
time_seq =list(freq.groupby(freq.index).sum())

class  DTMcorpus(corpora.textcorpus.TextCorpus):

    def get_texts(self):
        return self.input

    def __len__(self):
        return len(self.input)

corpus = DTMcorpus(bigram_tokens)
corpus.dictionary.filter_extremes(no_below=15, no_above=0.5)

model = DtmModel("dtm-win64.exe", corpus, time_seq, num_topics=30,
                 id2word=corpus.dictionary, initialize_lda=True)
```

„Google“ tendencijų laiko eilutės gavimas

```
import requests, datetime
import pandas as pd
from pytrends.request import TrendReq

def get_raw_daily_google_trend(topic, from_date, end_date ):

    pytrends = TrendReq(acc, pass, hl='en-US', tz=360, custom_useragent=None)
    df = pd.DataFrame()
    from_date = next_weekday(from_date, 6)
    loop_end_date = from_date

    while loop_end_date != end_date:
        if from_date + datetime.timedelta(181) < end_date:
            loop_end_date = from_date + datetime.timedelta(181)
        else:
            loop_end_date = end_date

        pytrends.build_payload(kw_list=[topic],           timeframe           =
from_date.strftime("%Y-%m-%d ") + loop_end_date.strftime("%Y-%m-%d"))
        df =  df.append(pytrends.interest_over_time())
        from_date = loop_end_date + datetime.timedelta(1)

    return df

def get_raw_google_trend(topic, from_date, end_date):
    pytrends   =   TrendReq('darbasbitcoin',   'bitcoin9',   hl='en-US',   tz=360,
custom_useragent=None)
    pytrends.build_payload(kw_list=[topic], timeframe = from_date.strftime("%Y-
%m-%d ") + end_date.strftime("%Y-%m-%d"))
    df =  pytrends.interest_over_time()
    return df

def get_google_trend(topic, from_date, end_date = datetime.datetime.today()):
  #Function returns daily google trends in given period

    df1 = get_raw_daily_google_trend(topic, from_date, end_date)
    df2 = get_raw_google_trend(topic, from_date, end_date)
    df3 = pd.DataFrame()
    length = len(df1)
```

```
    index = -7
    for value in df2[topic]:
        index += 7
        if index +7 < length:
            index2 = index + 7
        else:
            index2 = length
        sum = df1[topic][index:index2].sum()
        df3 = df3.append(pd.DataFrame(df1[topic][index:index2] * value /sum))
    return df3

def next_weekday(d, weekday):
    # 0 = Monday, 1=Tuesday, 2=Wednesday...
    days_ahead = weekday - d.weekday()
    if days_ahead <= 0: # Target day already happened this week
        days_ahead += 7
    return d + datetime.timedelta(days_ahead)
```

LSTM neuroninių tinklų eksperimentai

```
import os, six, datetime, pickle, gzip
import pandas as pd
import numpy as np
from gensim.models.wrappers.dtmmodel import DtmModel
from Scripts.LSTM import *
from Scripts.GoogleTrends import get_google_trend
from sklearn.metrics import confusion_matrix
from keras.utils.np_utils import to_categorical

%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('seaborn-whitegrid')

project_data_path = os.path.join('..', 'project_data')

df2 = pd.read_pickle(os.path.join(project_data_path, "freq_df"))
count = df2.groupby(("Topic","Date"))["freq"].sum()

urlMarket       =       'http://coinmarketcap.com/currencies/bitcoin/historical-
data/?start=20131213&end=20180316'
df = pd.read_html(urlMarket)[0]
df.index  = [datetime.datetime.strptime(date, '%b %d, %Y') for date in df["Date"]]
df.index.name = 'Date'
df = df.sort_index()

del df["Date"], df["Market Cap"], df["High"], df["Low"]

for topic in [1,8,16,19,20,24,25,29]:
    df[str(topic)] = count.ix[topic]
df = df.fillna(0)
df[["1","8","16",'19',"20",'24','25','29']]                                    =
df[["1","8","16",'19',"20",'24','25','29']].rolling(10).mean() + 1
df2          =     get_google_trend('bitcoin',     datetime.datetime(2013,12,27),
datetime.datetime(2018,3,19))
df2 = df2.drop(df2.index[[0,1,2,len(df2)-4,len(df2)-3,len(df2)-2,len(df2)-1]])
df = pd.merge(df, df2, left_index=True, right_index=True)
df["y"] = buy
df["y1"] = categorical_labels[:,0]
```

```
df["y2"] = categorical_labels[:,0]

#data = df[["Open","Close","1",'8',"16","19","20","24","25","29","bitcoin"]]
data = df[["1",'8',"16","19","20","24","25","29","bitcoin",'y']]
y_window_size = 1
window_size =5


repeats = 5
i = 0
neurons = [128,256,512]
result = []

for n1 in neurons:
    for n2 in neurons:

        rmse_scores, mape_scores = list(), list()
        loss_train, loss_validate = list(), list()

        for r in range(repeats):

            # fit the model
            model = build_model([len(list(data)), n1, n2, 1]) #LSTM
            loss = model.fit(
                x_train,
                y_train,
                batch_size=32,
                nb_epoch=200,
                validation_split=0.2)

             # forecast test dataset

            predictions = model.predict(x_test)
            pred_usd    =    get_real_values_multiple(predictions,    window_size,
y_window_size, data["Open"], 1, rows)
            rmse, mape = calculate_error_multiple(pred_usd, y_test_usd)
            rmse_scores.append(rmse)
            mape_scores.append(mape)
            loss_train.extend(loss.history["loss"])
            loss_validate.extend(loss.history["val_loss"])

            print('%d %d-%d) RMSE: %.2f  MAPE: %.3f' % (r+1, n1, n2, rmse, mape))
```

```
        result.append((rmse_scores, mape_scores, loss_train, loss_validate))

with open(os.path.join(project_data_path, 'neurons_'), 'wb') as f:
    pickle.dump(result, f, pickle.HIGHEST_PROTOCOL)



repeats = 5
pred_len = 1
y_window_size, y_col_num = 1, 1
windows_size = [5,10,15,20,25]
#windows_size = [5,10]
result = []

for w in windows_size:

    x_train, y_train, x_test, y_test, rows =  transform_multivariate_data(data,
["Open"], w, y_window_size, True, 11)
    _, __, ___, y_test_usd, _____ =  transform_multivariate_data(data, ["Open"],
w, y_window_size, False, 4)

    #initialize variables
    rmse_scores, mape_scores = list(), list()
    loss_train, loss_validate = list(), list()

    for r in range(repeats):

        # fit the model
        model = build_model([len(list(data)), 512, 512, 1]) #LSTM
        loss = model.fit(
            x_train,
            y_train,
            batch_size=32,
            nb_epoch=200,
            validation_split=0.2)

        # forecast test dataset

        predictions = model.predict(x_test)
        pred_usd  =  get_real_values_multiple(predictions,  w,  y_window_size,
data[["Open"]], 1, rows)
        rmse, mape = calculate_error_multiple(pred_usd, y_test_usd)
        rmse_scores.append(rmse)
```

```
            mape_scores.append(mape)
            loss_train.extend(loss.history["loss"])
            loss_validate.extend(loss.history["val_loss"])
            print('%d %d) RMSE: %.2f,  MAPE: %.4f' % (r+1, w, rmse, mape))

        result.append((rmse_scores, mape_scores, loss_train, loss_validate))

with open(os.path.join(project_data_path, 'windows_length'), 'wb') as f:
    pickle.dump(result, f, pickle.HIGHEST_PROTOCOL)
```