VILNIAUS UNIVERSITETAS

JELENA LIUTVINAVIČIENĖ

DIDELĖS APIMTIES DUOMENŲ VIZUALI ANALIZĖ

Daktaro disertacija,

Fiziniai mokslai, informatika (09 P)

Vilnius, 2019

Summary

This dissertation focuses on massive data visualization that is based on dimensionality reduction methods.

There is presented the comparative analysis of dimensionality reduction methods and existing visualization tools applying them. The speed and accuracy of various dimensionality reduction methods were investigated during this research. The insights how parallel computing methods can increase the performance of dimensionality reduction processes are presented as well.

There is proposed a new methodology, which improves visual data analysis process and brings new possibilities. It divides the whole data visualization process into separate interactive steps. In each step, some part of data can be selected for further analysis and visualization. The different dimensionality method can be chosen/changed in each step. The decision which methods to be chosen depends on desirable accuracy measures and visualization samples. In addition, there are provided statistical measures of the identified clusters. We have developed a special tool, which implements the proposed methodology. R language and Shiny package were used for developing the tool.

The main results of the dissertation were published in 7 research papers: 3 papers are published in periodicals, reviewed scientific journals (one of them is indexed in „Clarivate Analytics Web of Science“ and has impact factor) and 4 papers are published in conference proceedings. The main results have been presented and discussed at 7 national and international conferences.

The dissertation consists of 5 chapters and the list of references. The scope of the work is 98 pages including 74 figures and 3 tables. The list of references consists of 115 sources.

Santrauka

Šioje disertacijoje sprendžiami uždaviniai, susiję su didžiųjų duomenų analize ir vizualizavimu. Tyrimo objektas yra didelės apimties daugiamačiai duomenys ir dimensijų mažinimo metodai didelės apimties daugiamačiams duomenims vizualizuoti.

Tyrimo tikslas yra pasiūlyti integralią didelės apimties duomenų vizualios analizės metodologiją, apimančią skirtingų dimensijų mažinimo metodų taikymą, jų statistinių savybių panaudojimą metodų parinkimui, duomenų paruošimą analizei bei jų klasterizavimą. Pasiūlytą metodologiją realizuojantis įrankis turi veikti paskirstytų skaičiavimų sistemose / debesyje.

Darbe analitiškai apžvelgti ir palyginti didelės apimties duomenų vizualizavimo metodai, juos įgyvendinantys įrankiai bei technologijos, įgalinančios analizuoti didelės apimties duomenis. Ištirta, kaip lygiagrečiųjų skaičiavimų taikymas gali paspartinti duomenų dimensijų mažinimo bei vizualizavimo užduotis. Taip pat išnagrinėti vizualizavimo metodų, pagrįstų dimensijų mažinimu, tikslumo įvertinimo matai; atlikti metodų greičio ir tikslumo vertinimo tyrimai.

Darbe pristatoma duomenų vizualizavimo metodologija, paremta dimensijų mažinimo metodais. Pasiūlyta metodologija suteikia daugiau galimybių vizualizuojant didžiuosius duomenis. Ji leidžia duomenis analizuoti įvairiais pjūviais, duomenis analizuoti pažingsniui, kiekviename etape pasirenkant dominančią duomenų aibę ir jai pritaikant geriausiai tinkantį dimensijų mažinimo metodą, atsižvelgiant į dimensijų mažinimo greitį ir tikslumą konkrečiu atveju. Metodų pasirinkimą palengvina pateikiami grafiniai vizualizavimo pavyzdžiai bei greičio ir tikslumo rodikliai.

Pasiūlytą metodologiją realizuoja sukurtas įrankis, kurio architektūra įgalina duomenų apdorojimą atlikti panaudojant debesų kompiuterijos išteklius. Įrankio galimybės pristatomos aprašant realių duomenų vizualizavimo atvejus.

Tyrimų rezultatai publikuoti 7 moksliniuose leidiniuose: 3 periodiniuose recenzuojamuose mokslo žurnaluose, iš jų viename leidinyje, referuojamame „Clarivate Analytics Web of Science“ duomenų bazėje ir turinčiame citavimo indeksą, ir 4 pateikiami konferencijos pranešimų medžiagoje. Šie rezultatai pristatyti ir aptarti 7 nacionalinėse ir tarptautinėse mokslinėse konferencijose.

Disertaciją sudaro 5 skyriai. Visa disertacijos apimtis – 98 puslapiai. Joje pateikti 74 paveikslai ir 3 lentelės. Disertacijoje remtasi 115 literatūros šaltiniais.

Reikšminiai žodžiai: didieji duomenys, vizualizavimo metodai, vizualizavimo įrankiai, interaktyvi vizualizacija, duomenų tyryba

Santrumpos

PCA – Pagrindinių komponenčių analizė (angl. *Principal Component Analysis*)

MDS – Daugiamatės skalės (angl. *Multidimensional Scalling*)

ICA – Nepriklausomų komponenčių analizė (angl. *Independent Component Analysis*)

PC – Pagrindinės kreivės (angl. *Principal Curves*)

LLE – Lokaliai linijinis įterpimas (angl. *Locally Linear Embedding*)

RP – Atsitiktinė projekcija (angl. *Random Projection*)

Turinys

1 Įvadas

1.1 Tyrimo sritis ir problemos aktualumas

Šiuolaikiniame pasaulyje sunku būtų atrasti žmogaus veiklos sritį, kurioje nebūtų kaupiami ir analizuojami duomenys. Besivystant naujoms technologijoms, duomenų apimtys labai sparčiai didėja, tuo pačiu auga ir poreikiai analizuoti turimus duomenis [25]. Pastaruoju metu įvairiose mokslinių tyrimų srityse yra nuolat generuojami didžiuliai duomenų kiekiai, kurie yra saugomi saugyklose. Labai dažnai ne tik duomenų apimtis yra didelė, bet šie duomenys yra nuolatos atnaujinami ir papildomi naujais, be to, duomenų tipų ir šaltinių įvairovė taip pat yra labai plati. Tokie duomenys yra vadinami didžiaisiais duomenimis. Su sunkumais apdorojant ir analizuojant didžiuosius duomenis susiduriama įvairiose srityse, pavyzdžiui, medicinoje, finansų, ekonomikos srityse, inžinerijoje ir pan.

Didžiųjų duomenų analizės uždaviniams, tokiems kaip klasterizavimas, klasifikavimas, statistinė ir vizuali analizė, kuriami įvairūs metodai. Didžiųjų duomenų vizualizavimas yra vienas iš didžiausių uždavinių, su kuriais tenka susidurti duomenų analitikams ir mokslininkams, todėl kad metodai ir įrankiai, skirti įprastų duomenų vizualizavimui, yra netinkami didžiųjų duomenų vizualiai analizei [49], [107]. Vizualus didžiųjų duomenų atvaizdavimas leidžia aptikti, išrinkti ir efektyviai panaudoti naudingą informaciją. Gautas duomenų vaizdas leidžia pamatyti duomenų grupavimosi tendencijas, duomenų išskirtis. Tai gali padėti sprendžiant duomenų klasifikavimo ir klasterizavimo uždavinius.

1.2 Tyrimo objektas

Disertacijos tyrimo objektas:

- Didelės apimties daugiamačiai duomenys.
- Dimensijų mažinimo metodai didelės apimties daugiamačiams duomenims vizualizuoti.

1.3 Darbo tikslas ir uždaviniai:

Pasiūlyti integralią didelės apimties duomenų vizualios analizės metodologiją, apimančią skirtingų dimensijų mažinimo metodų taikymą, jų statistinių savybių panaudojimą metodams parinkti, duomenų paruošimą analizei bei jų klasterizavimą, ir kurią realizuojantis įrankis galėtų veikti paskirstytų skaičiavimų sistemose / debesyje.

Siekiant tikslo būtina išspręsti šiuos uždavinius:

- Analitiškai apžvelgti ir palyginti didelės apimties duomenų vizualizavimo metodus, juos įgyvendinančius įrankius bei technologijas, įgalinančias analizuoti didelės apimties duomenis.
- Išnagrinėti vizualizavimo metodų, pagrįstų dimensijų mažinimu, tikslumo įvertinimo matus. Atlikti metodų greičio ir tikslumo vertinimo tyrimą, siekiant palyginti skirtingus metodus ir pagrįsti skirtingų metodų taikymo naudą. Ištirti dimensijų mažinimo metodų pritaikymą lygiagretiesiems skaičiavimams.
- Pasiūlyti daugiapakopę didelės apimties duomenų vizualizavimo metodologiją, jos veikimo principus (skirtingų dimensijų mažinimo metodų taikymą, statistinių rodiklių panaudojimą, duomenų klasterizavimą) iliustruoti apdorojant skirtingus duomenų rinkinius.
- Sukurti programų sistemos prototipą, kuriame būtų realizuota pasiūlyta didelės apimties duomenų vizualizavimo metodologija.

1.4 Tyrimo metodai

Dimensijos mažinimo ir duomenų vizualizavimo sričių moksliniai ir eksperimentiniai pasiekimai analizuoti naudojant informacijos paieškos, analizės ir sisteminimo ir apibendrinimo metodai. Atlikta statistinė duomenų ir gautų rezultatų analizė, remiantis eksperimentinio tyrimo metodu. Analizės rezultatams įvertinti panaudotas apibendrinimo metodas.

1.5 Darbo mokslinis naujumas

1. Atlikta išsami didelės apimties duomenų vizualizavimo metodų ir juos realizuojančių įrankių lyginamoji analizė.
2. Išnagrinėti vizualizavimo metodų, pagrįstų dimensijų mažinimu, tikslumo įvertinimo matai; atlikti metodų greičio ir tikslumo vertinimo tyrimai. Išanalizuota, kaip lygiagrečiųjų skaičiavimų taikymas gali paspartinti duomenų dimensijų mažinimo bei vizualizavimo užduotis.
3. Pasiūlyta integrali daugiapakopio didelės apimties duomenų vizualizavimo metodologija, leidžianti duomenis analizuoti įvairiais pjūviais, analizę atlikti pažingsniui, kiekviename etape pasirenkant dominančią duomenų aibę ir jai pritaikant pasirinktą dimensijų mažinimo metodą, atsižvelgiant į dimensijų mažinimo greitį ir tikslumą konkrečiu atveju.

1.6 Ginamieji teiginiai

1. Pasiūlyta nauja daugiapakopio duomenų vizualizavimo metodologija leidžia pasirinkti dimensijų mažinimo metodą atsižvelgiant į metodo taikymo greitį ir tikslumą.
2. Pasiūlyta metodologija yra tinkama didelės apimties duomenų vizualizavimui pritaikant skirtingus dimensijų mažinimo metodus.

1.7 Darbo rezultatų praktinė reikšmė

Sukurtas metodologiją realizuojančios sistemos prototipas, kuris gali būti panaudotas realių duomenų analizei ir vizualizavimui.

1.8 Darbo rezultatų aprobavimas

Tyrimų rezultatai publikuoti 7 moksliniuose leidiniuose: 3 periodiniuose recenzuojamuose mokslo žurnaluose, iš jų viename leidinyje, referuojamame „Clarivate Analytics Web of Science“ duomenų bazėje ir turinčiame citavimo indeksą, ir 4 pateikiami konferencijos pranešimų medžiagoje.

Tyrimų rezultatai pristatyti ir aptarti šiose nacionalinėse ir tarptautinėse konferencijose:

1. VI-th International Workshop „Data Analysis Methods for Software Systems“, 4-6 December, 2014, Druskininkai, Lithuania. Pranešimo pavadinimas: „Challenges of Big Data Visualization“.
2. XVII mokslinė kompiuterininkų konferencija, „Kompiuterininkų dienos 2015“, 2015 m. rugsėjo 17-19 d., Panevėžys, Lietuva. Pranešimo pavadinimas: „Didelių duomenų vizualizavimo metodai ir įrankiai“.
3. VII-th International Workshop „Data Analysis Methods for Software Systems“, 3-5 December, 2015, Druskininkai, Lithuania. Pranešimų pavadinimai: „What is Big Data“, „Visual Analytics for Big Data“.
4. 22nd International Conference on Information and Software Technologies, 13-15 October, 2016, Kaunas, Lithuania. Pranešimo pavadinimas: „Parallel computing for dimensionality reduction“.
5. VII-th International Workshop „Data Analysis Methods for Software Systems“, 1-3 December, 2016, Druskininkai, Lithuania. Pranešimo pavadinimas: „Multi-level Method for Big Data Visualization“.

6. XVII mokslinė kompiuterininkų konferencija, „Kompiuterininkų dienos 2017“, 2017 m. rugsėjo 21-22 d., Kaunas, Lietuva. Pranešimo pavadinimas: „Daugiamatiškumo mažinimo metodai: greičio ir tikslumo palyginimas“.
7. 7th Nordic-Baltic Biometric Conference, 3-5 June, 2019, Vilnius, Lithuania. Pranešimo pavadinimas: „Multi-level Methodology for Massive Data Visualization“.

1.9 Disertacijos struktūra

Antrame skyriuje apžvelgta didžiųjų duomenų analizės problematika, aprašyti egzistuojantys duomenų analizės ir vizualizavimo metodai, išanalizuotos didžiųjų duomenų vizualizavimo įrankių galimybės. Skyriaus pabaigoje pateikiamas siūlomo sprendimo ir egzistuojančių įrankių palyginimas.

Trečiame skyriuje pristatomi atliktų duomenų vizualizavimo metodų tyrimų rezultatai. Kadangi siūloma didelės apimties duomenų vizualios analizės metodologija apima skirtingų dimensijų mažinimo metodų taikymą, todėl buvo atliktas dimensijų mažinimo metodų greičio ir tikslumo įvertinimas. Kitas svarbus aspektas, jog metodologiją realizuojantis įrankis turi būti pritaikomas veikti paskirstytų skaičiavimų sistemose / debesyje. Tam buvo atliki dimensijų mažinimo metodų pritaikymo lygiagretiesiems skaičiavimams tyrimai, palyginta lygiagrečiųjų skaičiavimų metodų sparta.

Ketvirtame skyriuje detaliai aprašoma siūloma daugiapakopio didžiųjų duomenų vizualizavimo metodologija, pristatomos ją realizuojančio įrankio galimybės.

Disertacijos pabaigoje pateikiamos išvados, literatūros sąrašas ir priedai.

2 Literatūros apžvalga

Šiame skyriuje nagrinėjama didžiųjų duomenų analizės problematika, egzistuojantys metodai ir įrankiai, aktualūs disertacijoje siūlomai metodologijai.

2.1 Didžiųjų duomenų analizės problematika

2.1.1 Duomenų tyrybos procesas

Disertacijoje gilinamasi į duomenų vizualizavimą, kuris yra viena iš esminių duomenų tyrybos komponenčių. Didelių duomenų problematika (išaugę duomenų kiekiai, didelis kintamumas ir t.t) reikalauja efektyvesnių sprendimų tiek duomenų vizualizavimui, tiek duomenų tyrybos procesui apskritai.

Duomenų tyryba – tai procesas, kurio metu, naudojant įvarius duomenų analizės įrankius, bandoma nustatyti ir atrasti „užslėptas“ duomenų struktūras ir ryšius [21]. Šio procesu metu naudojami statistiniai, matematiniai, dirbtinio intelekto ir mašininio mokymosi metodai atrasti naudingą informaciją ir žinias įvairiose duomenų bazėse [26]. Būtent tokia yra disertacijoje siūlomos metodologijos ir įrankio paskirtis, todėl šiame skyriuje pristatomi pagrindiniai duomenų tyrybos principai.

Duomenų tyryba yra šiuolaikinė informacijos analizės sritis. Analizės rezultatas yra naujų priklausomybių, apie kurių egzistavimą buvo, ar net nebuvo įtariama, radimas [96].

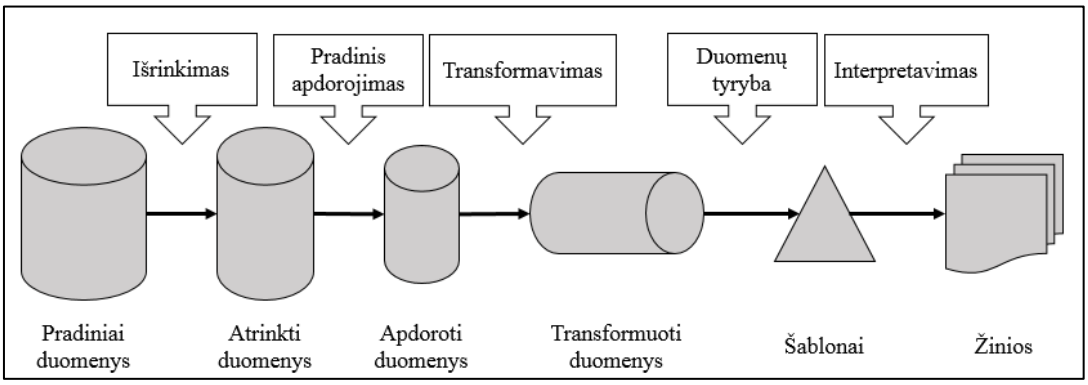
Angliškas terminas „Data Mining“ gali būti verčiamas kaip duomenų tyryba arba duomenų gavyba. Šie abu terminai literatūroje naudojami kaip tą patį reiškiančios sąvokos. Dažnai minimas dar vienas terminas „žinių radimas duomenų bazėse“ (angl. *Knowledge Discovery in Databases, (KDD)*). Žinių radimas duomenų bazėse (aibėse) apibrėžiamas kaip procesas, kurio metu ieškoma naujos informacijos didelėse duomenų bazėse (aibėse), kurios padės įgyti žinias apie analizuojamus duomenis.

Žinių radimo procesą sudaro šie žingsniai:

- Duomenų išrinkimas: iš įvairių duomenų šaltinių išrenkami analizuojami duomenys.
- Pradinis duomenų apdorojimas: analizuojami duomenys turi būti išvalyti, išfiltruoti, transponuoti, atrinkti pagal požymius, normuoti ir pan.
- Duomenų transformavimas: duomenys, surinkti iš skirtingų informacijos šaltinių, turi būti pateikiami vienoda tinkama forma.

- Duomenų tyryba: duomenų apdorojimui taikomas pasirinkto duomenų tyrybos metodo algoritmas.
- Interpretavimas/vertinimas: gaunami duomenų analizės rezultatai.

Žinių radimo duomenų bazėse schema pavaizduota 1 paveiksle.



1 pav. Žinių radimo duomenų bazėse proceso schema

Duomenų tyrybos procesas yra sudėtingas. Egzistuoja daugybė įvairių duomenų tyrybos metodų ir algoritmų. Tam, kad būtų efektyvūs, jie turi būti kruopščiai parenkami. Naudojami metodai ir algoritmai turi būti tinkamai įvertinti, siekiant užtikrinti, kad gauta informacija yra tiksli [21].

Didžiųjų duomenų tyryba suteikia galimybę gauti naudingos informacijos iš didžiųjų duomenų rinkinių. Bet dėl didelės duomenų apimties, nepastovumo, kintamumo greičio, įvairovės ir sudėtingumo yra būtini tokie duomenų tyrybos būdai, kurie būtų tinkami didiesiems duomenims. Duomenų tyrybos proceso pritaikymas didiesiems duomenims yra vienas iš didžiausių iššūkių, su kuriais susiduria duomenų analitikai ir mokslininkai [27].

2.1.2 Didžiųjų duomenų savybės

Didžiaisiais duomenimis (angl. *Big Data*) vadinami tokie duomenų rinkiniai, kuriuos dėl jų dydžio ir sudėtingos struktūros apdoroti paprastomis duomenų apdorojimo programomis ir įrankiais tampa gana sudėtinga ar net neįmanoma [99][61]. Didžiųjų duomenų sąvoka naudojama gana dažnai, tačiau ne visada ji apibrėžiama teisingai [46]. Kartais didieji duomenys charakterizuojami tik jų apimtimi. Nors pats žodis „didieji“ reiškia dydį, apimtį, tačiau didieji duomenys yra apibūdinami daugiau nei viena charakteristika:

- Apimtis – viena iš didžiųjų duomenų charakteristikų, nusakančių duomenų dydį.
- Įvairovė – charakteristika, nusakanti duomenų tipų įvairumą.
- Greitis – ši sąvoka suprantama kaip duomenų atsinaujinimo, didėjimo ir apdorojimo poreikio tenkinimas.
- Nepastovumas – charakteristika, naudojama norint apibūdinti nuolatinį duomenų kitimą ir atsinaujinimą.
- Tikrumas – ši sąvoka siejama su duomenų bei jų analizės teisingumu bei tikslumu.
- Sudėtingumas – siejamas su nuolatos augančiais duomenų kiekiais, jų įvairumu bei problemomis, atsirandančiomis analizuojant šiuos duomenis.

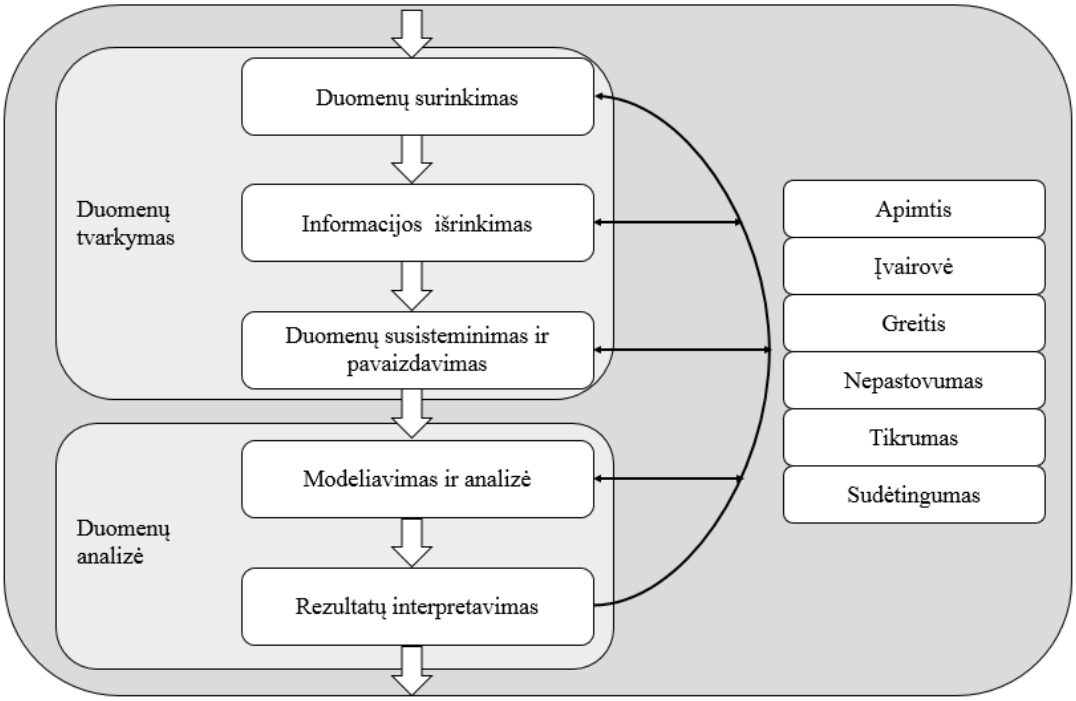
Šiame darbe pristatoma metodologija yra orientuota į didelės apimties ir įvairovės daugiamačius duomenis. Ją realizuojančio įrankio architektūra parinkta taip, kad būtų tinkama apdoroti duomenis, pasižyminčius dideliu greičiu ir nepastovumu.

Duomenų rinkinius sudaro objektai (dar vadinami elementais) ir jų parametrai (dar vadinami atributais, savybėmis, kintamaisiais, dimensijomis). Objektai, nusakomi tokiais pačiais parametrais $x_1, x_2, \dots x_n$ suformuoja duomenų rinkinį. Konkretų objektą X_i charakterizuoja visų jį nusakančių parametrų reikšmės $X_i = (x_{i1}, x_{i2}, \dots, x_{in}), i \in \{1, \dots, m\}$, kur *n* reiškia parametrų kiekį, o *m* yra objektų kiekis.

Jeigu objektai yra nusakomi daugiau nei 1 parametru, tuomet objektus charakterizuojantys duomenys yra laikomi daugiamačiais duomenimis. Jeigu yra *n* parametrų, tuomet $X_1, X_2, \dots, X_m$ yra vadinami *n*-mačiais duomenų elementais.

Jeigu duomenų rinkinį sudaro daug objektų, tuomet jis gali būti vadinamas dideliais duomenimis. Jeigu *n* yra didelis, tuomet tokie duomenys gali būti vadinami daugiamačiais duomenimis.

Duomenų analizė – pagrindinis didžiųjų duomenų uždavinys. 2 paveiksle pavaizduota didžiųjų duomenų apdorojimo schema. Pagrindiniai duomenų apdorojimo žingsniai pavaizduoti kairėje paveikslo dalyje, dešinėje pusėje išdėstytos duomenų savybės, kurios padaro duomenų apdorojimą sudėtingu ir komplikuotu. Iš schemos matyti, kad duomenų apdorojimą sudaro daug žingsnių, kiekviename iš jų susiduriama su tam tikrais uždaviniais, reikalaujančiais tinkamų sprendimo būdų.



2 pav. Didžiųjų duomenų apdorojimo schema

Pirmajame duomenų surinkimo žingsnyje turi būti nuspręsta, iš kokių šaltinių bus renkami duomenys: interneto (naršymo ar paieškos istorija, pirkimai elektroninėse parduotuvėse), mobiliųjų telefonų (diegiamos programėlės, integruoti jutikliai ir pan.), socialinių tinklų (Facebook, Twitter, Linkedln ir pan.), medicininių prietaisų (kompiuterinės tomografijos vaizdai, genetinių tyrimų duomenys ir pan.) [41]. Šiame etape turi būti pasirūpinta duomenų nuasmeninimu ir kitais duomenų tvarkymo teisiniais dalykais. Duomenų analizės etapui turi būti atrinkta reikiama informacija, išgryninta, susieta su galbūt jau turimais duomenimis, surinktais iš kitų šaltinių, ir viskas susisteminta. Paskutinis rezultatų interpretavimo žingsnis yra svarbus tuo, kad skelbiami rezultatai turi būti patikimi, patikrinti bei teisingi.

Didžiųjų duomenų saugojimas ir apdorojimas skiriasi nuo tradicinių duomenų analizės būdų. Kai pavienių kompiuterių išteklių neužtenka, į pagalbą yra pasitelkiamos paskirstytosios ir lygiagrečiosios sistemos arba paskirstytoji duomenų tyryba (angl. *Distributed Data Mining*). Analizuojamus duomenis suskirsčius tam tikrais būdais, duomenų tyrybos uždavinys lygiagrečiai sprendžiamas kompiuterių klasteriuose ar griduose. Kompiuterių klasteris – tai į vieną bendrą tinklą sujungti kompiuteriai, kurie geba vykdyti paskirstytus skaičiavimus. Gridas – tai kaip ir klasteris yra laisvai prieinama, suderinta infrastruktūra, tačiau ją sudaro atskiri skaičiavimo klasteriai [11]. Pagrindinis šių sistemų principas – „skaldyk ir valdyk“. Yra stengiamasi didelę užduotį

suskaidyti į smulkesnes, žymiai lengviau išsprendžiamas ir nepriklausomas dalis, kurios yra vykdomos lygiagrečiai. Tarpiniai rezultatai sujungiami, ir gaunamas galutinis rezultatas. Vienas charakteringas tokių sistemų pavyzdys – *Apache Hadoop* programinė įranga.

2.2 Duomenų analizės ir vizualizavimo metodų apžvalga

Duomenų analizė ir vizualizavimas – viena iš problemų, su kuria susiduriama daugelyje sričių. Kuo didesnė duomenų apimtis, tuo tampa sunkiau juos analizuoti ir suvokti objektų visumą bei jų savybes ir ypatybes. Duomenims analizuoti į pagalbą pasitelkiamas vizualizavimas. Didelių duomenų klasterizavimas į atskiras grupes leidžia analizuoti mažesnes duomenų aibes, nustatyti jų savybes bei tarpusavio ryšius.

Šiame skyriuje apžvelgiami daugiamačių duomenų **projekcijos** bei **klasterizavimo** metodai.

2.2.1 Projekcijos metodai

Projekcijos metodai – tai metodai, taikomi daugiamačiams duomenims transformuoti į mažesnio skaičiaus matmenų erdvę. Projekcijos metodai dar gali būti vadinami matmenų skaičiaus mažinimo metodais (angl. *Dimensional Reduction Techniques*). Yra išskiriami tiesinės ir netiesinės projekcijos metodai.

Pagrindinis projekcijos metodų tikslas – pateikti daugiamačius duomenis mažesnio skaičiaus matmenų erdvėje taip, kad būtų kiek galima tiksliau išlaikyta duomenų struktūra.

Būtent ši metodų grupė yra naudojama siūlomoje daugiapakopio duomenų vizualizavimo metodologijoje, todėl visame darbe jiems skiriama daugiausia dėmesio.

Straipsnių [28], [75], [102] autoriai atliko detalias dimensijų mažinimo metodų apžvalgas. Autoriai pabrėžia nuolatos augantį duomenų kiekį ir tradicinių metodų ribotumą juos apdorojant. Tai lemia didėjantį poreikį dimensijų mažinimo metodams. Nors duomenų kiekiai sparčiai didėja, tačiau dažnai patys duomenys yra pasikartojantys ar turintys mažai informacijos. Todėl dimensijų mažinimo metodai gali sėkmingai sumažinti duomenų apimtis neprarandant vertingos informacijos.

Toliau pateikiama trumpa santrauka metodų, kuriuos analizavo minėti autoriai, ir kurie yra taikomi disertacijoje pristatytuose tyrimuose ir metodologijoje.

2.2.1.1 Pagrindinių komponenčių analizė

Pagrindinių komponenčių analizė (angl. *Principal Component Analysis, (PCA)*) yra vienas populiariausių dimensijų mažinimo metodų. Šis metodas gerai įvertina duomenų ypatumus, kai jie turi tiesinę struktūrą.

PCA metodas randa tas komponentes, kurių projekcijos yra mažiausiai koreliuotos. Tokiu būdu randamos komponentės, aplink kurias yra didžiausias duomenų pasiskirstymas.

Esminė PCA idėja yra sumažinti duomenų matmenų skaičių atliekant tiesinę transformaciją ir atsisakant dalies po transformacijos gautų naujų komponenčių, kurių dispersijos yra mažiausios. Iš pradžių ieškoma krypties, kuria dispersija yra didžiausia. Didžiausią dispersiją turinti kryptis vadinama pirmąja pagrindine komponente. Ji eina per duomenų centrinį tašką. Tai taškas, kurio komponentės yra analizuojamą duomenų aibę sudarančių taškų atskirų komponenčių vidurkiai. Visų taškų vidutinis atstumas iki šios tiesės yra minimalus, t. y. ši tiesė yra kiek galima arčiau visų duomenų taškų. Antrosios pagrindinės komponentės (PK2) ašis taip pat turi eiti per duomenų centrinį tašką ir ji turi būti statmena pirmosios pagrindinės komponentės ašiai [25].

Pagrindinės komponentės yra lygties $CE = \lambda E$ sprendinys. Šioje lygtyje E yra vektorius-stulpelis, C yra duomenų kovariacinė matrica $C = \{c_{kl}, k, l = 1, \dots, m\}$, ir λ – tikrinė reikšmė, randama iš charakteringos lygties $|C - \lambda I| = 0$. Čia I yra vienetinė matrica, kurios matmenys tokie patys, kaip ir matricos C. Pagrindinėms komponentėms nustatyti užtenka rasti d didžiausių matricos C tikrinių reikšmių ir jas atitinkančių tikrinių vektorių. Tada duomenų aibės $X = \{X_1, \dots, X_n\}$ taško $X_i \in \mathbb{R}^d$ transformacija $Y_i \in \mathbb{R}^d$ mažesnės dimensijos erdvėje randama pagal formulę: $Y_i = (X_i - \bar{X})A$, čia $X_i = (x_{i1}, \dots, x_{im}), \bar{X} = (\bar{x}_{i1}, \dots, \bar{x}_{im})$ – požymių, kuriais apibūdinamas kiekvienas taškas, vidurkiai. $A = (E_1, \dots, E_m)$ – pagrindinių komponenčių matrica [82].

2.2.1.2 Daugiamatės skalės

Daugiamatės skalės (angl. *Multidimensional Scalling, (MDS)*) – grupė metodų, skirtų daugiamačių duomenų projekcijų mažesnio skaičiaus matmenų erdvėje paieškai. Dažniausiai projekcijų ieškoma dvimatėje arba trimatėje erdvėse. Vienas iš projekcijos paieškos tikslų – išlaikyti analizuojamos aibės objektų panašumus arba skirtumus.

Turint pradinius duomenis iš *n* elementų ir *d* dimensijų bei *n x n* panašumų tarp elementų matricą, daugiamačių skalių metodas (MDS) duomenis atvaizduoja naujoje

k-dimensijų ($k \leq d$) erdvėje kiek įmanoma labiau išlaikant panašumą tarp taškų pradinėje ir naujoje erdvėje.

Panašumas iš esmės reiškia atstumą: kuo du taškai yra panašesni vienas į kitą, tuo arčiau vienas kito jie yra. Populiariausi atstumų matai yra Euklido ir Manheteno atstumai. Tikslumo įvertinimui gali būti skaičiuojama paklaidos funkcija E_{DS}, dar vadinama Stress funkcija, pagal formulę:

$$E_{DS} = \frac{\sum_{i<j}(d(X_i, X_j) - d(Y_i, Y_j))^2}{\sum_{i<j}(d(X_i, X_j))^2}$$

Čia kiekvieną *n*-matį vektorių $X_i \in R^d, d < n, i \in \{1, \dots, m\}$, atitinka mažesnio skaičiaus matmenų vektorius $Y_i \in R^d, d < n$. Atstumas tarp vektorių X_i ir X_j žymimas $d(X_i, X_j)$, o atstumas tarp vektorių Y_i ir $Y_j - d(Y_i, Y_j), i, j = 1, \dots, m$, $(X_i, X_j) \neq 0$.

2.2.1.3 Nepriklausomų komponenčių analizė

Nepriklausomų komponenčių analizė (angl. *Independent Component Analysis, (ICA)*) – tai viena iš daugiamatės analizės metodikų, kurios pagrindinis tikslas yra išskirti nepriklausomus iš jų mišinio. Tai klasikinis „aklojo atskyrimo“ (angl. *Blind Source Seperation*) metodas, taikomas tuomet, kai turimas duomenų mišinys ir siekiama iš jo išskirti pradinius originalius duomenis, nežinodami nieko apie mišinį ir jo sudedamąsias dalis [26].

Nepriklausomų komponenčių analizė yra aukštesnio lygio metodas, ieškantis tiesinių projekcijų, nebūtinai statmenų viena kitai, tačiau kaip įmanoma labiau statistiškai nepriklausomų. Statistinė nepriklausomybė yra žymiai stipresnė sąlyga nei koreliacijos nebuvimas. ICA gali būti laikomas bendresniu PCA ir *Projection Pursuit* atveju. Jeigu PCA ieško nekoreliuotų kintamųjų, tai ICA ieško nepriklausomų kintamųjų.

2.2.1.4 Pagrindinės kreivės ir daugdaros

PCA metodas puikiai tinka dimensijų mažinimui tuomet, kai pradiniai k-dimensijų duomenys yra išsidėstę ant tam tikros tiesinės daugdaros (angl. *Manifold*). Tačiau pasitaiko situacijų, kai originalūs duomenys yra išsidėstę netiesinėje erdvėje. Tokiu atveju kreivės aproksimavimas į tiesę neduotų gerų rezultatų aproksimuojant pradinius duomenis. Dirbant su tokio tipo duomenimis naudojamos pagrindinės kreivės (angl. *Principal Curves, (PC)*) ir daugdaros (angl. *Manifolds*) [60].

Pagrindinių kreivių paieška yra svarbus duomenų analizės metodas. Atradus duomenis atitinkančia kreivę, dimensijų kiekis galia būti sumažintas naudojant netiesinius metodus.

2.2.1.5 Lokaliai linijinis įterpimas (LLE)

Lokaliai linijinis įterpimas (angl. *Locally Linear Embedding, (LLE)*), kaip ir Isomap (žiūrėti žemiau), MDS bei *Kernel* PCA, yra paremti tikrinių vektorių naudojimu.

LLE metodas naudojamas atrasti duomenis atitinkančias daugdaras ir atvaizduoti duomenis ant jų. Kiekvienam duomenų elementui ieškoma K-artimiausių kaimynų ir apskaičiuojami svoriai jo aproksimavimui. Toks aproksimavimas vienu metu atliekamas visiems elementams. Kai visi svoriai nustatomi, tuomet ieškoma taškų mažesnio dimensijų skaičiaus erdvėje. Nauji taškai suskaičiuojami pagal kaimyninius taškus tokiu pačiu principu (naudojant tokius pačius svorius) kaip ir pradiniai taškai [25].

2.2.1.6 Isomap

Jeigu atstumai tarp taškų matuojami kai geodeziniai atstumai, tuomet tokia MDS metodo atmaina vadinama Isomap. Geodezinis atstumas tarp dviejų taškų ir daugdaros matuojamas palei daugdaros paviršių. Praktiškai jis yra skaičiuojamas kaip trumpiausias kelias kaimyninių elementų grafe, jungiančiame kiekvieną elementą su jo K-artimiausiais kaimynais [25].

2.2.1.7 Atsitiktinė projekcija

Atsitiktinės projekcijos (angl. *Random Projection, (RP)*) metodas leidžia daug dimensijų turinčius pradinius duomenis atvaizduoti mažesnio dimensijų skaičiaus erdvėje išlaikant pradinius atstumus tarp objektų. Taikant šį metodą yra atliekama pradinės duomenų matricos daugyba iš atsitiktinai sugeneruotų skaičių matricos. Jeigu pradiniai duomenys yra $n \times d$ matrica, tuomet norint gauti projekciją yra panaudojama „tinkama“ $d \times k$ matrica R. Pradinių duomenų A projekciją E galima pažymėti kaip $E = A \cdot R$. Naujoji matrica E aproksimuoja pradinius duomenis į k-dimensijų (matrica E yra $n \times k$) [70].

R matrica yra sudaryta iš r_{ij} elementų. R gali būti konstruojama keletu būdu [19]:

- $r_{ij} = \pm 1$ su tikimybe $1/2$,

- $r_{ij} = \pm 1$ su tikimybe $1/6$, arba 0 su tikimybe $2/3$.

Steve Vincent (2004) palygino atsitiktinės projekcijos metodą su kitu plačiai taikomu metodu – pagrindinių komponenčių analize (angl. *Principal Component Analysis, (PCA)*). Šie metodai buvo pritaikyti teksto apdorojimui. Rezultatai atskleidė, kad atsitiktinės projekcijos metodas daugeliu atveju yra greitesnis, tačiau mažiau tikslus. Todėl patarimas buvo naudoti jį tuomet, kad kai svarbiausia yra greitis [19].

Šiame darbe atsitiktinės projekcijos metodas naudotas ištirti, kaip lygiagrečių skaičiavimų taikymas gali padidinti duomenų analizės greitį.

2.2.2 Duomenų klasterizavimo metodai

Didelio duomenų rinkinio apdorojimas, sprendžiant pakankamai tiksliai ir pakankamai greitai įvairius uždavinius, yra vienas iš pagrindinių duomenų tyrybos uždavinių, diegiant interaktyvias analitinio apdorojimo sistemas (angl. *On-line Analytical Processing, (OLAP)*). Apibrėžiant terminą „didelis duomenų rinkinys“, reikia atsižvelgti į turimos aparatinės (angl. *Hardware*) ir programinės (angl. *Software*) įrangos našumą, taip pat į sprendžiamą uždavinį, nes sudėtingesniems duomenų modeliams realizuoti net ir palyginti nedidelė duomenų apimtis gali sudaryti rimtų problemų. Vienas iš sprendimo būdų yra naudoti našesnę aparatinę ar programinę įrangą, tačiau šis sprendimo būdas praktikoje ne visada prieinamas. Kitas sprendimo būdas – pakeisti pradinį duomenų rinkinį mažesniu, pagal galimybes išlaikant pradines duomenų rinkinio savybes. Trivialus šio sprendimo būdo pavyzdys yra atsitiktinis mažesnės apimties duomenų imties išrinkimas. Tačiau šiuo atveju, sprendžiant analizės uždavinius, atitinkamai padidėja parametrų įverčių dispersija, į ką būtina atsižvelgti. Imčių metodai yra labai plačiai taikomi tais atvejais, kai duomenų rinkimo sąnaudos yra didelės (pvz., demografinėje statistikoje, ūkinėje statistikoje, sociologiniuose tyrimuose ir pan.). Tarkime, kad yra pateiktas didelis duomenų rinkinys, ir reikia pakeisti šį duomenų rinkinį mažesniu, maksimaliai išlaikant pagrindines duomenų rinkinio savybes ir panaudojant informaciją iš visų duomenų rinkinio elementų. Pradinio duomenų rinkinio pakeitimui mažesniu yra naudojami įvairūs klasterizavimo metodai, kuriuos galima suskirstyti į dvi grupes – skaidymo (angl. *Partitioning*) ir hierarchiniai (angl. *Hierarchical*) metodai. Skaidymo algoritmai suskaido duomenų rinkinį į klasterius, hierarchiniai algoritmai pateikia hierarchinę klasterinę struktūrą, tačiau neapibrėžia pačių klasterių išreikštiniu pavidalu.

Pastaruoju metu dažnai naudojamas straipsnyje [20] pasiūlytas duomenų sutraukimo (angl. *Data Squashing*) metodas. Šis metodas siekia sutraukti (angl. *Squash*) duomenis tokiu būdu, kad statistinė analizė, atliekama naudojant sutrauktus duomenis, duotų kiek įmanoma panašesnius rezultatus, kaip ir naudojant visą duomenų rinkinį. Tokiu būdu duomenų analizę galima atlikti su sutrauktais duomenimis įprastais metodais ir gauti žymiai tikslesnius rezultatus, nei naudojant panašaus dydžio išrinktą atsitiktinę imtį [47].

Madigan metodas stengiasi sukurti sutrauktą duomenų rinkinį, kuris tiesiogiai aproksimuoja specifinę tikėtinumo funkciją, vietoj to, kad remtųsi bendruoju Teiloro eilučių argumentu, aproksimuojančiu visas įmanomas tikėtinumo funkcijas. Autorius pasirinko fiksuoto atsako kintamojo logistinę regresiją kaip specifinį modelį, remiantis kuriuo ir vykdomas duomenų sutraukimas. Tokiu būdu gautas sutrauktas duomenų rinkinys gali nebūti toks naudingas visiems įmanomiems analizės tipams kaip bendruoju būdu gautas duomenų rinkinys, tačiau jo pranašumas išryškėja atliekant logistinės regresijos tipo analizę. Aproksimavimo metodas apima paiešką tikėtinumo profilio, kuris yra logistinės tikėtinumo funkcijos reikšmių vektorius su K reikšmėmis p-matėje logistinės regresijos koeficientų erdvėje. K parametrų vektoriai turi būti parinkti taip, kad p kintamųjų glodi (angl. *Smooth*) funkcija galėtų būti apytikriai identifikuota pagal atitinkamas K funkcijos reikšmes.

Vienas iš plačiai naudojamų metodų klasifikavime ir regresinėje analizėje yra CART (angl. *Classification and Regression Tree*) metodas [36]. Šiame metode nagrinėjama erdvės sritis (daugiamatis stačiakampis gretasienis) yra pažingsniui skaidoma į padalijimus, kol patenkinama tam tikra sustojimo sąlyga. Naudojant grafų teorijos sąvokas, CART metodo atliekamus žingsnius galima pavaizduoti kaip medį (angl. *Tree*), prasidedantį iš šakninės viršūnės (angl. *Root Node*), iš kurio tam tikros viršūnės (angl. *Node*) konkrečiame žingsnyje išeina dvi ar daugiau kraštinių (angl. *Edges*), besibaigiančių atitinkamomis viršūnėmis. Jei kiekviename žingsnyje padalijimo elementas skaidomas į dvi dalis (angl. *Binary Partitions*), tai turime binarinio medžio (angl. *Binary Tree*) metodą. Šis metodas daugiausia taikomas (kaip galima spręsti iš jo pavadinimo) klasifikavimui ir regresinei analizei.

Šiame darbe pristatomas įrankis naudoja tokius klasterizavimo metodus:

- K-vidurkių (k-means). Tai yra vienas iš nehierarchinių klasterinės analizės metodų. Nehierarchiniai metodai paprastai taikomi tada, kai iš

anksto žinomas (pasirenkamas) klasterių skaičius ir norima klasterizuoti tiriamus objektus. Klasterizavimo procedūrą sudaro tokie žingsniai: 1) objektai suskirstomi į K pradinių klasterių; 2) paeiliui apskaičiuojamas kiekvieno objekto atstumas iki klasterių centrų (atstumas paprastai skaičiuojamas naudojantis Euklido metrika arba jos kvadratu); objektas priskiriamas artimiausiam klasteriui; perskaičiuojami klasterių centrai; 3) algoritmas kartojamas tol, kol nė vieno objekto nereikia priskirti naujam klasteriui [14].

- *Dbscan*. Šis metodas ieško tokių klasterių, kurių kiekvieno taško aplinkoje (pagal pasirinktą spindulį) būtų bent jau minimalus taškų kiekis (jis yra nurodomas kaip algoritmo parametras). Taip yra atskiriami tankūs erdvės regionai nuo retesnių ir nustatoma klasterizavimo struktūra [68].
- *Clara* (angl. *Clustering Large Applications*) yra *k-medoids* (PAM) metodo išplėtimas, skirtas apdoroti duomenims, turintiems daug objektų (daugiau nei keletą tūkstančių). Jis leidžia sumažinti skaičiavimų laiką ir poreikį RAM. Clara atveju metoidų yra ieškoma ne visame duomenų rinkinyje, tačiau tik mažoje fiksuotoje srityje. PAM algoritmas naudojamas optimaliam metoidų rinkiniui rasti [15].
- *Optics* (angl. *Ordering Points to Identify the Clustering Structure*) kaip ir *dbscan* yra tankio metodas. Jis aprašo klasterius kaip tankius regionus duomenų erdvėje [1].

2.3 Susijusių mokslinių darbų analizė

Duomenų analizės ir vizualizavimo temos plačiai nagrinėjamos tiek pasaulio, tiek Lietuvos mokslininkų [45], [55], [84], [35], [17]. Pastaruoju metu didžiausias dėmesys skiriamas problemoms, susijusioms su didžiųjų duomenų apdorojimu.

Lietuvos mokslininkai yra paskelbę eilę tyrimų, susijusių su daugiamačių duomenų tyryba, jų dimensijų mažinimu, vizualizavimu. J. Bernatavičienė savo disertacijoje [3] pasiūlė vizualios žinių gavybos metodologiją, leidžiančią atlikti išsamią ir informatyvią tiriamų duomenų analizę. Taip pat ji detaliai ištyrė daugiamačių skalių metodą ir pasiūlė bazinių vektorių (angl. *Basic Vectors*) parinkimo ir jų skaičiaus nustatymo būdus taikant santykinių daugiamačių skalių metodą (angl. *Relative*

Multidimensional Scaling). Šiame darbe analizuojama didesnė aibė dimensijų mažinimo metodų ir kartu sprendžiama problema, kaip konkrečiu atveju parinkti tinkamiausią iš jų.

R. Karbauskaitė savo disertacijoje [53] nagrinėjo daugiamačių duomenų vizualizavimo algoritmus ir metodus, išlaikančius lokalią struktūrą. Autorė analizavo tris daugiamačių duomenų projekcijų mažesnės dimensijos erdvėje vertinimo kriterijus: Spirmano koeficientą, Konigo matą ir kaimynystės klaidos koeficientą. Šioje disertacijoje analizuojami ir tyrimuose naudojami Stress matas, Spirmano koeficientas ir Šenono entropija.

G. Dzemyda, O. Kurasova ir J. Žilinskas monografijoje [22] plačiai išnagrinėjo įvairius daugiamačių duomenų vizualizavimo metodus, jų modifikacijas ir taikymus.

O. Kurasova su bendraautoriais tyrinėjo daugiamačių duomenų vizualizavimą taikant dirbtinius neuroninius tinklus [24], [23]. V. Medvedev daktaro disertacijoje [69] taip pat nagrinėjo tiesioginio sklidimo dirbtinius neuroninius tinklus, skirtus daugiamačiams duomenims vizualizuoti. V. Marcinkevičiaus [67] ištyrė daugiamačių duomenų atvaizdavimą netiesiniais daugiamačių skalių algoritmais ir saviorganizuojančiais neuroniniais tinklais.

A. Žilinskas ir J. Žilinskas tyrinėjo daugiamačių skalių metodą su miesto kvartalų metrika ir ieškojo daugiamačių skalių paklaidos (įtempimo, tikslo) funkcijos globalaus minimumo [114], [115], [112], [113].

K. Paulauskienė tyrė dimensijų mažinimo metodus didelės apimties daugiamačiams duomenis vizualizuoti ir projekcijos paklaidų įvertinimą. Savo disertacijoje ji pasiūlė didelės apimties duomenų aibės vizualizavimo strategiją, leidžiančią išvengti duomenų aibės taškų persidengimo ir išlaikyti bendrą duomenų struktūrą [82]. Pasiūlyta strategija yra orientuota į kuo tikslesnį pradinės duomenų aibės vizualizavimą, tačiau ji nesprendžia problemos, kaip išlaikyti vizualizavimo tikslumą analizuojant atskiras duomenų sritis ir tam taikant skirtingus dimensijų mažinimo metodus.

Užsienio mokslininkai yra paskelbę darbų, kuriuose didžiųjų duomenų analizės ir vizualizavimo problemą sprendžia kompleksiškai.

Sacha D., Zhang L. ir kt. atliko su duomenų vizualizavimu ir dimensijų mažinimu susijusios mokslinės literatūros analizę. Pagrindinis tikslas buvo sistematiškai ištirti ir įvertinti, kaip duomenų analitikai naudoja dimensijų mažinimo

metodus. Autoriai išskyrė septynis tipinius veiksmus, atliekamus analizės metu, pvz. pasirinkimas tarp keleto dimensijų mažinimo metodų, algoritmų parametrų apibrėžimas, reikšmingų komponenčių nustatymas. Svarbu, kad dimensijų mažinimas būtų atliekamas interaktyviai [95].

L. Kuang, L. T. Yang ir kt. sprendė tris problemas, susijusias su didžiųjų duomenų analize: pradinių duomenų šaltinių apjungimas, duomenų dimensijų mažinimas, paskirstytų skaičiavimų sistemos konstravimas. *Chunk Tensor* metodas naudotas nestruktūrizuotų, dalinai struktūrizuotų ir struktūrizuotų duomenų apjungimui į vieningą duomenų modelį. Lanczos algoritmu paremtas metodas naudotas dimensijų mažinimui. Skaidraus skaičiavimo (angl. *Transparent Computing*) paradigma naudota konstruojant paskirstytų skaičiavimų platformą [58].

X. Qin, Y. Luo, N. ir kt. pasiūlė automatinio didžiųjų duomenų vizualizavimo metodologiją. Siekiama automatizuoti tokias užduotis kaip duomenų užkrovimas, filtravimas, transformavimas ir tinkamo vizualizavimo būdo parinkimas. Pagrindinės sprendžiamas problemos: 1) kaip nustatyti ar parinktas vizualizavimo būdas yra suprantamas analitikui, 2) kaip dideliame duomenų rinkinyje, kuris visas savaime neduoda vertingos informacijos, surasti reikšmingas duomenų dalis, 3) kaip optimalų vizualizavimo būdą parinkti naudojant kuo mažiau bandymų (neverčiant analitiko gaišti laiko). Pirmos problemos sprendimui naudojamas binarinis klasifikatorius. Grupavimo technikos naudojamos antros problemos sprendimui. Siekiant pagreitinti rezultatų pateikimą naudojamas skaičiavimo užduočių paskirstymas [84].

Didžiųjų duomenų vizualizavimas yra resursams labai imlus procesas. Darbe [101] pristatyta platforma ir metodika, pritaikyta didžiųjų geografinių duomenų vizualizavimui. Autoriai savo darbe apjungė racionalų fizinių išteklių panaudojimą bei metodų pritaikymą intensyviems skaičiavimams [101]. Metodų pritaikymą optimaliam fizinių resursų panaudojimui akcentavo ir M. Nazemi [78].

Disertacijoje pristatomas įrankis taip pat pasižymi tokia architektūra, kad galėtų būti naudojamas paskirstytose sistemose ir gebėtų atlikti sudėtingus bei intensyvius skaičiavimus.

Straipsnyje [2] pasiūlytas sprendimas, kaip teisingiau vizualiai įvertinti grafiškai pateikiamų rezultatų teisingumą. Tačiau svarbu atkreipti dėmesį, kad vizualaus vertinimo dažnu atveju nepakanka, o kiekybiniai rodikliai ženkliai palengvina rezultatų vertinimą.

Z. Lai analizavo skirtingus tiesinio dimensijų mažinimo metodus, tam kad pasiūlyti apibendrintą dimensijų mažinimo metodologiją [62]. Pagrindinis pasiūlyto metodo tikslas yra eliminuoti metodų jautrumą nuokrypiams ir vaizdų variacijoms.

M. Harandi pristatė algoritmus, gebančius apdoroti daugiadimensines SPD (angl. *Symmetric Positive Definite*) matricas sukuriant mažesnio dimensijų kiekio SPD daugdaras, dimensijų mažinimui panaudojant ortonormalią projekciją. Ieškant optimalios projekcijos buvo sprendžiamas optimizavimo uždavinys ant Grassmann‘o daugdaros [34].

Galima išskirti atskirą grupę mokslinių tyrimų, kuriuose dimensijų mažinimo metodai naudojami tam tikros taikomosios srities problemoms spręsti.

D. Kaur, G. S. Aujla ir kt. tyrė didžiųjų duomenų, kuriuos generuoja elektros tinklams priklausantys išmanieji įrenginiai, problemą. Dimensijų mažinimui autoriai siūlė naudoti tenzoriais paremtą didžiųjų duomenų apdorojimo metodą [54].

S. Doerr taikė dimensijų mažinimo metodus molekulių tyrimuose, nes molekulių simuliacijų metu sukuriami daugiamačiai duomenų rinkiniai, kurių nepajėgia apdoroti standartiniai duomenų analizės metodai. Šiuo atveju spręsta problema, kad pritaikius dimensijų mažinimo metodus neaptinkamos svarbiausios komponentės, reikalingos Morkovo modelio konstravimui. Autoriai taikė įvarius dimensijų mažinimo metodus (pvz. PCA, ICA) dviejų duomenų rinkinių vizualizavimui [18].

X. Zhong ir kt. PCA, FRPCA ir KPCA dimensijų mažinimo metodus pritaikė prognozuojant akcijų rinkos pokyčius. Autorių pasiūlyta duomenų analizės metodika derina dimensijų mažinimo metodus su neuroninių tinklų taikymu. Atlikti bandymai naudojant 10 metų istorinius duomenis, apimančius 60 finansinių ir ekonominių rodiklių, parodė, kad PCA ir dirbtinio neuroninio tinklo derinimas leidžia pasiekti didesnį pelningumą [108].

M. Nilashi (2018) dimensijų mažinimo technikas pritaikė rekomendacijų (sprendimo paramos) sistemai. Singular Value Decomposition (SVD) taikymas leido išspręsti dvi pagrindines problemas: „sparsity and scalability“ [79].

O. Claveria ir kt. (2017) dimensijų mažinimą pritaikė turistų lankomų vietų pozicionavimui ir klasterizavimui. Pagal pasiūlytą metodą vietovės iš pradžių yra reitinguojamos, tuomet klasterizuojamos pagal poziciją reitinge, ir galiausiai yra sukuriami žemėlapiai panaudojant CATPCA ir MDS metodus [12].

J. Gui ir kt. (2018) išplėtė daugiafaktorinį dimensijų mažinimo (MDR) algoritmą ir pritaikė jį vėžio tyrimams [33]. Panašius tyrimus atliko ir C. Yang ir kt. (2018), vėžio tyrimams naudoję daugiaobjektį (angl. *Multiobjective*) MDR metodą – MOMDR [44].

T. Hou pasiūlė apibendrintą MDR variantą (angl. *Generalized Multifactor Dimensionality Reduction, (GMDR)*), kurį pritaikė genetiniams tyrimams. Kaip ir kitais atvejais, pagrindinis tikslas yra pasiekti kuo didesnį tikslumą [40].

N. F. Chikhi ir kt. panaudojo PCA, NMF, ICA ir RP metodus tiriant saityno struktūrą bei pateikė jų lyginamąją analizę [9]. B. Tang teksto analizei panaudojo ir palygino Nepriklausomų komponenčių analizės (ICA), Latentinio semantinio indeksavimo (LSI), Atsitiktinės projekcijos (RP) ir kitus metodus [105]. Šiuo atveju geriausi rezultatai buvo pasiekti naudojant ICA ir LSI. A. Galletta pasiūlė metodologiją didžiųjų duomenų vizualizavimui telemedicinos srityje [31].

Įvairiuose tyrimuose naudoti metodai ir gauti rezultatai skiriasi – nėra vieningos išvados, jog tam tikras metodas būtų geriausias visose situacijose. 1 lentelėje analizuoti moksliniai darbai yra sugrupuoti pagal tematiką, taip pat nurodyta, kaip jie siejasi su šia disertacija.

1 lentelė. Literatūros apžvalga

Tematika	**Straipsniai**	**Sąsaja su disertacija**
Didžiųjų duomenų problematika	[5][100]	Disertacijoje taip pat nagrinėjama didžiųjų duomenų vizualizavimo problematika
Duomenų vizualizavimo metodologijos	[3][82] [58] [84] [95][62][101][98]	Disertacijoje pasiūlyta daugiapakopė duomenų vizualizavimo metodologija, leidžianti kiekviename žingsnyje taikyti skirtingus dimensijų mažinimo metodus
Bendrieji dimensijų mažinimo metodų tyrimai	[22][53][114][115][112][113] [24][23][67][69] [34]	Disertacijoje pristatomi dimensijų mažinimo metodų greičio ir tikslumo tyrimų rezultatai
Dimensijų mažinimo metodų tyrimai juos pritaikant specifinėms sritims	[54] [18][108][79][12] [33][44][40][105] [31]	Disertacijoje dimensijų mažinimas pritaikytas skirtingo tipo atsitiktinai generuotiems ir realiems finansiniams duomenims
Racionalus fizinių išteklių panaudojimas vizualizuojant duomenis	[101][78]	Disertacijoje ištirtas dimensijų mažinimo metodų pritaikymas lygiagretiems skaičiavimams
Vizualizavimo įrankių pristatymas	[13][4] [51] [52] [30][94][7][42][57]	Mokslinėje literatūroje pristatomų vizualizavimo įrankių apžvalga ir palyginimas pateikiami 2.4.1 skyriuje

Vaizdo priartinimas	[13] [52]	Disertacijoje pristatomas įrankis palaiko vaizdo priartinimą bei įgalina pritaikyti naujus dimensijų mažinimo metodus priartintoms duomenų sritims
Esminių komponenčių radimas	[13][4][51][52]	Disertacijoje siūloma metodologija leidžia rasti analitiniu požiūriu svarbias duomenų sritis ir analizuoti klasterių parametrus

2.4 Duomenų vizualizavimo įrankių apžvalga

Svarbus didelės apimties duomenų apdorojimo etapas yra jų vizualizavimas. Šiame etape nereikėtų daryti tokių klaidų:

- *Nevizualizuoti visų duomenų.* Duomenų vizualiai analizei turi būti atrinkti tik tie duomenys, kurie yra reikalingi. Pertekliniai, neturintys didelės reikšmės duomenys vizualiai analizei neturi būti atrenkami.
- *Nevizualizuoti „neteisingų" duomenų.* Vizualizavimui turi būti naudojami tik tarpusavyje glaudžiai susiję duomenys. Turi būti apsvarstyta, kurie duomenys turi didžiausią įtaką atliekamam tyrimui.
- *Pateikti duomenis tvarkingai*. Labai svarbu tinkamai parinkti duomenų vizualizavimo metodą: grafikas, diagrama, matrica, žemėlapis ir pan. Duomenis patogu pateikti sugrupuotus, surūšiuotus pagal dydį, svarbą ir kt., naudoti spalvas kategorijoms ar klasteriams žymėti [74].

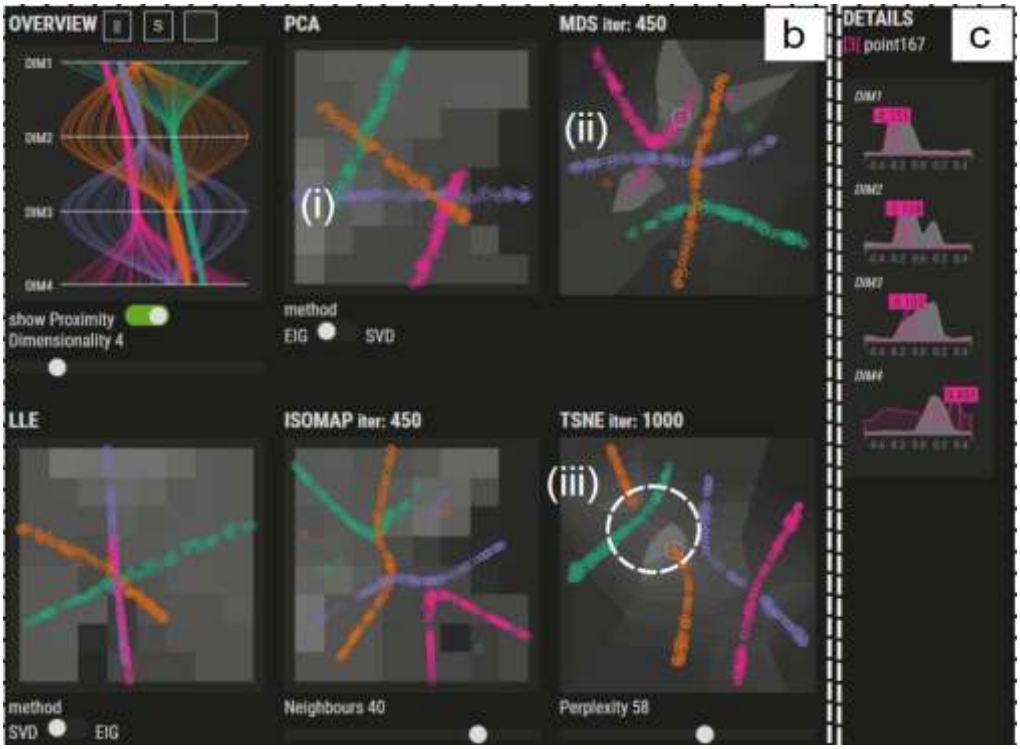
Su didžiųjų duomenų vizualios analizės uždaviniais gali susidoroti ne visi duomenų vizualizavimo įrankiai. Todėl svarbu iš gausybės siūlomų įrankių parinkti tinkamą.

2.4.1 Mokslinėje literatūroje pristatomų įrankių apžvalga

N. Bitakis atliko didžiųjų duomenų vizualizavimo įrankių apžvalgą, pagrinde orientuojantis į jų savybes. Autorius pabrėžė hierarchinio analizės principo svarbą. Tai reiškia, jog vartotojui pirmiausia reikia patekti apibendrintą vaizdą ir informaciją, vėliau sudarant galimybę atlikti išsamią analizę (pvz. *Roll-up, Drill-down, Zoom, Filtering*), ir galiausiai pateikiant detales apie analizuojamus duomenų rinkinius [4]. Visų šių principų buvo laikomasi kuriant disertacijoje siūlomą metodologiją ir įrankį.

R. Cutura (2018) pristatė įrankį VisCoDeR, skirtą vizualiam skirtingų dimensijų mažinimo metodų palyginimui. Pagrindinis tikslas yra atskleisti metodų panašumus ir skirtumus. Naudojami du rėžimai – kokybiniam vizualiam grafikų palyginimui ir kiekybiniam palyginimui atsižvelgiant į dimensijų mažinimo parametrus. Naudojami

PCA, LLE, MDS, ISOMAP ir t-SNE. VisCoDeR vizualizuoja pradinį duomenų rinkinį, tačiau tuo procesas ir baigiasi [13]. Nėra galimybės tolesnei analizei pasirinkti norimas pradinių duomenų dalis, ir kiekviename žingsnyje taikyti labiausiai tinkantį dimensijų mažinimo metodą.



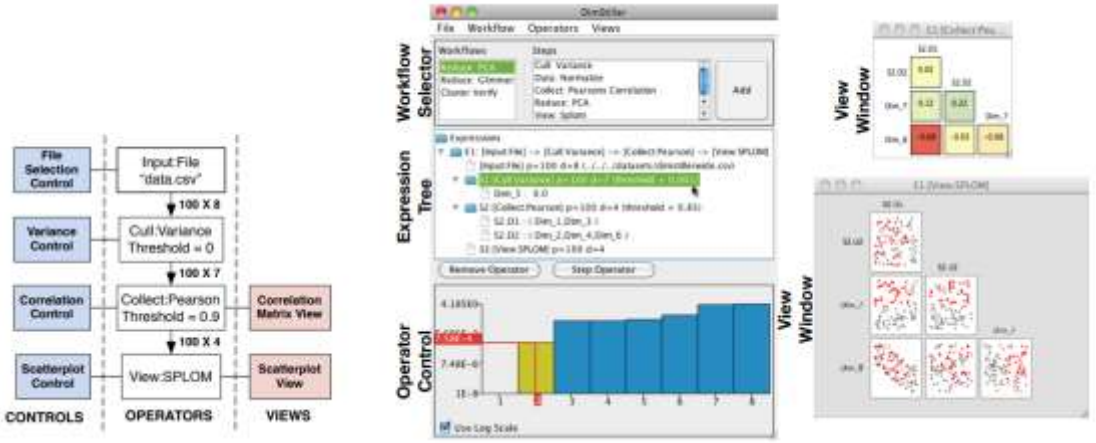
3 pav. VisCoDeR įrankio langas

D. Kammer ir kt. (2018) pristatė interaktyvų didžiųjų duomenų vizualizavimo įrankį, turintį priartinimo funkciją (VANDA projekto dalis). Duomenys atvaizduojami dvimatėje erdvėje panaudojus dimensijų mažinimo metodus, tiesa tik du. Pasiūlytas įrankis leidžia dimensijų mažinimą atlikti keliais skirtingais būdais, o gautus rezultatus palyginti paeiliui arba lygiagrečiai tame pačia ekrane. Duomenų objektai yra pagal panašumą yra sugrupuojami į klasterius, kuriuos detaliai analizuoti galima juos priartinant. Užvedus pelę ant tam tikro objekto yra parodomos jo savybės. Savybių palyginimui yra naudojamos histogramos [50]. Interaktyvios duomenų analizės principus D. Kammer aprašo ir kitose savo publikacijose [52], [51]. Tačiau Kammer įrankyje dimensijų mažinimo metodas yra pritaikomas tik kartą pradiniam duomenų rinkiniui. Vėliau galima vaizdą priartinti, tačiau pasirinktam klasteriui iš naujo pritaikyti kitus dimensijų mažinimo metodus galimybės nėra.

4 pav. VANDA projekto įrankio langas

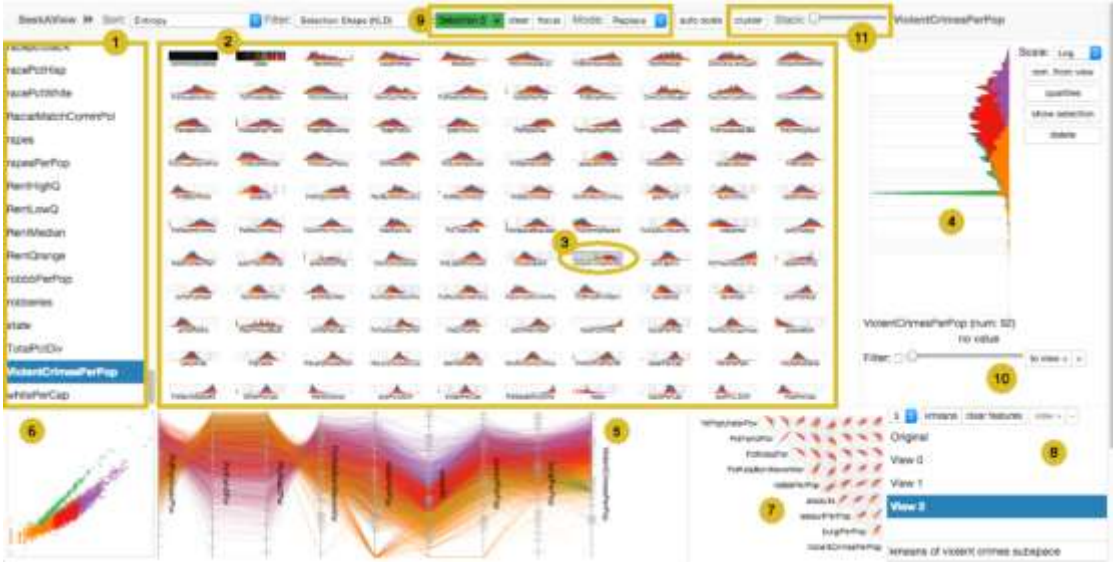
T. Frantz ir Z. Ruan ir kt. ir L. Braun pristatė vizualizavimo įrankius tinklo srauto duomenų analizei ir vizualizavimui [30], [94], [7]. Tačiau šie darbai turi ribotą panaudojimo sritį.

S. Ingram ir kt. (2010) pristatė *DimStiller* įrankį dimensijų mažinimui, kuris duomenų analizės procesą dalina į atskirus etapus. Jame galima keisti parametrus ir matyti, kaip tai įtakoja rezultatus [42]. Tačiau *DimStiller* nėra itin interaktyvus, neleidžia tiesiogiai priartinti vaizdo ar keisti dimensijų mažinimo metodų.



5 pav. DimStiller įrankio langas

J. Krause pristatyta *Seek-a-view* metodika ir įrankis leidžia detaliai analizuoti atskiras dimensijas, nes, autorių teigimu, tipiniai vizualizavimo būdai, kaip pvz. sklaidos diagramos, neleidžia detaliai analizuoti skirtingų dimensijų ryšių. Taip pat egzistuojantys metodai turi ribotas galimybes aptikti ir pasiūlyti potencialiai reikšmingus ir įdomius duomenų sub-rinkinius [57]. Tačiau jų pasiūlytas įrankis nepateikia bendro duomenų rinkinio vizualizavimo. Duomenys yra vizualizuojami pritaikius tik PCA metodą, be galimybės šio grafiko vaizdą priartinti. Patogus įrankis turėtų spręsti abi problemas – tiek aiškiai skirtingais lygiais vizualizuoti analizuojamus duomenis, tiek pateikti duomenų savybių tyrimui reikalingus skaitinius parametrus.



6 pav. *Seek-a-view* įrankio langas

2.4.2 Komercinių įrankių apžvalga

Šiame skyriuje bus pademonstruotos vizualizavimo įrankių ***ZingChart, D3.js, Tableau public, Visualize free, Flare***, ***Microsoft Power BI, Pentaho, Datameer ir IBM Watson***, skirtų didiesiems duomenims vizualizuoti, funkcionalumas.

Duomenų vizualizavimo įrankis ***ZingChart*** – tai *JavaScript* diagramų biblioteka, leidžianti kurti interaktyvias *Flash* arba HTML5 diagramas. Siūloma daugiau nei 100 grafikų tipų. Naudotojas turi siūlomus *JavaScript* scenarijus įkelti į savo kuriamą HTML dokumentą.

ZingChart sistemoje yra pateikta keliolika diagramų bei grafikų tipų, kuriais galima pasinaudoti vizualizuojant norimus duomenis. Galima rinktis iš stulpelinių diagramų, taškinių grafikų, žemėlapių ir kitų dažniausiai naudojamų vizualizavimo metodų (7 pav.). Šiuo įrankiu naudotis paprasta, bet kai duomenų apimtis gana didelė, ne visais metodais pavyksta atvaizduoti norimus duomenis, kadangi šiame įrankyje nėra įgyvendintų dimensijos mažinimo metodų.

7 pav. Vizualizavimo metodai *ZingChart* sistemoje

Grafikų ir diagramų objektai gali būti išreiškiami fiksuotais dydžiais arba procentine išraiška. Kiekviena vizualizacija gali būti modifikuojama keičiant objektų spalvas, grafikų ir diagramų tipus, įtraukiant norimas objektų grupes.

D3.js – tai *JavaScript* biblioteka duomenų vizualizacijoms kurti. *D3.js* padeda vizualizuoti duomenis naudojant HTML, SVG ir CSS.

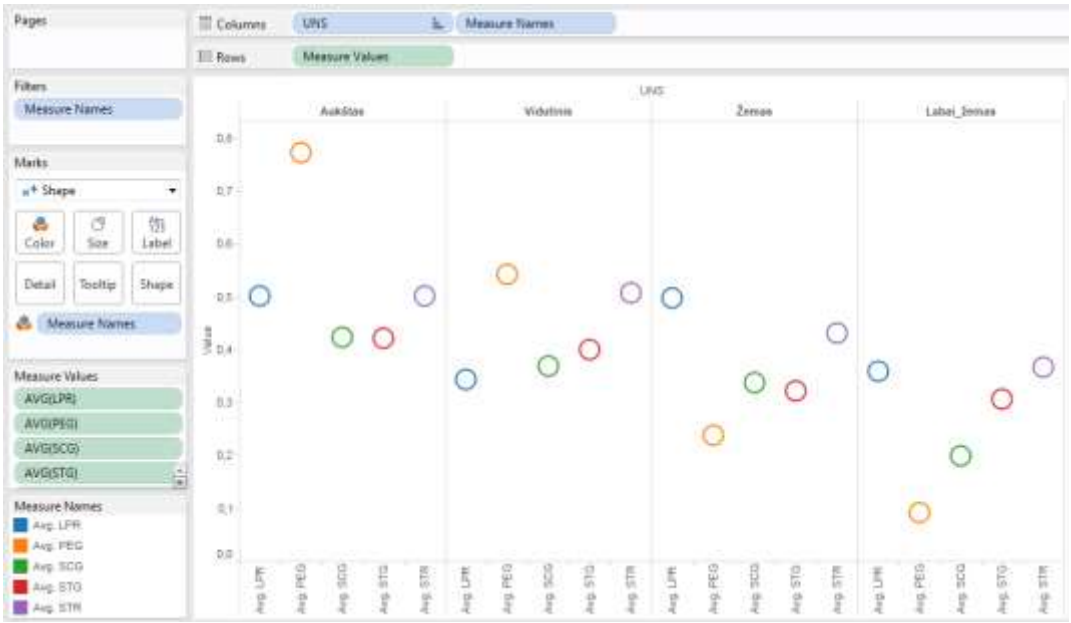
D3.js sistema siūlo gana platų vizualizavimo metodų pasirinkimą (8 pav.).

8 pav. *D3.js* sistemos siūlomi vizualizavimo metodai

Tableau public – tai nemokama Tableu programinės įrangos versija, skirta kurti interaktyvias duomenų vizualizacijas ir jas įkelti į kuriamą internetinę svetainę,

nereikalaujant programavimo įgūdžių. Nors tai į kompiuterį instaliuojama programa, tačiau sukurti grafikai saugomi viešame serveryje. Čia pat yra galimybė naudotojams tarpusavyje dalintis sukurtomis vizualizacijomis.

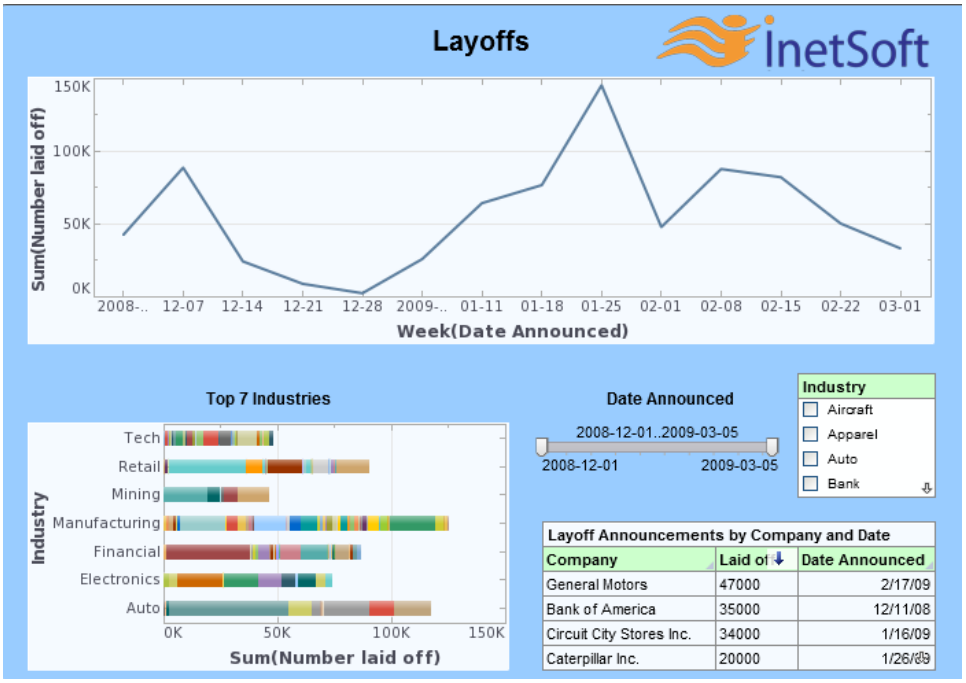
Įdiegiama *Tableau public* programėlė siūlo 24 duomenų vizualizavimo metodus – juostines, stulpelines ir taškines diagramas, žemėlapius, linijinius grafikus ir kt. Vizualizavimo pavyzdžiai pateiktas 9 paveiksle. *Tableau public* įrankiu naudotis gana paprasta, patogi vartotojo sąsaja, atliekami pakeitimai iškart atvaizduojami grafikuose.



9 pav. *Tableau public* programos langas su testinių duomenų vidurkių taškine diagrama

Visualize free įrankis – tai nemokama alternatyva komerciniam vizualizavimo įrankiui *InetSoft*. Tai internetinė svetainė, į kurią galima įkelti norimus vizualizuoti duomenis ir, pasirinkus siūlomą vizualizavimo būdą, gauti duomenų interaktyvią vizualizaciją.

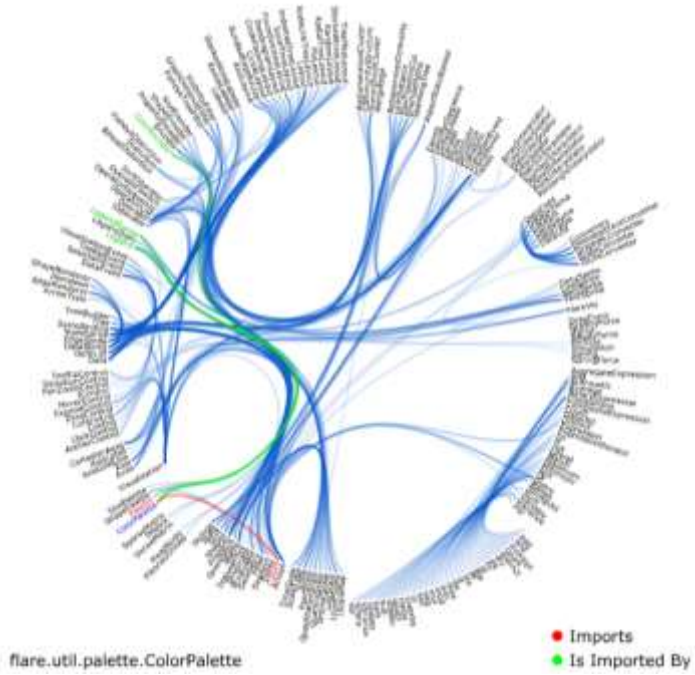
Norint vizualizuoti savo duomenis *Visualize free* įrankio pagalba būtina registracija, po kurios galima įkelti savo duomenis ir juos vizualizuoti arba pasinaudoti siūlomais duomenų rinkiniais, peržiūrėti jau sukurtas vizualizacijas. Įrankiu gauto rezultato pavyzdys pateiktas 10 paveiksle.



10 pav. Duomenų vizualizavimo pavyzdys, gautas *Visualize free* sistema

Flare – tai *ActionScript* biblioteka, skirta kurti duomenų vizualizacijas, veikiančias *Adobe Flash Player* sistemoje. Tai atviro kodo programa, kurią naudoja gerai žinomos organizacijos, tokios kaip *IBM Visual Communication Lab* ir *BBC News*.

Įdiegus *Flare* įrankio biblioteką, galima naudotis visais siūlomais vizualizavimo metodais. 11 paveiksle pateiktas *Flare* svetainėje esantis vizualizavimo pavyzdys - jame pavaizduotos duomenų klasių tarpusavio priklausomybės.

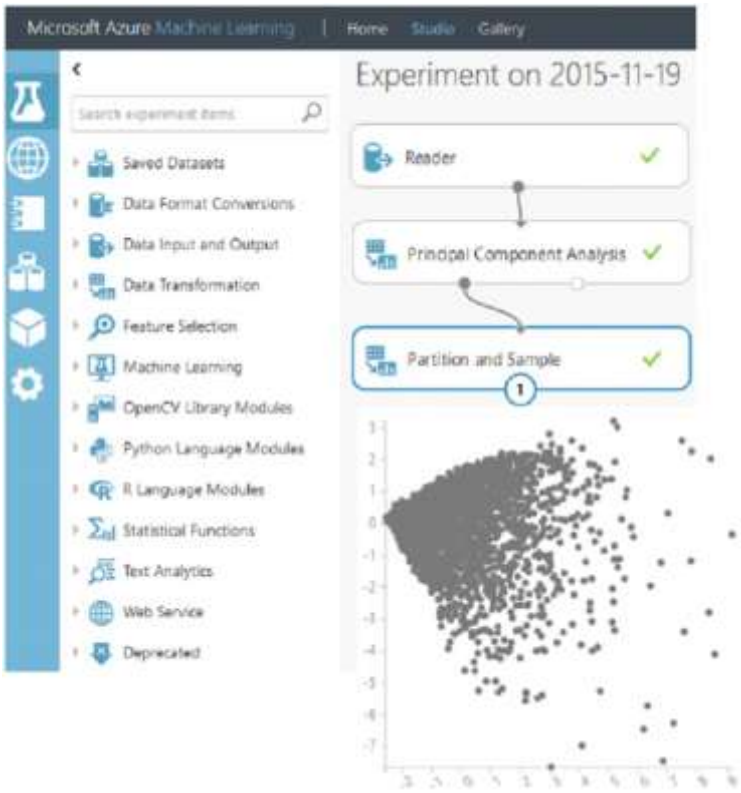


11 pav. Duomenų vizualizavimo pavyzdys (priklausomybių grafas), pateiktas *Flare* sistemoje

Microsoft Azure – daugybę analitinių, duomenų bazių, interneto svetainių ir mobilių programėlių, tinko ir duomenų saugyklų sprendimų debesyje siūlanti platforma.

Didelės apimties duomenų vizualizavimui yra realizuotas pagrindinių komponenčių metodas, kurio pagalba buvo vizualizuotas 5 000 000 eilučių ir 18 stulpelių testinių duomenų rinkinys, aprašantis fizinių procesų signalus (12 pav.).

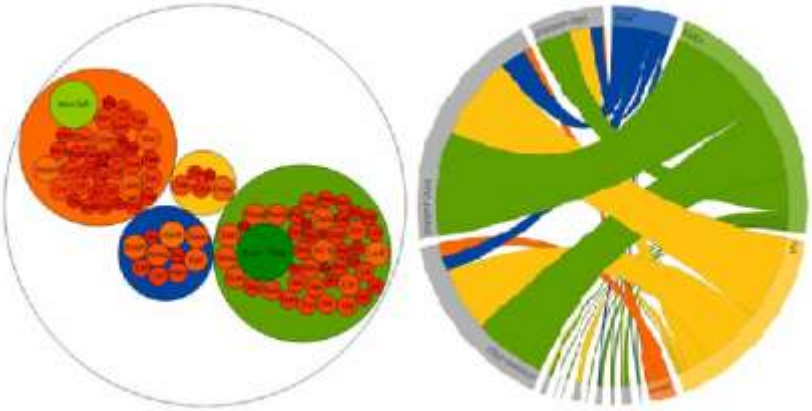
Duomenų analitikai ir vizualizavimui Microsoft siūlo atskirą sprendimą Power BI. Tai galingas įrankis, leidžiantis kurti duomenų analizės skydelius (angl. *Dashboards*): sujungti duomenis iš kelių šaltinių, juos apdoroti ir pateikti įvairiomis formomis.



12 pav. Testinių duomenų vizualizavimas *Microsoft Azure* pagalba

Pentaho yra pirmaujanti duomenų integravimo ir verslo analitikos kompanija, siūlanti atviro kodo platformą didžiųjų duomenų apdorojimui.

Įrankyje realizuotos didžiųjų duomenų užkrovimo, paruošimo, vizualizavimo ir analizės galimybės. Pateikiamas gana platus interaktyvių duomenų vizualizavimo metodų pasirinkimas. Keli testinių duomenų vizualizavimo pavyzdžiai pateikiami 13 paveiksle. Nors šis įrankis ir yra skirtas didžiųjų duomenų analizei, tačiau jokių realizuotų dimensijos mažinimo metodų nėra. Todėl vizuali analizė tampa pakankamai komplikuota, kadangi vaizdai yra perkrauti arba dalis duomenų nėra atvaizduojama.



13 pav. Testinių duomenų vizualizavimas *Pentaho* pagalba

Datameer – didžiųjų duomenų analizės *Hadoop* platforma, skirta duomenų integravimui, analizei ir vizualizavimui. Įrankio pagalba gaunami interaktyvūs vaizdai, galima rinktis iš daugybės siūlomų diagramų ir grafikų tipų. Tačiau nėra dimensijų mažinimo pritaikymo galimybės. Todėl gaunami vaizdai dažniausiai nėra informatyvus.

IBM Watson – įrankis, skirtas didžiųjų duomenų analizei. Siūloma išbandyti testinių duomenų rinkinius, pateikiami analizių pavyzdžiai. Siūlomų arba savo užkrautų duomenų vizualizavimui galima rinktis vieną iš pateikiamų vizualizavimo būdų. Testinių duomenų vizualizavimo pavyzdys pateikiamas 14 paveiksle. *IBM Watson* galima pritaikyti PCA metodą.

14 pav. Testinių duomenų vizualizavimas *IBM Watson* pagalba

Išnagrinėjus didžiųjų duomenų vizualizavimo įrankius ***Microsoft Azure, Pentaho, Datameer, IBM Watson*** bei šių įrankių siūlomus vizualizavimo metodus, galima teigti, kad vieno geriausio ir labiausiai tinkančio duomenų vizualizavimo įrankio

nėra. Kiekvienas įrankis turi savo privalumų (lengva navigacija, didelis vizualizavimo metodų pasirinkimas, interaktyvių vizualizacijų kūrimo galimybės) ir trūkumų (ne visais įrankiai pavyksta atvaizduoti norimus duomenis dėl dimensijos mažinimo metodų stokos, dėl didelių apimčių gaunamas vaizdas labai perkrautas). Renkantis įrankį duomenų vizualizavimui, reikėtų atsižvelgti į vizualizuojamų duomenų specifiką bei siekiamą gauti rezultatą.

2 lentelėje pateikta šiame skyriuje nagrinėtų didžiųjų duomenų vizualizavimo įrankių apžvalga. Lentelėje pateiktos įrankių diegimo galimybės, išorinių duomenų šaltinių naudojimo bei duomenų analizės galimybės. Dauguma apžvelgtų duomenų vizualizavimo įrankių yra interneto svetainės ir atviro kodo programinė įranga.

2 lentelė. Duomenų vizualizavimo įrankių apžvalga

Duomenų vizualizavimo įrankis	**Diegimo galimybės**	**Duomenų šaltiniai**	**Duomenų analizės galimybės**
ZingChart	Interneto svetainė (HTML5)	CSV duomenų formatas	Duomenų vizualizavimas taikant 24 skirtingų metodų
D3.js	Interneto svetainė	CSV, JSON duomenų formatai	Duomenų vizualizavimas
Tableau	Diegimas serveryje; Naudojimas debesyje (mokama, nemokama versijos)	XLS, CSV duomenų formatai	Duomenų vizualizavimas taikant 24 skirtingų metodų , dalijimasis rezultatais
Visualize free	Interneto svetainė	XLS, CSV duomenų formatai	Duomenų vizualizavimas, interaktyvios vizualizacijos, dalijimasis rezultatais
Flare	Atviro kodo programinė įranga	TAB, JSON, XML duomenų formatai	Duomenų vizualizavimas, interaktyvios vizualizacijos, dalijimasis rezultatais
IBM Watson	Naudojimas debesyje	XLS, XLSX, CSV ir kt. duomenų formatai	Duomenų vizualizavimas, interaktyvios vizualizacijos
Microsoft Azure; Power BI	Naudojimas debesyje	XLS, JSON ir kt. duomenų formatai	Duomenų vizualizavimas, interaktyvios vizualizacijos

Datameer	Diegimas serveryje; Naudojimas debesyje	TXT, CSV, JSON, XML, HTML ir kt. duomenų formatai	Duomenų vizualizavimas, interaktyvios vizualizacijos
Pentaho	Atviro kodo programinė įranga	XLS, XML, JDBC ir kt. duomenų formatai	Duomenų vizualizavimas, interaktyvios vizualizacijos

2.5 Skyriaus apibendrinimas

Šiame skyriuje aprašytas didžiųjų duomenų tyrybos procesas bei identifikuota, su kokiais sunkumais susiduriama analizuojant didžiuosius duomenis. Apžvelgti didžiųjų duomenų tyrybos metodai.

Didžiųjų duomenų vizualizavimas yra iššūkis duomenų analitikams ir mokslininkams. Egzistuojantys tradiciniai metodai ir įrankiai dažnai netinka didžiųjų duomenų tyrimams.

Darbe apžvelgti ir išbandyti Microsoft Azure, Pentaho, Datameer, IDM Watson įrankiai. Nors šie įrankiai leidžia užkrauti pakankamai didelės apimties duomenų rinkinius, bet tik Microsoft Azure turi duomenų dimensijos mažinimo metodų pritaikymo galimybę. Kiti minėti įrankiai neturi realizuotų dimensijos mažinimo metodų ir duomenų vizuali analizė tampa komplikuota, kadangi dalis duomenų lieka neatvaizduota.

Darbe taip pat išnagrinėti didžiųjų duomenų vizualizavimo įrankiai *ZingChart*, *D3.js*, *Tableau public*, *Vizualize free* ir *Flare* bei šių įrankių siūlomi vizualizavimo metodai, tokie kaip juostinės, stulpelinės ir taškinės diagramos, žemėlapiai, linijiniai grafikai ir kt. Išnagrinėjus visus minėtus duomenų vizualizavimo įrankius ir jų siūlomus metodus, galima teigti, kad vieno geriausio ir labiausiai tinkančio duomenų vizualizavimo įrankio nėra. Kiekvienas įrankis turi savo privalumų (lengva navigacija, patogi vartotojo sąsaja, didelis vizualizavimo metodų pasirinkimas, interaktyvių vizualizacijų kūrimo galimybės) ir trūkumų (ne visais įrankiais pavyksta atvaizduoti norimus duomenis dėl dimensijos mažinimo metodų stokos, dėl didelių apimčių gaunamas vaizdas labai perkrautas).

Šiame skyriuje taip pat išanalizuoti mokslinėje literatūroje pristatomi VisCoDeR, VANDA, DimStiller ir Seek-a-view įrankiai. Nors jie išsprendžia atskiras su duomenų vizualizavimu susijusias probemas, tačiau kiekvienas turi savų trūkumų. VisCoDeR neturi galimybės tolesnei analizei pasirinkti norimas pradinių duomenų

dalis, ir kiekviename žingsnyje taikyti labiausiai tinkantį dimensijų mažinimo metodą. Kammer įrankyje (VANDA) dimensijų mažinimo metodas yra pritaikomas tik kartą pradiniam duomenų rinkiniui. Vėliau galima vaizdą priartinti, tačiau pasirinktam klasteriui iš naujo pritaikyti kitus dimensijų mažinimo metodus galimybės nėra. DimStiller nėra interaktyvus, neleidžia tiesiogiai priartinti vaizdo ar keisti dimensijų mažinimo metodų. Seek-a-view nepateikia bendro duomenų rinkinio vizualizavimo.

Šioje disertacijoje pristatoma strategija ir įrankis, kurie pašalina minėtus trūkumus. 3 lentelėje pateikiamas jau egzistuojančių ir siūlomo įrankio palyginimas.

3 lentelė. Įrankių palyginimo lentelė

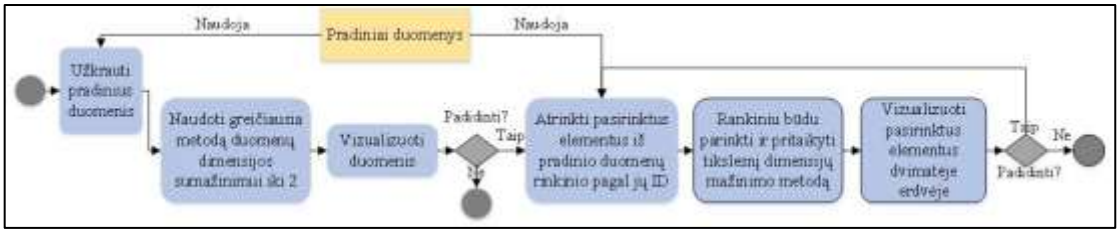
	Prototipas™	Tableau	Pentaho	Datameer	Zing Chart	Azure/Power BI	IBM Watson	VisCoDeR	Vanda	DimStiller	Seek-a-view
Dimensijų mažinimas pasirinktoms duomenų sritims	+	–	–	–	–	–	–	–	–	–	–
Palaikomi dimensijų mažinimo metodai	PCA, MDS, ICA, PC, LLE, Isomap	–	PCA (***)	–	–	PCA	PCA	PCA, LLE, MDS, ISOMAP ir t-SNE	Mds, negative	PCA	PCA
Pasirinktos duomenų erdvės priartinimas	+	+ (*, **)	+ (*, **)	+ (*, **)	+	+ (scale)	+ (maps)	+	+	–	+
Galimybė kiekviename žingsnyje keisti dimensijų mažinimo metodą	+	–	–	–	–	–	–	–	–	–	–
Rezultatų, gautų pritaikius skirtingus metodus, palyginimas	+	–	–	–	–	–	–	+	+	–	–
Rezultatų su visais metodais išankstinė peržiūra (preview)	+	–	–	–	–	–	–	–	–	–	–
Dimensijų mažinimo metodų tikslumo rodikliai	+	–	–	–	–	+	–	–	–	–	–
Pasirinktų duomenų sričių/dimensijų charakteristikų pateikimas	+	+	+	+	+	+	+	+	+ (histograms)	+	+
Integracija su R	+ (***)	+	+	–	–	+	CognizeR. ****	–	–	–	–

* Galima priartinti pasirinktas žemėlapių sritis;
**Yra „drill-down“ (pasirinktų dimensijų detalizavimo) funkcija
*** Sukurta R pagrindu
**** RStudio, integruotas į IBM Watson Studio, suteikia IDE darbui su R

3 Duomenų vizualizavimo metodų tyrimai

3.1 Dimensijų mažinimo metodų greičio ir tikslumo įvertinimas

Disertacijoje siūloma duomenų vizualizavimo strategija yra daugiapakopė – kiekviename žingsnyje gali būti parenkamas tam tikras dimensijų mažinimo metodas, atsižvelgiant į duomenų tipą ir kiekį. Pradiniuose etapuose, kai duomenų yra labai daug, greičio faktorius gali būti svarbesnis nei tikslumo. Tolesniuose žingsniuose, kai atfiltruojama tik dalis duomenų, tikslumo svarba išauga.



15 pav. Daugiapakopė didžiųjų duomenų vizualizacijos strategija

Mokslinėje literatūroje daugiausiai aptinkami kiekybiniai metodų palyginimai [28], [93], [102]. Kai kuriuose straipsniuose [70], [56] galima rasti ir pasirinktų metodų greičio arba tikslumo įvertinimų. Tyrimų rezultatai leidžia daryti prielaidą, jog vieni metodai yra greitesni, bet mažiau tikslūs, o kiti – atvirkščiai. Taip pat pastebima, kad skirtingi metodai geriau apdoroja skirtingo tipo duomenis.

Trūksta bendro skirtingų metodų įvertinimo, todėl šiame skyriuje analizuojami dimensijų mažinimo metodų greitis ir tikslumas skirtingomis sąlygomis. Analizei pasirinkti šie populiarūs metodai: Daugiamačių skalių (MDS), Pagrindinių komponenčių analizės (PCA), Nepriklausomų komponenčių analizės (ICA), Pagrindinių kreivių (PC), Lokaliai tiesioginio įterpimo (LLE) ir Isomap.

3.1.1 Tyrimo metodologija

Pagrindinis tyrimo tikslas buvo įvertinti pasirinktų metodų greitį ir tikslumą mažinant pradinių dimensijų skaičių. Kaip tyrimo instrumentas buvo pasirinkta R platforma, turinti daug atviro kodo paketų, realizuojančių skirtingus dimensijų mažinimo metodus. Tyrimo vykdymui panaudota RStudio programinė įranga.

3.1.2 Testavimo duomenų aprašymas

Tyrimui panaudoti skirtingų rūšių duomenys. Duomenų rinkiniai taip pat skiriasi savo dydžiais (tiek objektų, tiek dimensijų skaičiumi). Vykdant tyrimą visais atvejais pradinis dimensijų skaičius yra mažinamas iki dviejų.

Atsitiktinai sugeneruoti neklasterizuoti duomenys

Visų pirma panaudojant R funkciją *sample()* buvo sukurta 50 duomenų rinkinių su atsitiktinai sugeneruotais skaičiais. Stulpelių skaičius yra nuo 10 iki 50. Objektų skaičius svyruoja nuo 1000 iki 10000. Mažiausias duomenų rinkinys yra 1000x10, o didžiausias - 10000x50.

Atsitiktinai sugeneruoti klasterizuoti duomenys

Antroje grupėje yra atsitiktinai sugeneruotų klasterizuotų duomenų rinkiniai – viso 25.

R funkcija *genRandomClust* iš paketo ‘clusterGeneration’ buvo panaudota skirtingo išsibarstymo laipsnio klasterių generavimui [86]. Kiekvienas duomenų rinkinys turi 4 klasterius. Dimensijų kiekis yra nuo 10 iki 50. Objektų kiekis yra nuo 1000 iki 9000. Mažiausias duomenų rinkinys yra 1000x10, o didžiausias - 9000x50.

Realūs finansiniai duomenys

Trečioje grupėje yra 20 duomenų rinkinių. Juos sudaro realūs finansiniai duomenys iš finviz.com [103]. Iš viso yra 7000 įmonių akcijų (duomenų objektų). Kiekvieną įmonę aprašo 50 parametrų (dimensijų). Visi parametrai gali būti suskirstyti į 6 kategorijas: apžvalginiai (kapitalizacija, kaina, apyvarta t.t.), įvertinimo (P/E, PEG, P/B, EPS ir t.t.), finansiniai (ROA, ROE, ROI ir t.t), našumo (kainos pokytis, kintamumas, rekomendacijos), techniniai (ATR, Beta, SMA ir t.t.), priklausomybės (vidinių akcininkų dalis, institucinių investuotojų dalis ir t.t).

Dimensijų kiekis rinkiniuose svyruoja nuo 10 iki 50. Objektų (įmonių) kiekis yra nuo 1000 iki 7000. Mažiausias duomenų rinkinys yra 1000x10, o didžiausias - 7000x50.

3.1.3 Metodų vertinimo kriterijai

Metodams įvertinti panaudoti 2 pagrindiniai kriterijai:

- **Greitis.** Skaičiuotas dimensijų mažinimo algoritmo vykdymo laikas.

- **Tikslumas.** Jam įvertinti panaudoti 3 matai:
    - **Stress**. Šis rodiklis parodo santykinį skirtumą tarp atstumų (tarp tų pačių objektų) skirtingo dimensijų skaičiaus erdvėse. Jis gaunamas sprendžiant mažiausiųjų kvadratų įtempimo funkciją. Kuo ši reikšmė artimesnė 0, tuo dimensijų mažinimo metodas pritaikytas tiksliau. Dydis gautas apskaičiavus MDS metodo mažiausiųjų kvadratų įtempimo funkcija. Tam panaudota R funkcija *mds()* iš paketo 'smacof'.
    - **Spirmeno koeficientas** (angl. *The Spearman's Rank Correlation Coefficient*). Tai statistinis matas, naudojamas įvertinti ryšio stiprumą tarp dviejų duomenų rinkinių [37]. Šis matas naudoja kintamųjų rangus vietoje jų reikšmių. Galimos reikšmės svyruoja nuo -1 (stiprus neigiamas ryšys) iki 1 (stiprus teigiam sąryšis). Jeigu koeficiento reikšmė lygi 0, tai reiškia, jog nėra jokio statistinio ryšio tarp duomenų rinkinių. R funkcija *cor()* su metodu "spearman" buvo panaudota šio mato apskaičiavimui.
    - **Šenono entropija** (angl. *Shannon Entropy*). Entropijos gavimui panaudota R funkcija *entropy* iš paketo 'entropy', kuri atsitiktiniam kintamajam Y apskaičiuoja Šenono entropiją H iš atitinkamų stebimų reikšmių [86], [38]. Šis matas parodo, kaip gerai naujai gauta duomenų projekcija išlaiko pradinį informacijos kiekį. Mažesnė mato reikšmė reiškia didesnį tikslumą.

3.1.4 Tyrimo rezultatai

Šioje dalyje pristatomi greičio ir tikslumo tyrimo rezultatai. Pradžioje atskirai pateikiami kiekvienos duomenų grupės rezultatai, o pabaigoje – apibendrintas įvertinimas.

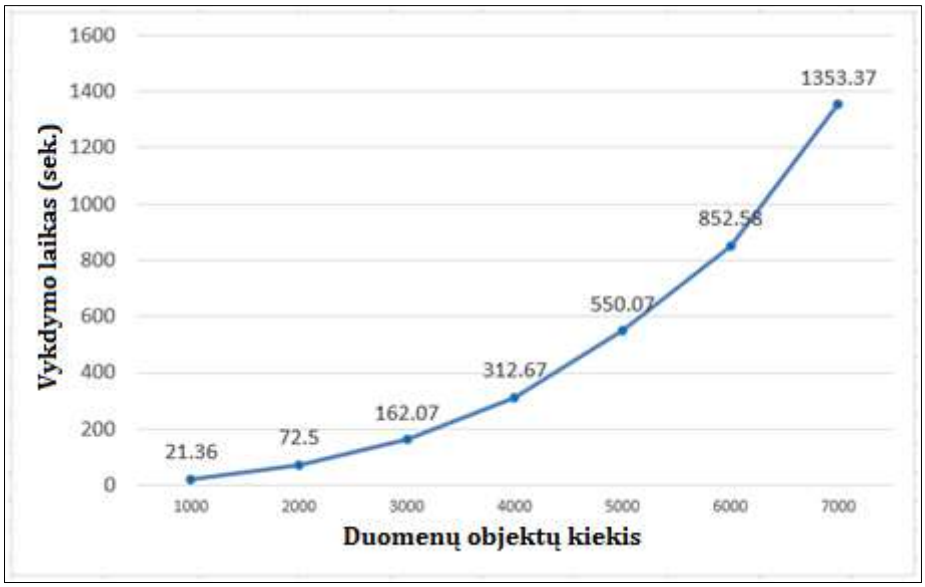
3.1.4.1 Atsitiktinai sugeneruotų neklasterizuotų duomenų atvejis

Metodų vykdymo greitis

Rezultatai rodo, kad MDS (smacof), Isomap ir LLE metodai pasižymi tomis pačiomis savybėmis:

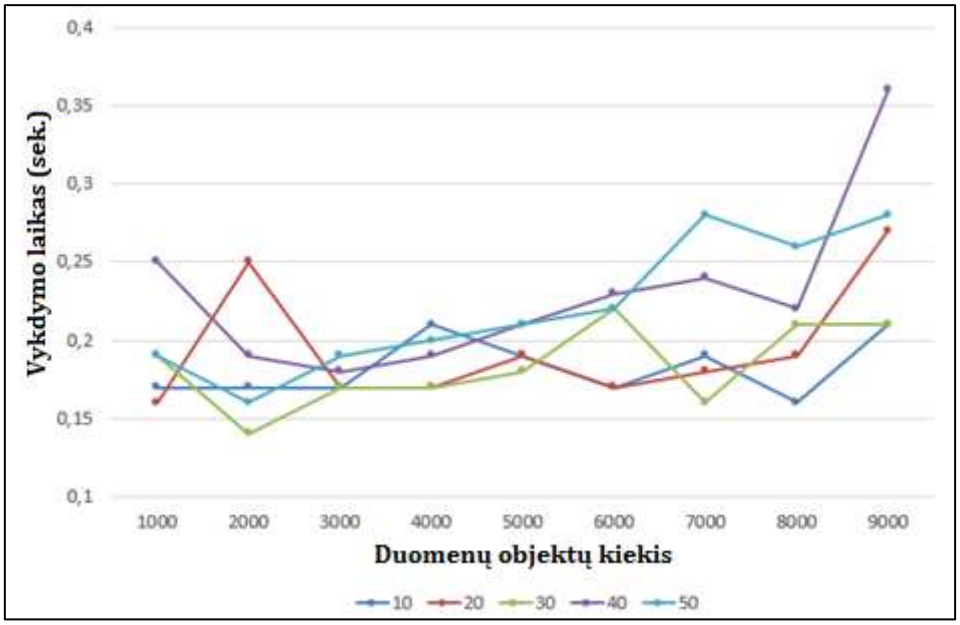
- Didėjant objektų kiekiui, vykdymo laikas ilgėja
- Pradinis dimensijų kiekis neturi reikšmingos įtakos vykdymo laikui

16 paveiksle pavaizduoti vykdymo laikai apdorojant duomenų rinkinius, turinčius 10 dimensijų, bet skirtingą objektų skaičių. Vykdymo laikų grafikai esant didesniam dimensijų skaičiui atrodo beveik vienodai, nes šis faktorius neturi reikšmingos įtakos greičiui. Tiesa, Isomap metodas pasirodė žymiai lėtesnis (pav. 20).



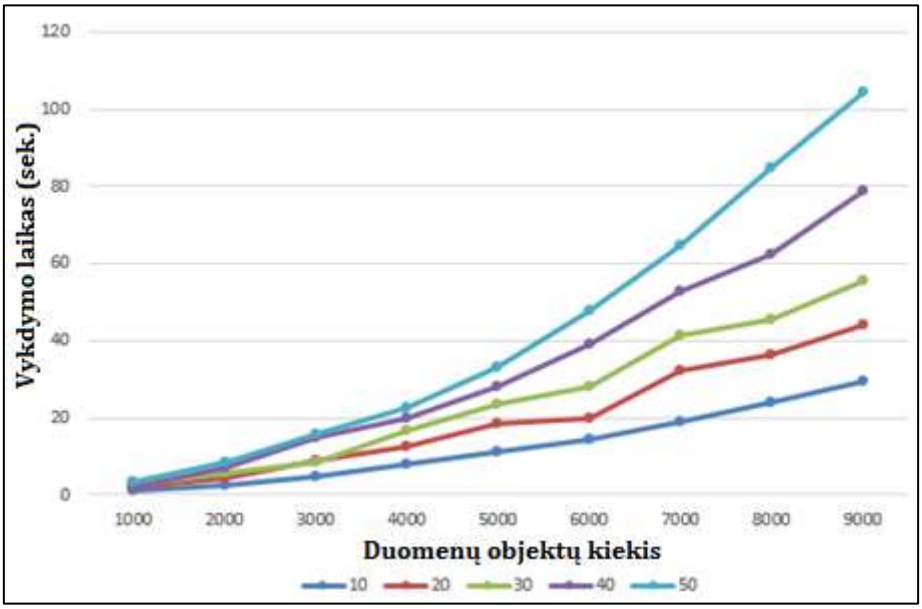
16 pav. MDS (smacof) metodo vykdymo laikas

PCA metodo atveju vykdymo laikas šiek tiek išauga abiem atvejais: tiek didėjant objektų skaičiui, tiek didėjant pradinių dimensijų skaičiui (17 pav.):



17 pav. PCA metodo vykdymo laikas

ICA vykdymo laikas yra panašus į PCA. Tik pagrindinių kreivių metodas išsiskiria pastoviu vykdymo laiko ilgėjimu tiek didėjant objektų skaičiui, tiek didėjant pradinių dimensijų skaičiui (18 pav.).



18 pav. Pagrindinių kreivių metodo vykdymo laikas

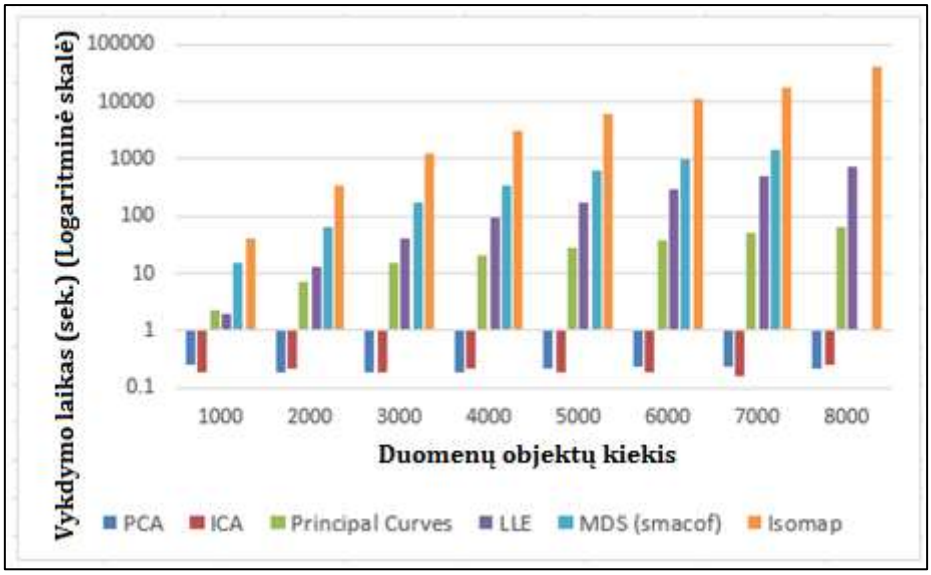
19 paveiksle palyginti visų metodų vykdymo laikai objektų skaičiui svyruojant nuo 1000 iki 10000. Pradinis dimensijų skaičius neturi žymios įtakos greičiui, todėl pavaizduotas tik vienas atvejis – esant 40 pradinių dimensijų.

	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
PCA	0,25	0,19	0,18	0,19	0,21	0,23	0,24	0,22	0,36	0,28
ICA	0,18	0,22	0,19	0,22	0,18	0,18	0,16	0,25	0,29	0,23
PC	2,21	6,89	14,73	19,98	28,29	39,22	52,55	62,47	78,98	105,88
LLE	2,00	13,22	41,51	93,66	178,67	304,12	486,33	727,57	1029,79	-
MDS (smacof)	15,76	65,64	173,64	340,48	624,07	981,79	1482,42	-	-	-
Isomap	39,75	350,41	1229,72	3168,00	6222,66	11078,3	17710,5	41077,2	-	-

19 pav. Metodų vykdymo laikų palyginimas

Atkreiptinas dėmesys, kad LLE negalėjo apdoroti daugiau nei 9000 objektų, Isomap – daugiau 8000, o MDS (smacof) viršutinė riba buvo 7000 objektų. Tai lėmė operatyviosios atminties trūkumas (naudotas kompiuteris, turintis Intel Core i5-2450M 2.5 GHz procesorių ir 4 GB RAM).

20 paveiksle vykdymo laikai pateikti logaritminėje skalėje. PCA ir ICA parodė geriausius rezultatus. Pagrindinės kreivės, MDS ir LLE buvo ženkliai lėtesni. Tačiau lėčiausiai buvo įvykdytas Isomap algoritmas – jo vykdymo laikas smarkiai atsiliko nuo kitų metodų.



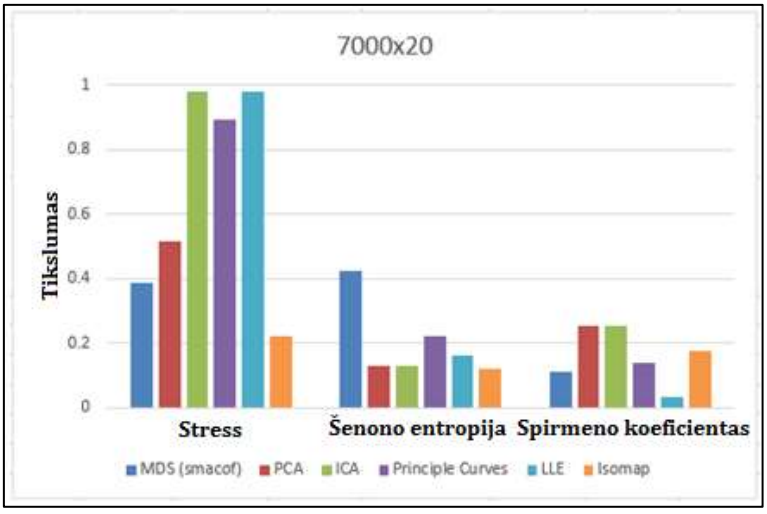
20 pav. Metodų vykdymo laikų palyginimas (logaritminė skalė)

Metodų tikslumas

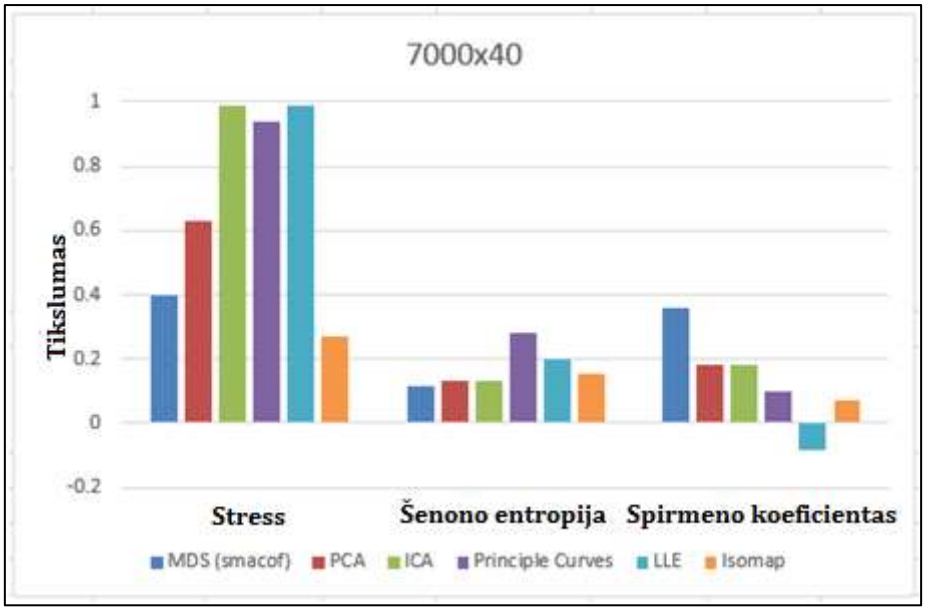
Rezultatai parodė, kad visiems tirtiems metodams galioja tos pačios taisyklės:

- Didesnis objektų skaičius neturi įtakos tikslumui
- Didėjant pradinių dimensijų kiekiui, tikslumas mažėja

Šias taisykles patvirtino visi 3 tikslumo matai. Tiesa, tikslumo sumažėjimas yra nevienodas skirtingiems metodams. 21 ir 22 paveiksluose palygintas visų metodų tikslumas. Kadangi objektų kiekis neturi reikšmingos įtakos, pavaizduotas tik vienas atvejis – dimensijų mažinimas esant 7000 objektų. Esant skirtingam pradiniam dimensijų kiekiui metodai pagal tikslumą išsirikiuoja labai panašiai. Todėl 21 ir 22 paveiksluose pavaizduoti tik du atvejai – tikslumo palyginimas esant 20 ir 40 pradinių dimensijų.



21 pav. Metodų tikslumo palyginimas

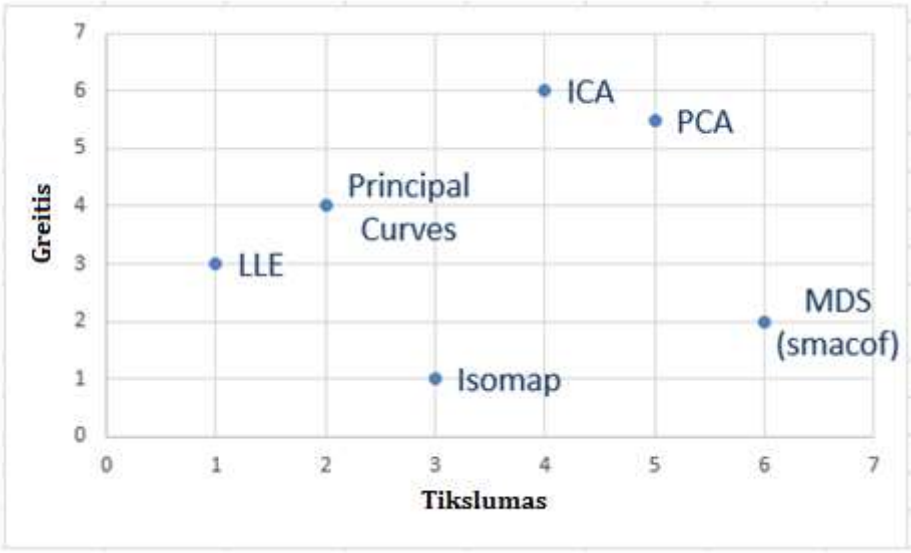


22 pav. Metodų tikslumo palyginimas

Rezultatai rodo, kad apdorojant šiuos testinius duomenų rinkinius tiksliausi buvo PCA ir MDS metodai. LLE tikslumas buvo mažiausias.

Metodų palyginimas

23 paveiksle pateikiami apibendrinti įvertinimai. Visi metodai buvo sureitinguoti pagal greitį ir tikslumą (“6” reiškia didžiausią greitį/tikslumą, “1” – mažiausią).



23 pav. Metodų greičio ir tikslumo palyginimas

PCA ir ICA metodai yra greičiausi. MDS buvo tiksliausias, bet ne toks greitas. Pagrindinės kreivės parodė vidutinius rezultatus. LLE ir Isomap rezultatai buvo

prasčiausi. Nors Isomap yra ženkliai lėtesnis, tačiau tam tikrais atvejais jo tikslumas yra geresnis nei kitų metodų.

3.1.4.2 Atsitiktinai sugeneruotų klasterizuotų duomenų atvejis

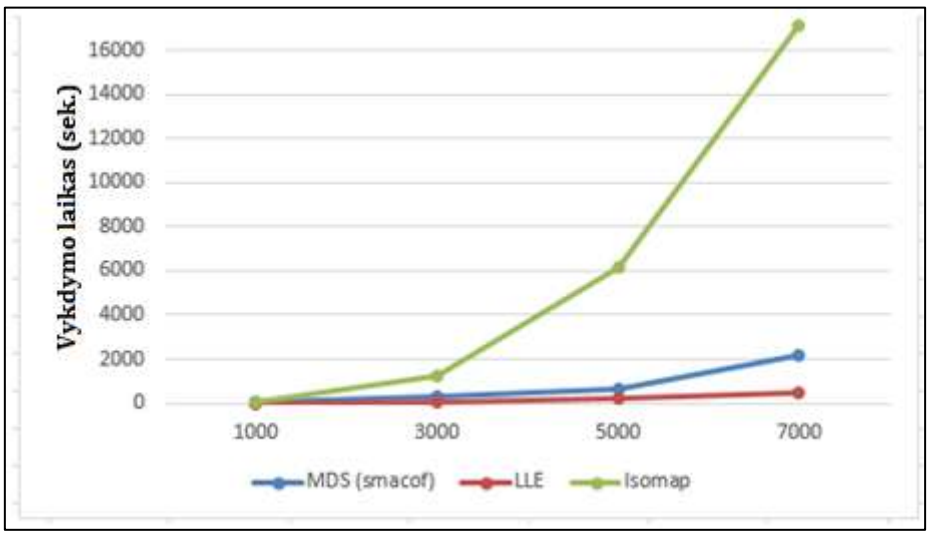
Antru atveju tirtas metodų greitis ir tikslumas apdorojant atsitiktinai sugeneruotus klasterizuotus duomenis.

Metodų vykdymo greitis

MDS (smacof), Isomap ir LLE pastebimos tos pačios taisyklės kaip ir su neklasterizuotais duomenimis:

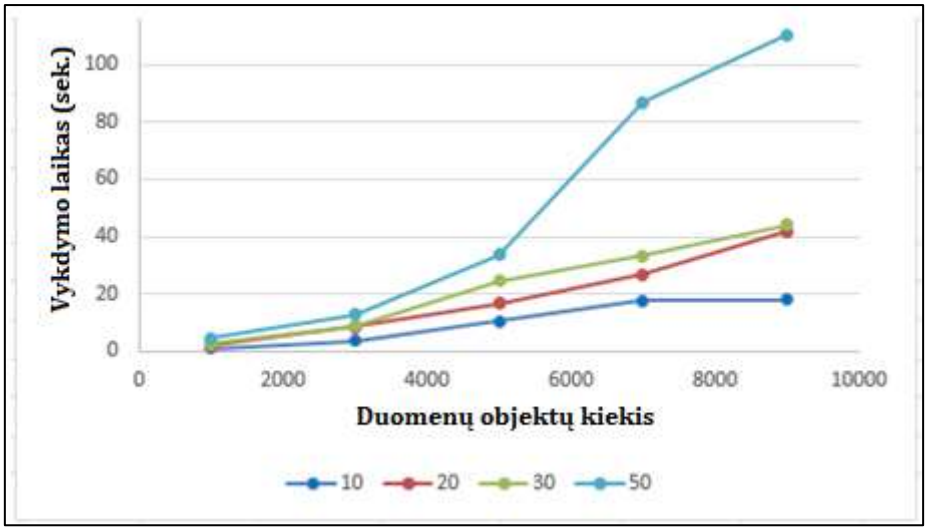
- Didėjant objektų kiekiui, vykdymo laikas ilgėja
- Pradinis dimensijų kiekis neturi reikšmingos įtakos vykdymo laikui

24 paveiksle pavaizduoti šių metodų algoritmų vykdymo laikai apdorojant duomenų rinkiniams nuo 1000x10 iki 7000x10.



24 pav. MDS (smacof), Isomap ir LLE metodų algoritmų vykdymo laikas

PCA atveju vykdymo laikas šiek tiek pailgėja didėjant tiek objektų, tiek dimensijų kiekiui (su tam tikromis išimtimis). ICA vykdymo laikas taip pat panašus į PCA. Klasterizuotų duomenų atveju nėra tokio ryškaus vykdymo laiko didėjimo duomenis apdorojant pagrindinių kreivių metodu (25 pav.), koks buvo matomas neklasterizuotų duomenų atveju (18 pav.)



25 pav. Pagrindinių kreivių metodo vykdymo laikas

26 paveiksle pateiktas visų metodų algoritmų vykdymo laikų palyginimas. Objektų kiekis duomenų rinkiniuose: 1000, 3000, 5000, 7000 ir 9000. Pradinis dimensijų kiekis neturi reikšmingos įtakos, todėl 26 paveiksle pateikiamas vienas atvejis su 40 pradinių dimensijų.

	1000	3000	5000	7000	9000
PCA	0,21	0,2	0,21	0,25	0,23
ICA	0,19	0,17	0,19	0,19	0,17
PC	2,62	11,91	22,81	49,08	109,98
LLE	2,14	41,45	184,04	484,42	
MDS (smacof)	14,81	185,36	42,13	1750,67	
Isomap	38,33	1219,35	6297,34	18008,40	

26 pav. Metodų vykdymo laikų palyginimas

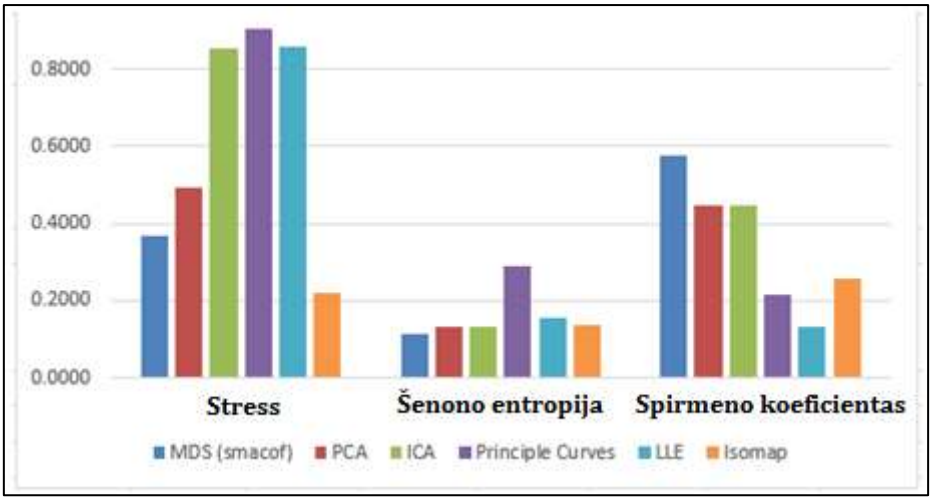
Rezultatai yra panašūs į gautus apdorojant neklasterizuotus duomenis (pav. 20). Galima atkreipti dėmesį į faktą, kad šiuo atveju LLE, MDS (smacof) ir Isomap metodais nepavyko apdoroti duomenų rinkinių turinčių daugiau nei 9000 objektų.

Metodų tikslumas

Apdorojant klasterizuotus duomenis pasitvirtino tos pačios taisyklės:

- Didesnis objektų skaičius neturi įtakos tikslumui
- Didėjant pradinių dimensijų kiekiui, tikslumas mažėja

27 paveiksle pateikti visų metodų tikslumo rodikliai gauti apdorojant rinkinį, turintį 7000 objektų ir 40 dimensijų. MDS (smacof) yra tiksliausias pagal 2 rodiklius: Šenono entropiją ir Spirmeno koeficientą. Tačiau pagal Stress reikšmę, Isomap yra tikslesnis už MDS (smacof). PCA ir ICA parodė vidutinius rezultatus. LLE ir Pagrindinių kreivių tikslumas buvo mažiausias.



27 pav. Metodų tikslumo rodiklių palyginimas

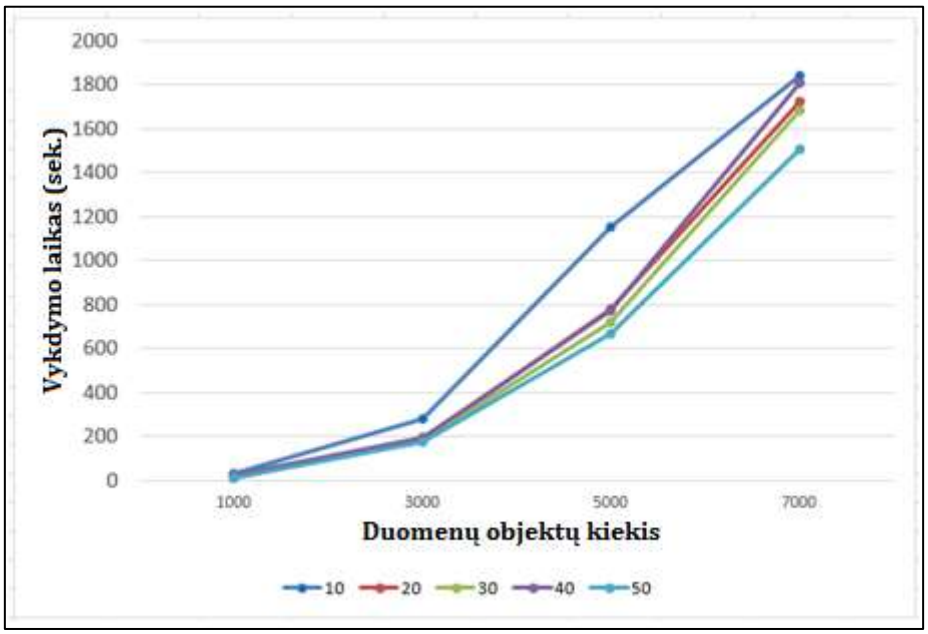
Apibendrintai, greičio ir tikslumo rezultatai apdorojant neklasterizuotus ir klasterizuotus duomenis yra labai panašūs.

3.1.4.3 Realių finansinių duomenų atvejis

Trečiuoju atveju metodai buvo tirti apdorojant realius finansinius duomenis.

Metodų vykdymo greitis

Prieš tai nagrinėtais atvejais, galiojo taisyklės, jog didėjant objektų kiekiui, vykdymo laikas ilgėja, o pradinis dimensijų kiekis neturi reikšmingos įtakos vykdymo laikui. Realių duomenų atveju juos apdorojant MDS (smacof) metodu galima pastebėti, jog pradinis dimensijų kiekis turi nedidelės įtakos vykdymo laikui (28 pav.).



28 pav. MDS (smacof) metodo vykdymo laikas

Vertinant kitų metodų vykdymo laikus galima pastebėti, kad išlieka tos pačios tendencijos kaip ir atsitiktinai generuotų duomenų atveju (29 pav.). Tiesa, realių duomenų apdoroti naudojant LLE metodą nepavyko. Šiam metodui duomenys buvo per daug koreliuoti.

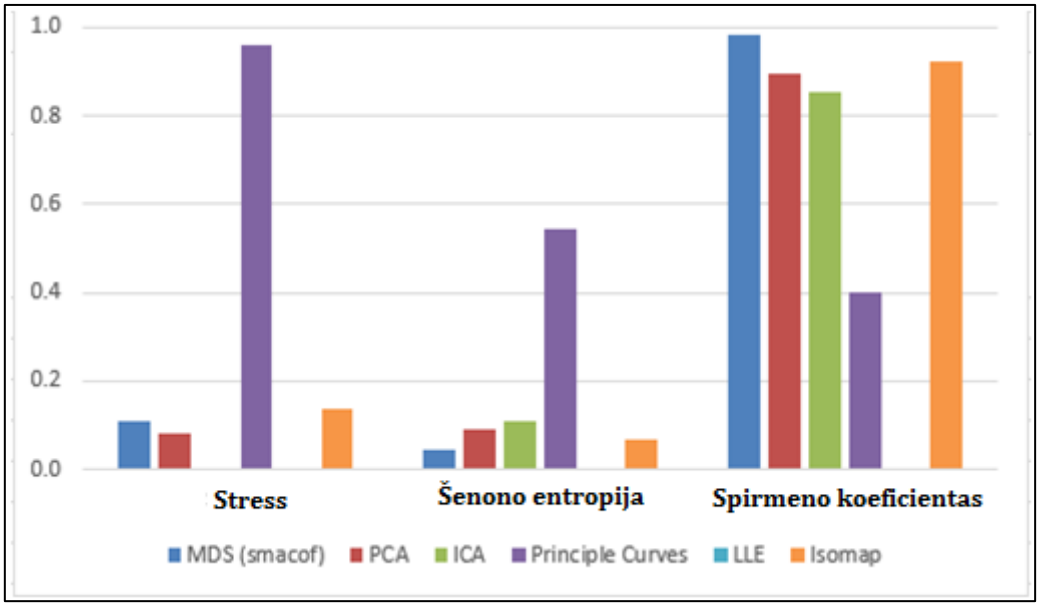
	1000	3000	5000	7000
PCA	0,18	0,15	0,19	0,23
ICA	0,19	0,17	0,16	0,20
PC	4,43	17,27	38,89	81,25
MDS (smacof)	22,58	198,82	774,70	1809,37
Isomap	42,37	1130,76	6175,44	16599,50
LLE	-	-	-	-

29 pav. Metodų vykdymo laikų palyginimas

Metodų tikslumas

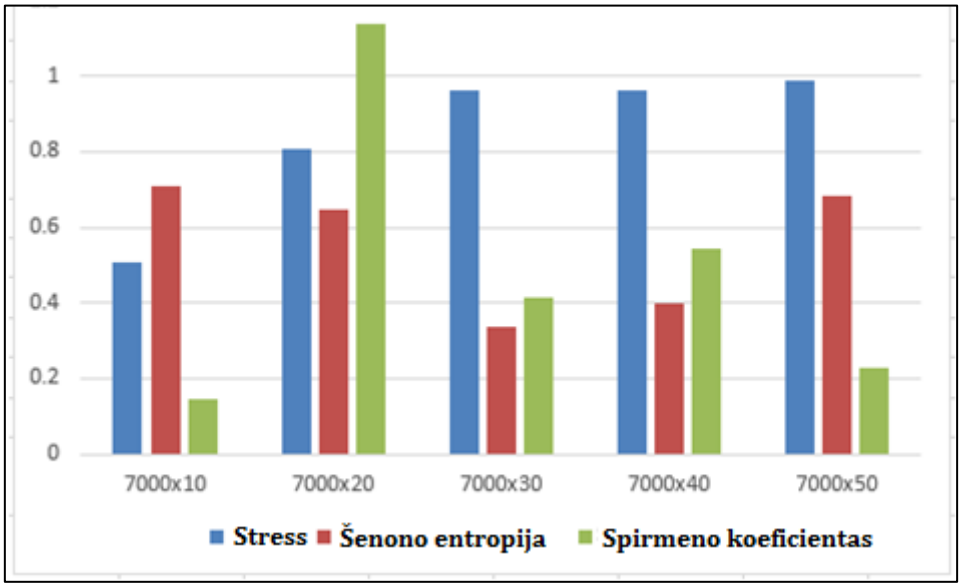
Analizuojant realius duomenis, jų nepavyko apdoroti LLE metodu. Taip pat nepavyko apskaičiuoti Stress reikšmės ICA metodui. Tai parodo, jog visi metodai gali nesunkiai apdoroti atsitiktinai sugeneruotus duomenis, tačiau realių duomenų apdorojimas yra sudėtingesnis uždavinys, su kuriuo kartais susidoroja nevisi metodai.

30 paveiksle pavaizduoti tikslumo rodikliai apdorojant duomenų rinkinį, turintį 7000 objektų ir 40 dimensijų. MDS (smacof), PCA, ICA ir Isomap tikslumas yra panašus su visais duomenų rinkiniais (jis priklauso nuo pradinio dimensijų kiekio, tačiau tendencijos išlieka tos pačios).

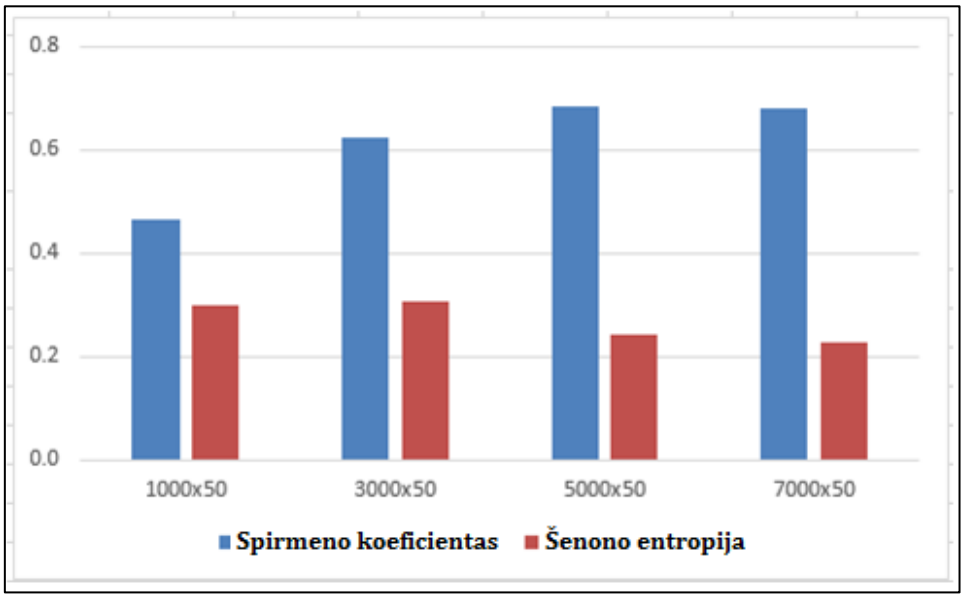


30 pav. Metodų tikslumo rodiklių palyginimas

Tačiau pagrindinių kreivių metodo atveju nepavyko nustatyti bendrų taisyklių, kurios galiotų su visų dydžių duomenų rinkiniais. 31, 32 paveiksluose matoma, kad didėjant pradinių dimensijų kiekiui, Spirmeno koeficiento ir Šenono entropijos reikšmės svyruoja. Tai leidžia daryti prielaidą, kad duomenyse esanti informacija (jos pobūdis) turi įtakos dimensijų mažinimo tikslumui. Todėl yra skirtumas tarp pridėjimo papildomų atsitiktinai sugeneruotų duomenų ir realių duomenų: pastarieji gali įnešti įvairesnių aspektų analizuojamam subjektui.



31 pav. Pagrindinių kreivių metodo tikslumas

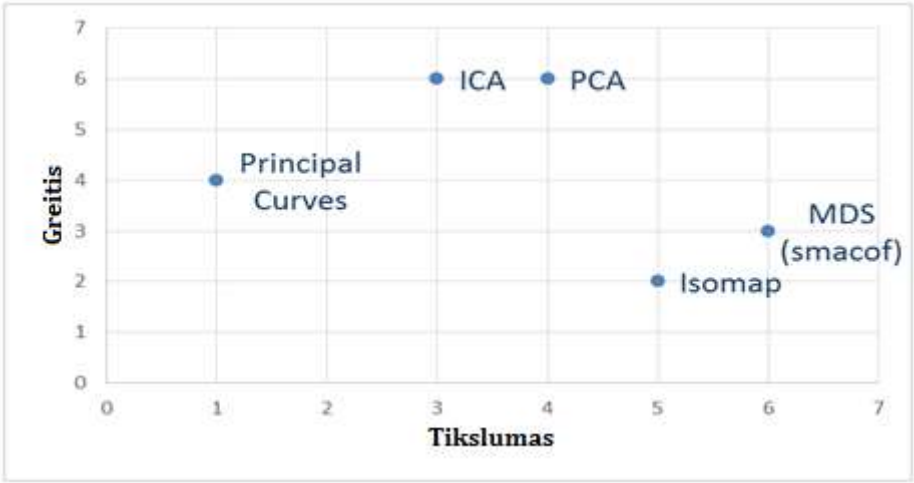


32 pav. Pagrindinių kreivių metodo tikslumas

31, 32 paveiksluose matoma, kad didesnis objektų skaičius lemia didesnį tikslumą. Tai irgi pastebima tik apdorojant realius duomenis.

Metodų palyginimas

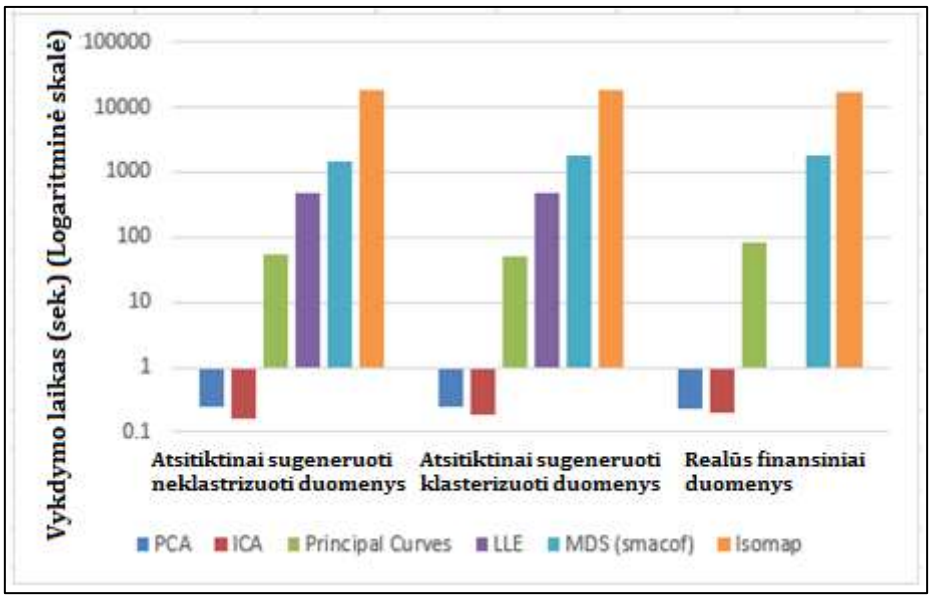
33 paveiksle sureitinguoti visi metodai pagal jų greitį ir tikslumą apdorojant realius duomenis. MDS tikslumas buvo didžiausias. Tačiau jis ne toks greitas kaip PCA ar ICA. Pastarieji metodai buvo greičiausi, tačiau ne tokie tikslūs. ICA greitis yra toks pats kaip PCA, tačiau ICA yra mažiau tikslus.



33 pav. Metodų greičio ir tikslumo palyginimas

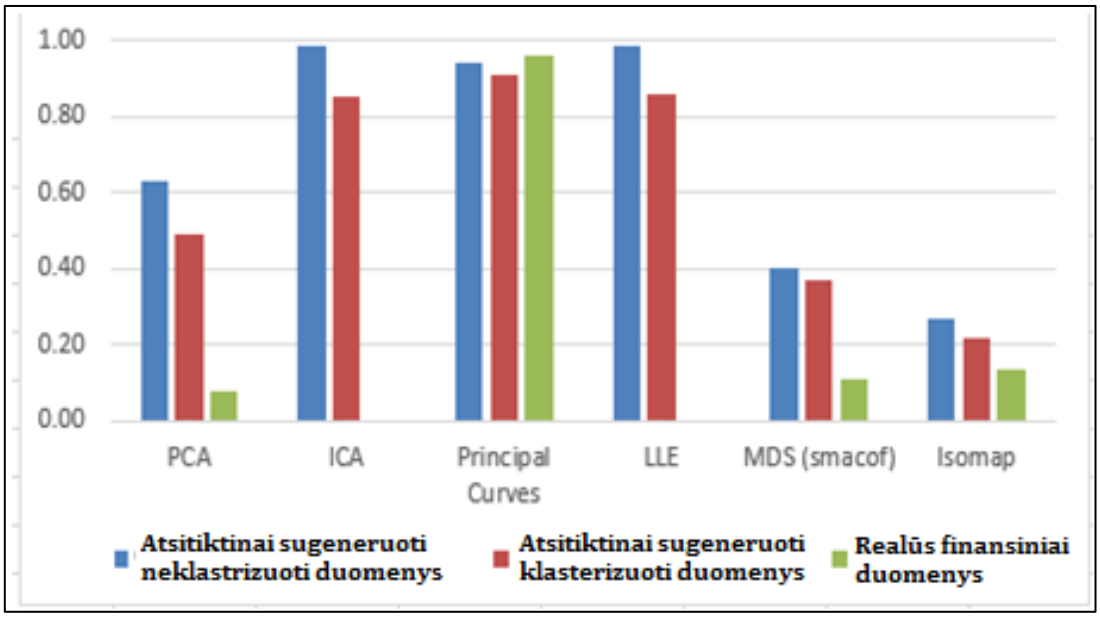
3.1.5 Bendras metodų įvertinimas

Šioje dalyje palyginamas visų metodų greitis ir tikslumas priklausomai nuo apdorojamų duomenų tipo. 34 paveiksle matoma, kad duomenų tipas neturi įtakos metodų vykdymo greičiui.



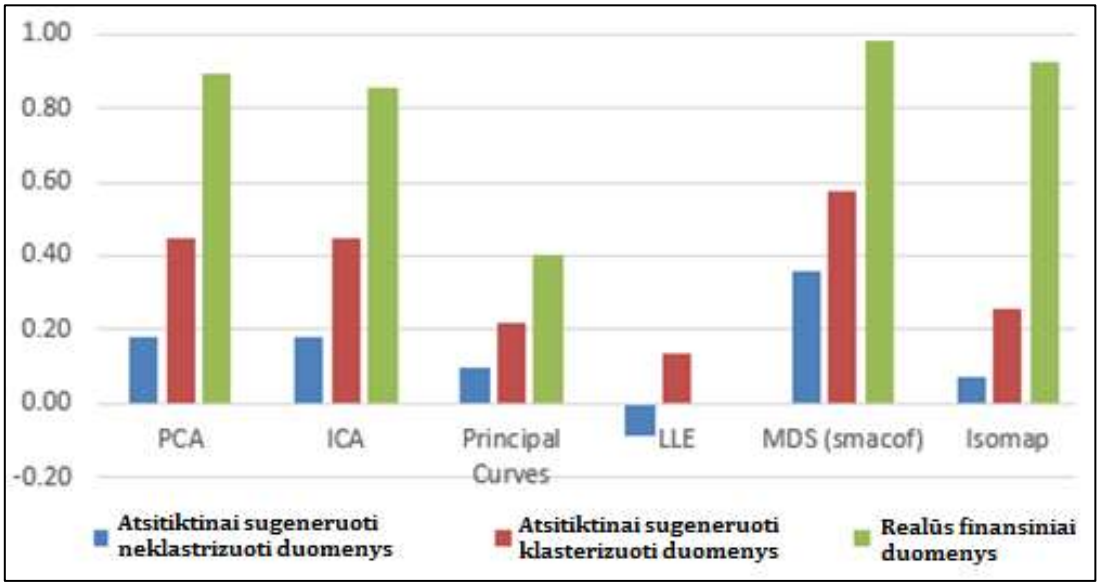
34 pav. Metodų vykdymo laikų palyginimas

Tačiau jis turi įtakos dimensijų mažinimo tikslumui. Klaterizuotų duomenų atveju Stress reikšmės yra mažesnės nei neklasterizuotų duomenų atveju. Apdorojant duomenis PCA, MDS (smacof) ir Isomap metodais geriausias tikslumas buvo pasiektas būtent realių finansinių duomenų atveju (35 pav.).

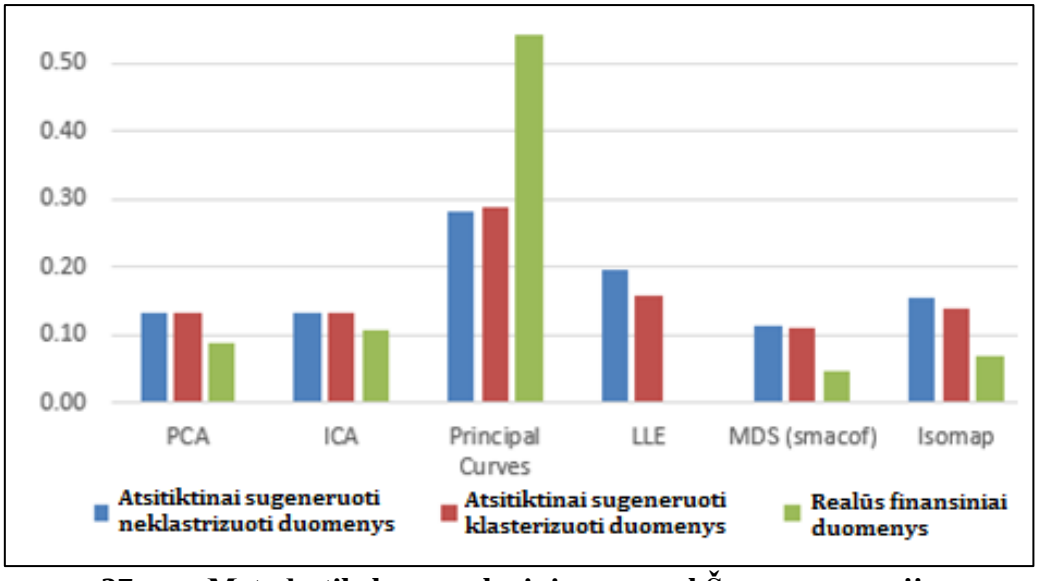


35 pav. Metodų Stress reikšmių palyginimas

Pagal Spirmeno koeficientą (36 pav.) geriausi tikslumo rezultatai pasiekiami apdorojant realius duomenis. Apdorojant klasterizuotus duomenis tikslumas irgi didesnis nei apdorojant neklsterizuotus duomenis.



36 pav. Metodų Spirmeno koeficientų reikšmių palyginimas



37 pav. Metodų tikslumo palyginimas pagal Šenono entropiją

Pagal Šenono entropiją (37 pav.) nėra reikšmingo tikslumo skirtumo apdorojant klasterizuotus ir neklasterizuotus atsitiktinai sugeneruotus duomenis. Tačiau vėlgi, realių duomenų atveju tikslumas žymiai didesnis (išskyrus Pagrindinių kreivių metodo atvejį).

3.1.6 Tyrimo išvados

Buvo atliktas 6 dimensijų mažinimo metodų algoritmų vykdymo greičio ir tikslumo tyrimas. Tikslumas vertintas pagal 3 kriterijus: Stress reikšmę, Spirmeno koeficientą ir Šenono entropiją. Dimensijų mažinimas atliktas su skirtingo tipo duomenimis: atsitiktinai sugeneruotais neklasterizuotais duomenimis, atsitiktinai sugeneruotais klasterizuotais duomenimis ir realiais finansiniais duomenimis.

Atsitiktinai sugeneruotų duomenų atvejais (klasterizuotiems ir neklasterizuotiems) buvo patvirtinta keletas taisyklių. Didėjant objektų skaičiui, vykdymo laikas ilgėja. Tačiau pradinis dimensijų kiekis neturi reikšmingos įtakos greičiui. Tikslumui galioja priešingos taisyklės. Didesnis objektų skaičius neturi įtakos tikslumui, tačiau didėjant pradinių dimensijų kiekiui, tikslumas mažėja.

Realių duomenų atveju pastebėta, jog pradinis dimensijų kiekis gali turėti nedidelės įtakos vykdymo laikui. Taip pat šių duomenų nepavyko apdoroti LLE metodu ir gauti Stress reikšmės ICA metodui. Tai rodo, jog realių duomenų apdorojimas yra labiau komplikuotas. Rezultatai parodė, kad didesnis objektų kiekis šiuo atveju lemia didesnį tikslumą.

Duomenų pobūdis neturi reikšmingos įtakos metodų vykdymo laikui, tačiau turi įtakos tikslumui. Su klasterizuotais duomenimis pasiekimas didesnis tikslumas. Tiksliausiai dimensijų mažinimas atliekamas su realiais duomenimis.

Tyrimo rezultatai parodė, kad tiksliausias buvo MDS metodas. PCA ir ICA buvo mažiau tikslūs, tačiau patys greičiausi. Prasčiausi rezultatai gauti panaudojus LLE ir pagrindinių kreivių metodus. Isomap yra lėtas metodas, tačiau kartais duoda tiksliausius rezultatus.

Gauti rezultatai patvirtina prielaidą, kad priklausomai nuo duomenų pobūdžio, jų apimties, analizės tikslų reikia naudoti skirtingus metodus. Be pradinės analizės, negalima iš anksto pasakyti, koks metodas duos geriausius rezultatus konkrečiu atveju. Todėl mūsų siūloma metodologija leidžia kiekviename analizės etape įvertinti situaciją ir parinkti tinkamą dimensijų mažinimo metodą.

3.2 Lygiagrečiųjų skaičiavimų metodų tyrimas

Didžiųjų duomenų vizualizavimui labai svarbus aspektas yra greitis. Būtina, kad siūloma metodologija būtų teisinga ne tik teoriniame lygmenyje, bet taip pat galėtų efektyviai veikti apdorojant realius didelės apimties/realaus laiko duomenis. Mūsų tikslas yra pasiūlyti tokią metodologiją ir ją realizuojančio įrankio architektūrą, kad sukurtas sprendimas leistų panaudoti paskirstytų sistemų išteklius ir debesų kompiuterijos pranašumus.

Šiame skyriuje analizuojami lygiagretaus skaičiavimo privalumai prieš nuoseklų skaičiavimą. Lygiagrečių skaičiavimų taikymas leidžia paspartinti didžiųjų duomenų vizualizavimo procesą. Šiame skyriuje detaliau apžvelgiamos OpenMP ir MPI technologijos bei programinė įranga, kuri įgalina jas panaudoti.

3.2.1 Mokslinių darbų apie lygiagretųjį skaičiavimą analizė

Pasak straipsnio [48] autorių, didėjantis gijų skaičius šiuolaikiniuose mikroprocesoriuose visas didelio našumo skaičiavimo (angl. *High Performance Computing*) sistemas perkelia į *peta* ir *exa* duomenų erą. Autoriai tyrė hibridinio MPI ir OpenMP technologijomis paremto lygiagretaus programavimo privalumus. Šie metodai buvo pritaikyti tiek realiems atvejams, tiek specialiai sumodeliuotoms situacijoms. Buvo gauti įvairūs našumo ir resursų panaudojimo rodikliai panaudojant 3 skirtingas lygiagretiems skaičiavimams skirtas daug gijų turinčias sistemas. Rezultatai atskleidė hibridinio programavimo privalumus, tačiau kartu buvo pastebėta dalykų,

kurie trukdo dar spartesniam veikimui: nepilnai apibrėžta sąsaja tarp MPI procesų ir OpenMP gijų bei pačios OpenMP technologijos spartos ribotumas.

Pasak straipsnio [90] autorių Rabenseifner, G. Hager ir kt., daugelis itin sparčiam skaičiavimui skirtų sistemų turi hierarchinę techninės įrangos struktūrą: jas sudaro bendros atminties mazgai su keletu daugiagijų procesorių, kurie komunikuoja tarpusavyje per tinklo infrastruktūrą. Todėl lygiagretus programavimas turi sujungti paskirstytos atminties lygiagretinimą tarp visų mazgų kartu su bendros atminties lygiagretinimu kiekvieno mazgo viduje.

D. A. Mallon, G. L. Taboada straipsnyje [65] palygino MPI ir OpenMP veikimo našumą ir nustatė, kad MPI paprastai pasiekia geresnius rezultatus, kai naudojama bendra atmintis, tačiau OpenMP kai kuriais atvejais yra pranašesnis. Pastaroji technologija turi tiesioginį priėjimą prie bendros atminties ir išvengia atminties kopijų darymo kaip MPI atveju.

P. D. Mininni ir D. Rosenberg darbe [73] pristatė hibridinę schemą, kuri panaudojama MPI paskirstytos atminties, o OpenMP bendros atminties lyagiagretinimui. Sistemos veikimo efektyvumas siekė 83 % panaudojant 20 000 gijų. Autoriai pasiūlė kaip pasirinkti optimalų MPI procesų ir OpenMP gijų kiekį norint optimizuoti kodo veikimą. Pasak jų, esant pakankamai sudėtingiems uždaviniams, optimalu yra pasirinkti 12 gijų (1 sistemos mazgui tenka vienas MPI procesas). Esant mažesniam apkrovimui, užduotys greičiau atliekamos su 6 gijomis.

Dar vienas MPI ir hibridinių sistemų palyginimas buvo atliktas F. Cappello, D. Etiemble darbe [8]. Jų tyrimo rezultatai atskleidė, kad pavienė MPI technologija daugeliu atveju yra greitesnė. Hibridinės sistemos tampa pranašesnės tik tuomet, kai greiti procesoriai pasiekia beveik idealų komunikacijos greitį ir lygiagretinimo lygis yra pakankamas.

Panašias išvadas gavo ir M. J. Chorley su D. W. Walker [10]. Jų tyrimo rezultatai parodė, kad hibridinis kodas sugaišta mažiau laiko komunikavimo procesams ir pasiekia geresnius rezultatus, kai yra didesnis gijų skaičius. Tačiau kai gijų kiekis yra nedidelis, hibridinio kodo sparta mažėja dėl šiuo atveju neefektyvių OpenMP procesų.

R. Rabenseifner, G. Hager ir kt. nustatė, kad įrangos topologija turi labai didelės reikšmės lygiagretaus skaičiavimo sistemų našumui [90]. F. Wolf ir B. Mohr darbe [106] pristatė sistemą, skirtą automatinei MPI ir OpenMP naudojančių lygiagretaus skaičiavimo sistemų veikimo analizei.

Šiame darbe lygiagretinamui pasirinktas Atsitiktinės projekcijos metodas, kuris remiasi matricų daugyba. Rifat Chowdhury [92] atliko panašų tyrimą ir sukūrė algoritmą, kuris matricų daugybai naudoja OpenMP. Rezultatai parodė, kad šis algoritmas naudojant 4 gijas veikė beveik 4 kartus greičiau nei kodas su 1 gija.

3.2.2 Dimensijų mažinimo metodo pritaikymas lygiagretiesiems skaičiavimams

Norint įvertinti lygiagretaus skaičiavimo algoritmų pranašumą prieš nuoseklaus vykdymo algoritmą, visų pirmą reikia pastarojo kodą pritaikyti OpenMP ir MPI metodams. Šiuo atveju bandymai atlikti su atsitiktinės projekcijos dimensijų mažinimo metodu, lygiagretiems skaičiavimams pritaikant šio metodo algoritmą.

3.2.2.1 Nuoseklus kodas

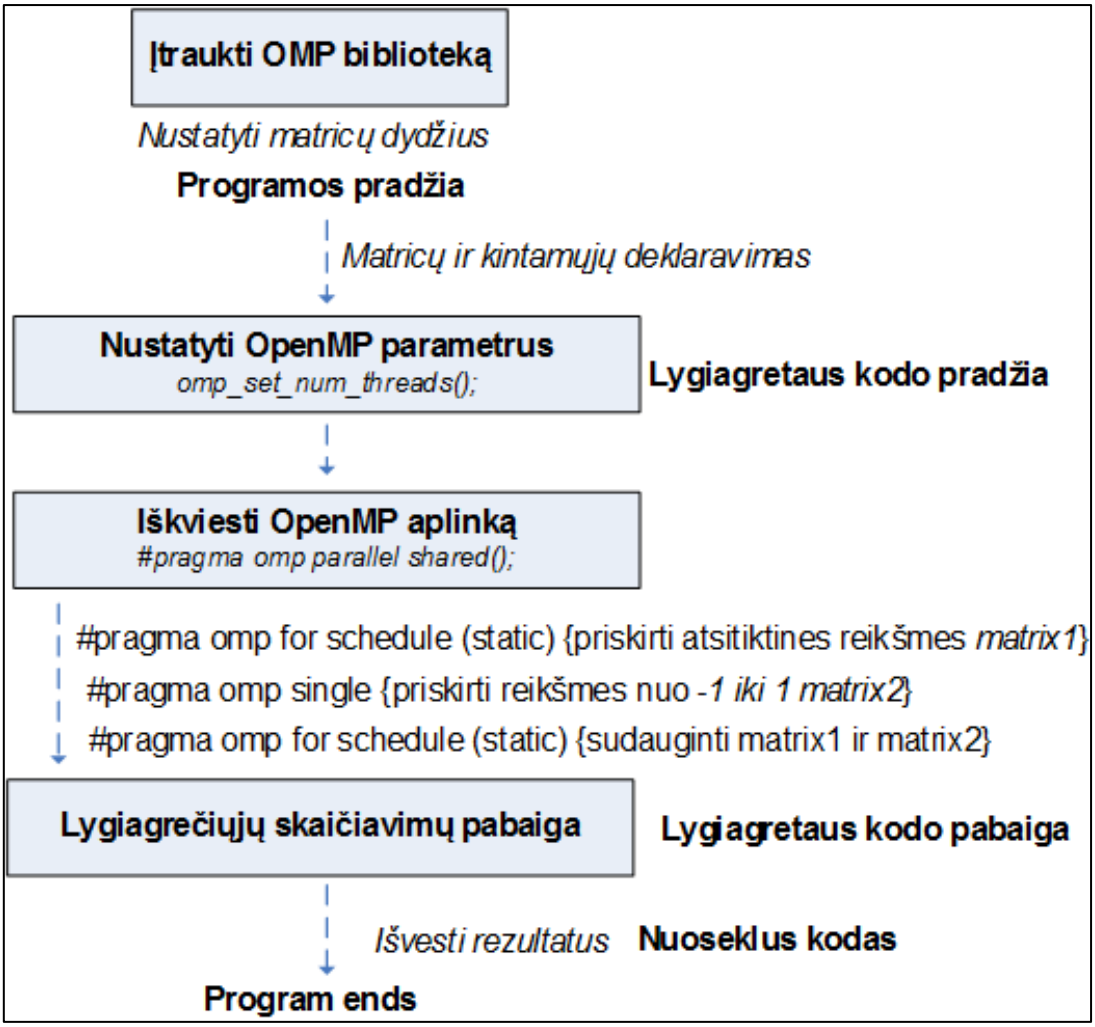
Pirmiausia apibrėžiamos 3 matricos: $n \times d$ matrix1, $d \times k$ matrix2 ir $n \times k$ matrix3. Pastaroji yra rezultatų (jau sumažintų dimensijų skaičiaus) matrica. Antrame žingsnyje pirmosios matricos elementams priskiriami pradiniai duomenys (arba atsitiktiniai skaičiai). Tuomet antrosios matricos elementams priskiriamos reikšmės 1 arba -1 su tikimybe $1/2$. Galiausiai pirmoji ir antroji matricos yra sudauginamos, o gautos reikšmės priskiriamos trečiajai matricai.

Kodas yra parašytas C kalba. Jis vykdomas kompiuteryje, turinčiame Intel Core i5-2450M 2.5 GHz procesorių ir 4 GB RAM. Naudojama CodeBlocks platforma su GNU GCC kompiliatoriumi. Šiuo atveju programos vykdymui naudojamas tik viena gija.

3.2.2.2 OpenMP kodas

Antruoju atveju naudojama OpenMP technologija atlikti lygiagrečius skaičiavimus personaliniame kompiuteryje ir taip panaudoti visas procesoriaus gijas. Pradinis nuoseklus kodas turi būti pertvarkytas pagal OpenMP reikalavimus.

OpenMP algoritmas pavaizduotas 38 paveiksle. *omp.h* biblioteka yra įraukiama į kodą. Po matricų ir darbinių kintamųjų apibrėžimo yra nustatomas naudojamų gijų kiekis. Funkcija *#pragma omp for schedule ()* yra naudojama pradinės matricos sukūrimui ir matricų daugybai. Tai reiškia, jog šios programos dalys yra vykdomos lygiagrečiu rėžimu. Funkcija *#pragma omp single* leidžia ir viduryje kodo esančias eilutes vykdyti nuosekliu rėžimu (pritaikyta antros matricos generavimui). Baigus visus lygiagrečius skaičiavimus yra išvedami rezultatai.

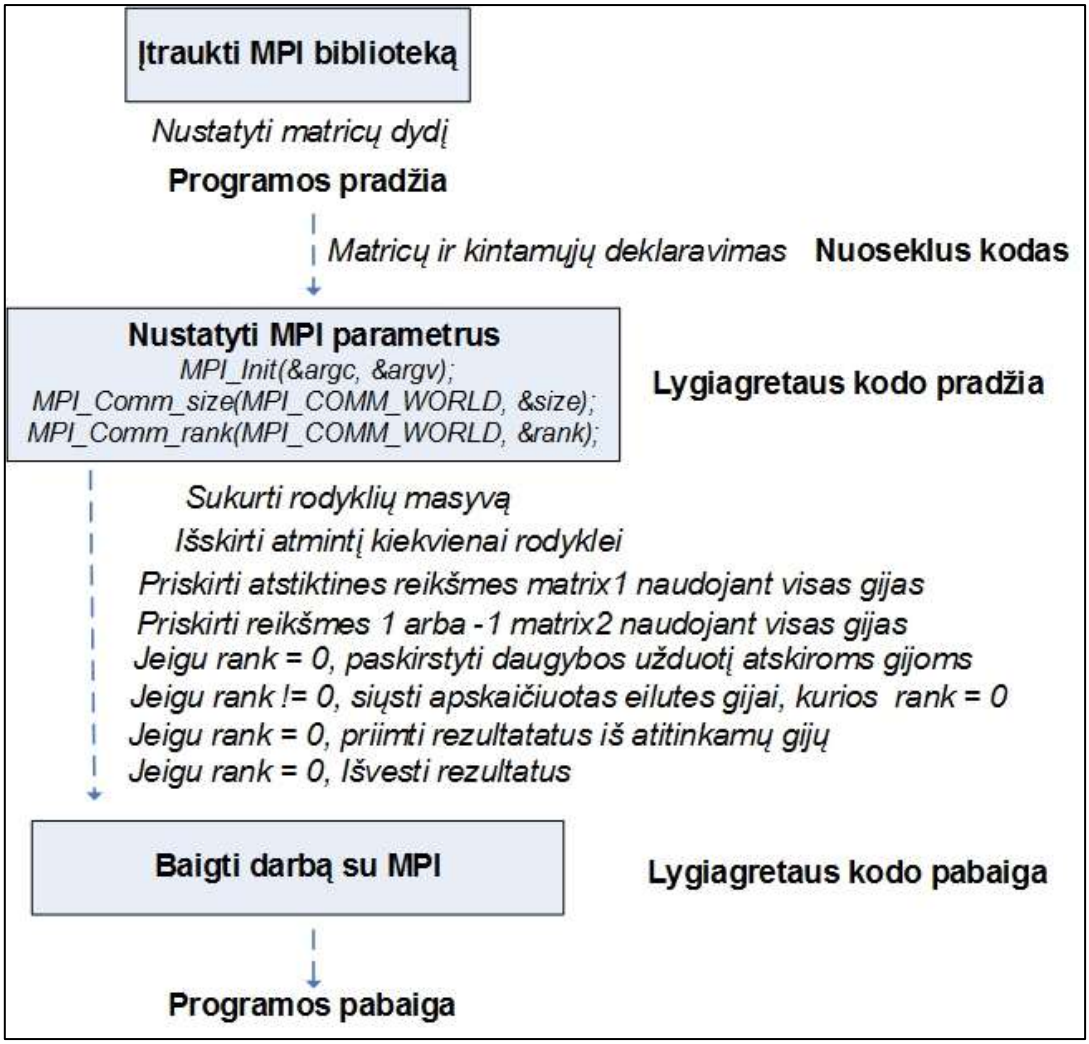


38 pav. OpenMP algoritmo schema

3.2.2.3 MPI kodas skaičiavimams klasteryje

Siekiant išbandyti MPI technologiją buvo pasitelktas VU Matematikos ir informatikos instituto turimas kompiuterių klasteris, veikiantis Linux operacinėje sistemoje.

Iš viso klasteris turi 120 gijų. Pradinis nuoseklus kodas buvo atitinkamai modifikuotas. Į kodą įtraukiamos MPI bibliotekos. Visos matricos ir kintamieji yra deklaruojami nuosekliame kode, tačiau MPI yra inicializuojamas prieš pradedant lygiagrečius skaičiavimus. Programa naudoja iš anksto apibrėžtą gijų skaičių. „Nulinė“ gija paskirsto užduotis kitoms gijos ir surenka rezultatus. Visos skaičiavimo operacijos yra atliekamos lygiagrečiai visų gijų (39 pav.).



39 pav. MPI algoritmo schema

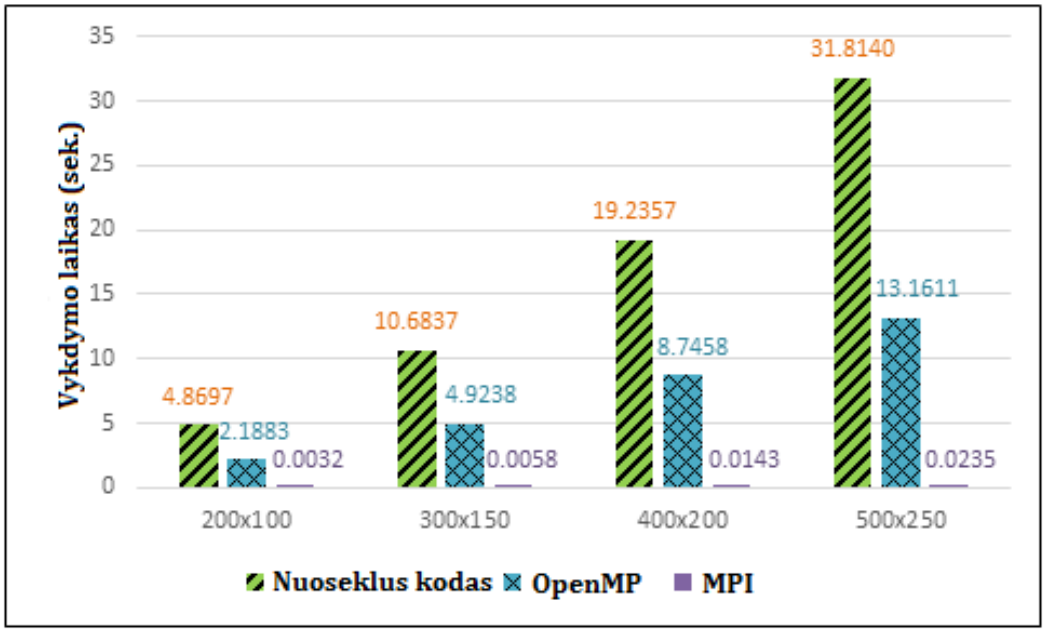
3.2.3 Lygiagrečiųjų skaičiavimų metodų spartos palyginimas

Šioje dalyje pristatomi rezultatai, kurie buvo gauti pritaikius Atsitiktinės projekcijos metodą skirtingo dydžio matricoms. Palyginami užduočių vykdymo laikai naudojant nuoseklų kodą, lygiagretų kodą personaliniame kompiuteryje (OpenMP) bei MPI kodą kompiuterių klasteryje. Visais atvejais pradinis dimensijų skaičius mažinamas per pusę.

Palyginimui naudojami du dydžiai, naudojami lygiagretiesiems algoritmams analizuoti. Lygiagrečiojo algoritmo spartinimo koeficientu vadinamas santykis $S_p = T_0/T_p$, įvertinantis pagreitėjimą, kuris pasiekiamas sprendžiant uždavinį lygiagrečiuoju algoritmu ir naudojant p procesorių. Čia T_p žymi laiką, per kurį duotas uždavinys išsprendžiamas lygiagrečiuoju algoritmu, naudojant p procesorių. T_0 yra laikas, per kurį tas pats uždavinys išsprendžiamas greičiausiu nuosekliuoju algoritmu.

Algoritmo efektyvumo koeficientas parodo, kokią dalį procesorių pajėgumo pasitelkiama sprendžiant uždavinį duotuoju lygiagrečiuoju algoritmu: $E_p = S_p/p$.

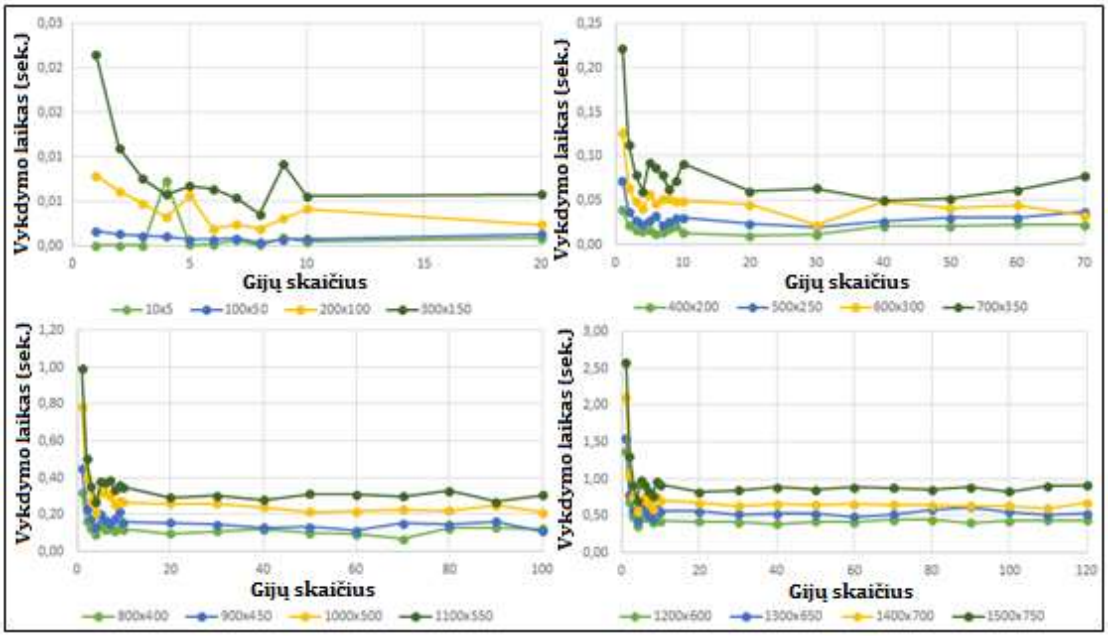
40 paveiksle palyginti programos vykdymo laikai apdorojant sąlyginai nedideles matricas. Rezultatai rodo, kad personaliniame kompiuteryje OpenMP kodo veikimas su 4 gijomis yra 2-3 kartus greitesnis negu naudojant 1 giją. Skirtumas auga, jeigu didėja apdorojamų matricų dydžiai. Tačiau didžiausias efektas jaučiamas vykdant MPI kodą klasteryje. MPI programa tas pačias užduotis įvykdė nuo 100 iki 1000 karto greičiau palyginus su OpenMP kodu, vykdomu viename kompiuteryje.



40 pav. OpenMP ir MPI algoritmų vykdymo spartos palyginimas

Tai atskleidžia visų pirmą kompiuterių klasterio naudą, tačiau kadangi nėra teisinga lyginti skirtingus metodus kai jie veikia skirtingose platformose, toliau visi bandymai su didesnėmis matricomis buvo atliekami tik klasteryje. Klasteryje MPI kodas su 4 gijomis veikė keletą kartų greičiau nei kodas su viena gija. Svarbu pažymėti, kad pradinei matricai pasiekus dydį 1950×950 nuoseklus kodas nebepajėgus atlikti užduoties. Todėl dirbant su didelės apimties duomenimis lygiagretus programavimas ir kompiuterių klasterio infrastruktūra yra neišvengiama būtinybė.

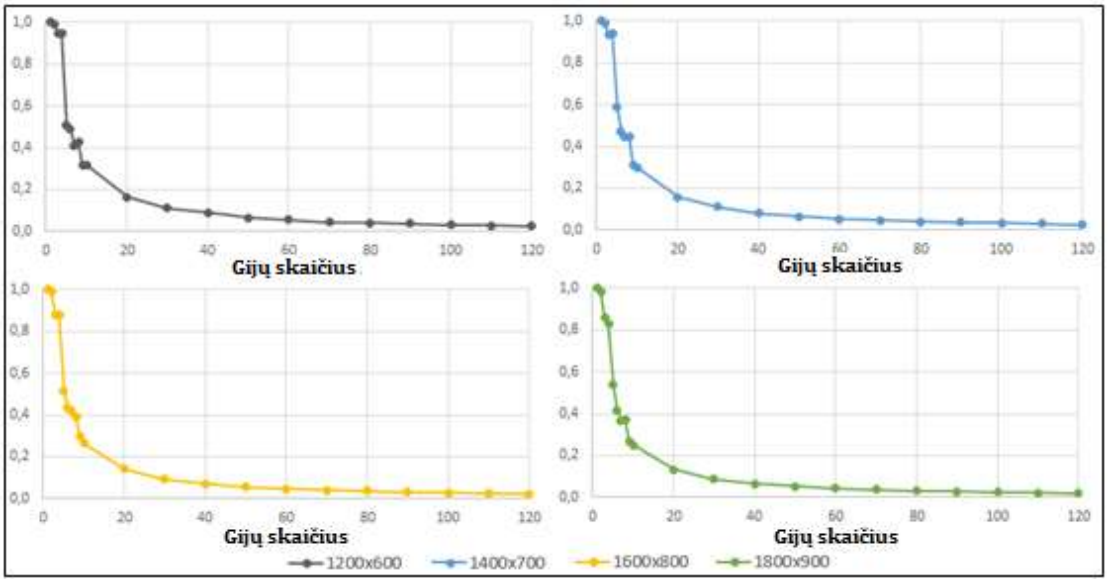
Gali atrodyti, jog visais atvejais didinant naudojamų gijų skaičių, kodo vykdymo laikas turėtų trumpėti. Tačiau gauti rezultatai tai paneigia. Tos pačios dimensijų mažinimo užduotys buvo pakartotos pamažu didinant naudojamų gijų skaičių.



41 pav. MPI algoritmo vykdymo laikas naudojant skirtingą gijų kiekį

41 paveikslas rodo, jog šiuo atveju norint sumažinti pradinį duomenų dimensijų skaičių perpus, iš esmės nėra prasmės naudoti daugiau nei 20 gijų. Didžiausias efektas matomas pačioje pradžioje, o vėliau jis mažėja. Tačiau galima rasti atvejų, kai užduotys greičiausiai įvykdomos naudojant 20 arba netgi 100 gijų.

Kitame žingsnyje dar buvo paskaičiuotas ir efektyvumo koeficientas. 42 paveikslas rodo, kad taikant lygiagretųjį skaičiavimą, kuo mažesnis gijų skaičius naudojamas, tuo labiau kiekviena iš jų yra apkraunama. Todėl ir efektyvumo koeficiento reikšmės mažėja, didėjant naudojamų gijų skaičiui. Galima daryti išvadą, kad nedidelių matricų apdorojimo užduotys yra per lengvos, norint didelį gijų kiekį. Tuomet išnaudojama tik nedidelė kiekvienos gijos pajėgumo dalis.



42 pav. Algoritmo efektyvumo koeficiento reikšmės

3.2.4 Tyrimo išvados

Šiame darbe analizuota, kaip lygiagrečiųjų skaičiavimų taikymas gali paspartinti duomenų dimensijų mažinimo bei vizualizavimo užduotis.

Išanalizavus OpenMP ir MPI lygiagretaus skaičiavimo technologijas buvo nustatyti jų pranašumai. Atlikti bandymai parodė, kad vykdant lygiagretiems procesams pritaikytą OpenMP kodą personaliniame kompiuteryje užduotys yra įvykdomos keletą kartų greičiau nei naudojant nuoseklų kodą. Kuo didesni duomenys apdorojami, tuo efektas didesnis. Dimensijų mažinimą vykdant kompiuterių klasteryje panaudojant MPI technologiją programų vykdymo laikas buvo šimtus kartų trumpesnis, nei tai darant personaliniame kompiuteryje su OpenMP kodu. Lyginant lygiagretaus kodo vykdymą klasteryje ir nuoseklaus kodo vykdymą viename kompiuteryje, laiko skirtumas būtų dar didesnis. Šie rezultatai parodo, kaip stipriai lygiagretaus skaičiavimo metodų ir kompiuterių klasterių naudojimas paspartina duomenų analizę.

Didesnis naudojamų gijų skaičius ne visada lemia didesnę spartą. Esant mažam duomenų kiekiui, per didelis naudojamų gijų skaičius gali išauginti duomenų persiuntimo kaštus ir sumažinti efektyvumą.

Dimensijų mažinimas panaudojant lygiagrečius skaičiavimus buvo išbandytas realių finansinių duomenų apdorojimui. Tas užduotis, kurių buvo nebeįmanoma atlikti su asmeniniu kompiuteriu, panaudojant MPI technologiją, pavyko įgyvendinti greičiau nei per 0,4 sekundės.

Galima teigti, kad be lygiagrečiųjų skaičiavimų didžiųjų duomenų apdorojimas yra neefektyvus ir netgi neįmanomas. Tam kad siūloma metodologija galėtų efektyviai veikti apdorojant realius didelės apimties duomenis, ji ir ją realizuojantis įrankis turi būti pritaikyti veikimui paskirstytose sistemose. Todėl disertacijoje siūlomą metodologiją realizuojantis įrankis yra parašytas R kalba, o jo programinis kodas padalintas į dvi pagrindines dalis – serverio (Server.R) ir kliento (ui.R). Strategiją realizuojantis kodas (duomenų užkrovimas, analizė, dimensijų mažinimas, duomenų vizualizavimas grafikuose) yra serverio dalyje. Ši R kodo dalis gali būti lengvai pritaikyta lygiagretiems skaičiavimams panaudojant pvz. MPI/CUDA ir skaičiavimus vykdant debesyje (pvz. MS Azure/AWS)

4 Daugiapakopis didžiųjų duomenų vizualizavimas

Didžiųjų duomenų analizė įgalina atrasti paslėptą informaciją ir panaudoti ją efektyvesnių sprendimų priėmimui. Svarbi tokios analizės dalis yra duomenų vizualizavimas, nes būtent jis įgalima pastebėti paslėptus ryšius tarp objektų, ką yra sunku padaryti taikant standartinius analizės metodus [110].

Mūsų pagrindinis tikslas yra patobulinti duomenų vizualizavimo procesą ir pasiūlyti naujų efektyvesnių būdų didžiųjų duomenų vizualizavimui. Šiame skyriuje pristatoma daugiapakopė didžiųjų duomenų vizualizavimo strategija, paremta dimensijų mažinimo metodais. Taip pat pristatomas įrankis, realizuojantis siūlomą strategiją. Įrankis sukurtas panaudojant R kalbą ir Shiny paketą.

Daroma prielaida, kad analizės proceso pradžioje (kai apdorojami visi duomenys) svarbiausia yra greitis. Tolesniuose žingsniuose pasirenkant mažesnes duomenų dalis vis labiau išauga tikslumo svarba. Ankstesni tyrimai parodė, kad dimensijų mažinimo tikslumas priklauso nuo duomenų pobūdžio bei pradinių dimensijų kiekio. Todėl siūloma metodologija ir įrankis leidžia kiekviename žingsnyje pasirinkti labiausiai tinkantį metodą (pateikiami vizualizacijos pavyzdžiai ir statistiniai rodikliai).

Antroje šio skyriaus dalyje detaliai aprašoma siūloma metodologija, o trečiame pristatomos įrankio galimybės.

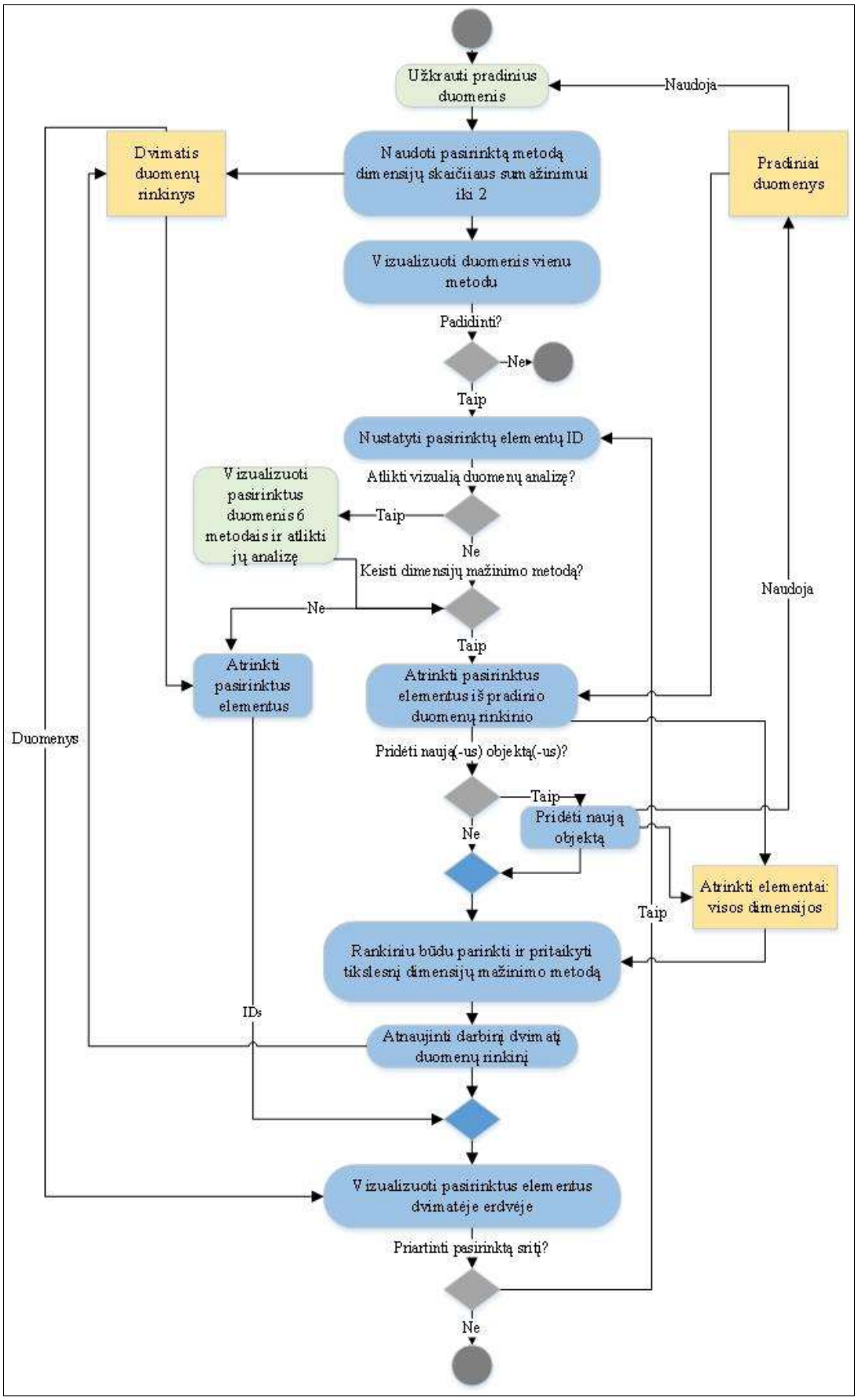
4.1 Daugiapakopio duomenų vizualizavimo strategija

Šiame skyriuje aprašoma didžiųjų duomenų vizualizavimo strategija, kuri suskaido vizualizavimo procesą į atskirus žingsnius (43 pav.). Kiekviename žingsnyje tam tikras dimensijų mažinimo metodas gali būti pritaikytas atsižvelgiant į duomenų tipą ir kiekį. Metodai parenkami pagal jų algoritmų vykdymo greitį ir tikslumą. Kuo daugiau pradinės informacijos metodas išsaugo, tuo jis tikslesnis. Naudojami tikslumo rodikliai plačiau aprašyti 4.2.2 skyriuje. Kai duomenys apdorojami ir vizualizuojami, yra galimybė peržiūrėti visų klasterių parametrų statistinius duomenis. Tolesnę analizę/vizualizavimą galima atlikti tik su pasirinktais duomenų elementais (grafike pažymint norimą plotą).

Duomenų vizualizavimo procesas gali būti suskirstytas į šiuos etapus:

1) Visų pirma į sistemą užkraunami pradiniai duomenys. Detalus šio žingsnio aprašymas pateikiamas 4.2.1 skyriuje.

2) Analizės pradžioje tikslumas nėra toks svarbus, todėl galima panaudoti greičiausią dimensijų mažinimo metodą. Sukuriamas naujas 2 dimensijų duomenų rinkinys, duomenys atvaizduojami dvimačiame grafike.
3) Analitikas gali grafike pažymėti visus arba dalį duomenų tolesnei analizei/vizualizavimui. Jeigu pageidaujama, pateikiami pasirinktos duomenų aibės vizualizacijos skirtingais metodais pavyzdžiai. Taip pat pateikiami vizualizavimo tikslumo rodikliai ir kita statistinė informacija. Detaliau šios funkcijos aprašytos 4.2.3 skyriuje.
4) Pateikiami vizualizacijos pavyzdžiai ir tikslumo rodikliai leidžia lengviau pasirinkti, kokį metodą taikyti konkrečiame žingsnyje, pvz. palikti prieš tai naudotą metodą, ar naudoti kitą. Galima daryti prielaidą, jog analizės pradžioje dažniau bus pasirenkami greitesni metodai, o vėliau – tikslesni.
    a) Jeigu atliekamas paprastas vizualizavimo mastelio keitimas (pasirinktos srities priartinimas nekeičiant dimensijų mažinimo metodas), tuomet pasirinkti duomenų elementai gali būti atfiltruojami iš 2 dimensijų duomenų rinkinio, sukurto 2 žingsnyje. Tokiu atveju nėra būtinybės kartoti dimensijų mažinimo procesą, duomenys greičiau atvaizduojami grafike.
    b) Jeigu pasirenkama taikyti kitą metodą arba duomenis tuo pačiu metodu apdoroti iš naujo, tuomet pasirinkti elementai (duomenų objektai) atfiltruojami iš pradinio duomenų rinkinio, turinčio visas pradines dimensijas. Prieš atliekant šią operaciją yra galimybė pridėti daugiau objektų (iš pasirinkto duomenų šaltinio). Tuomet pasirenkamas pageidaujamas dimensijų mažinimo metodas, kuriuo apdorojami duomenys. „Darbinis“ 2 dimensijų duomenų rinkinys yra atnaujinamas, o šie rezultatai atvaizduojami 2D grafike.
5) Jeigu analitikas pasirenka tęsti analizę su pasirinktais duomenų elementais, tuomet procesas kartojamas nuo 3 žingsnio.



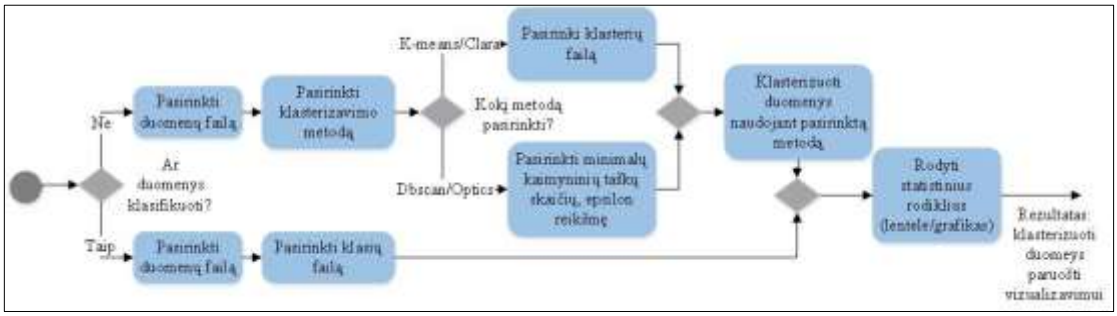
43 pav. Daugiapakopio duomenų vizualizavimo metodologija

4.1.1 Pradinių duomenų užkrovimas ir įvertinimas

Duomenų užkrovimo ir pradinės analizės principinė schema pavaizduota 44 paveiksle. Jeigu norima vizualizuoti jau klasifikuotus duomenis (elementai iš anksto priskirti tam tikrai klasei), tuomet reikia pasirinkti ir užkrauti duomenų failą (saugomi duomenų elementų parametrai) bei klasių failą (jame nurodyta, kokiai klasei priklauso kiekvienas elementas).

Jeigu norima vizualizuoti neklasifikuotus duomenis, tuomet pradžioje pasirenkamas tik duomenų failas. Norimas klasterizavimo metodas ir jo parametrai pasirenkami kitame žingsnyje. Kokius parametrus reikia nurodyti priklauso nuo pasirinkto metodo. Nurodžius parametrus pradiniai duomenys yra suklasifikuojami, pateikiami klasifikavimo rezultatai. Jeigu vartotojo jie netenkina, galima pakeisti klasifikavimo parametrus ir suskirstyti elementus į klases iš naujo.

Galiausiai, tiek klasifikuotų, tiek neklasifikuotų pradinių duomenų atvejais, yra galimybė peržiūrėti jų statistinius duomenis (lentelių ir grafikų pavidale).



44 pav. Duomenų užkrovimo ir analizės schema

4.1.2 Duomenų analizė ir metodų parinkimas

Siekiant geriau pasirinkti dimensijų mažinimo metodą, kuris pasirinktiems duomenims bus pritaikytas kitame vizualizavimo proceso etape, galima atlikti detalią šių duomenų analizę.

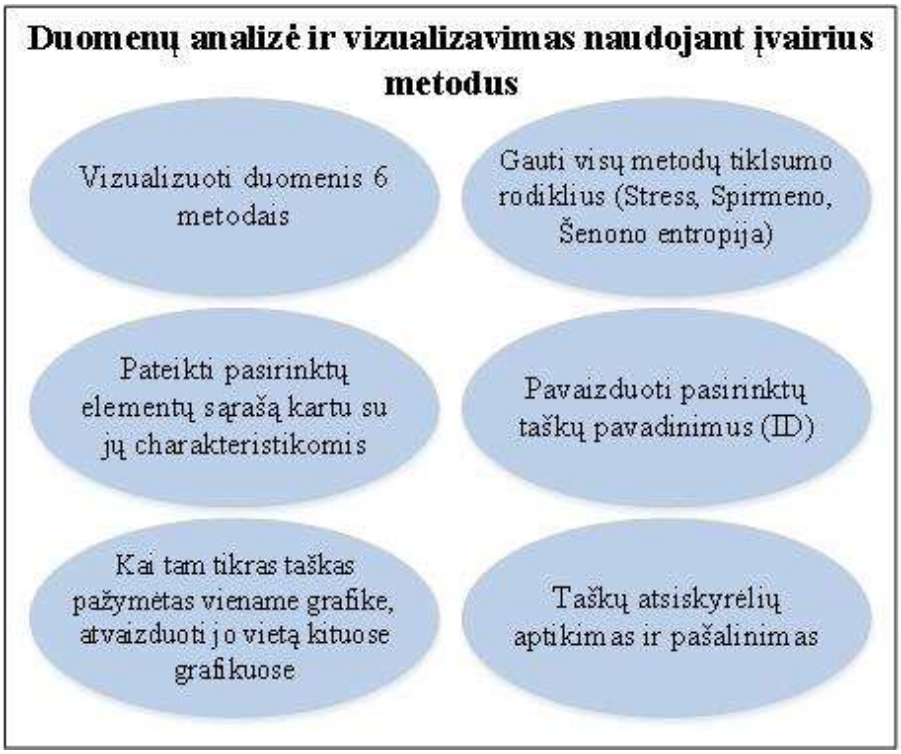
Dimensijų mažinimui naudojami šie metodai: Daugiamačių skalių (MDS), Pagrindinių komponenčių analizės (PCA), Nepriklausomų komponenčių analizės (ICA), Pagrindinių kreivių (PC), Lokaliai tiesioginio įterpimo (LLE) ir Isomap [70], [93], [102], [75], [93], [102]. Daugiamačius duomenis apdorojus šiais metodais yra gaunami 6 dvimačiai duomenų rinkiniai (po vieną kiekvienam metodui). Jie atvaizduojami 6 sklaidos diagramose. Tie patys duomenys gali būti atvaizduojami skirtingai priklausomai nuo pritaikyto dimensijų mažinimo metodo. Galimybė

pasirinkti iš kelių metodų leidžia pritaikyti tą, kuris geriausiai tinka konkrečiu atveju (tam tikro pobūdžio duomenims). Greta vizualizavimo pavyzdžių apskaičiuojami ir pateikiami tikslumo rodikliai: Stress reikšmė, Spirmeno koeficientas, Šenono entropija (rodikliai plačiau aprašyti 3.1.3 skyriuje):

- **Stress.** MDS metodo Stress reikšmės radimui naudojama R funkcija *mds()* iš paketo 'smacof' [88]. Kitiems metodams ši reikšmė apskaičiuojama pagal Stress formulę.
- **Spirmeno koeficientas**
- **Šenono entropija**

Siūloma metodologija taip pat palaiko šias galimybes:

- Lentelėje pateikiamas sąrašas pasirinktų duomenų elementų kartu su jų pradiniais parametrais;
- Grafike pasirinktiems elementams galima uždėti jų pavadinimus (arba ID);
- Viena grafike pasirinkus tam tikrą elementą, jis paryškinamas kituose grafikuose (kuriuose duomenys atvaizduoti pritaikius kitus metodus);
- Anomalijų aptikimas. Anomaliniai taškai yra tie, kure itin skiriasi nuo likusių analizuojamų duomenų. Tai gali reikšti matavimų netikslumą, eksperimento klaidas arba reikšmingą naujumą [84].



45 pav. Metodologijos duomenų analizės galimybės

4.2 Duomenų vizualizavimo įrankio galimybių pristatymas

Siekiant pademonstruoti siūlomos metodologijos galimybes, buvo sukurtas įrankio prototipas, kuris realizuoja siūlomą duomenų vizualizavimo metodologiją. Šiame skyriuje aprašomi testiniai duomenų rinkiniai ir panaudojimo atvejai, parodantys įrankio galimybes.

4.2.1 Pirmo duomenų rinkinio analizė

4.2.1.1 Duomenų rinkinio aprašymas

Įrankio prototipo pristatymui panaudotas duomenų rinkinys, saugantis informaciją apie varles [64]. Kitų mokslininkų šis duomenų rinkinys anksčiau buvo panaudotas kitoms klasifikavimo užduotims, kai buvo siekiama atpažinti varlių rūšis pagal jų skleidžiamą garsą. Šis duomenų rinkinys buvo gautas apdorojus 60 audio įrašų, kuriuose saugoti 4 šeimų, 8 genčių ir 10 rūšių varlių duomenys. Iš viso rinkinyje yra 7195 objektai, mūsų atveju pristatymui naudojama tik dalis jų.

Kadangi duomenys yra jau suklasifikuoti, todėl reikia pasirinkti 2 failus:

- **Duomenų failas**. Jame yra 2610 objektų, kiekvieną nusako 10 parametrų (dimensijų).
- **Klasių failas**. Jame yra 2610 elementų (po vieną kiekvienam objektui duomenų faile) ir vienas parametras, kuris nusako, kokiai klasei konkretus objektas priklauso. Viso objektai yra suskirstyti į 4 klases (šeimas). 68 objektai priklauso pirmai klasei (Bufonidae), 542 objektai priklauso antrai klasei (Dendrobatidae), 1000 objektų priklauso trečiai klasei (Hylidae), o likęs 1000 objektų priklauso 4 klasei (Leptodactylidae).

4.2.1.2 Duomenų užkrovimas

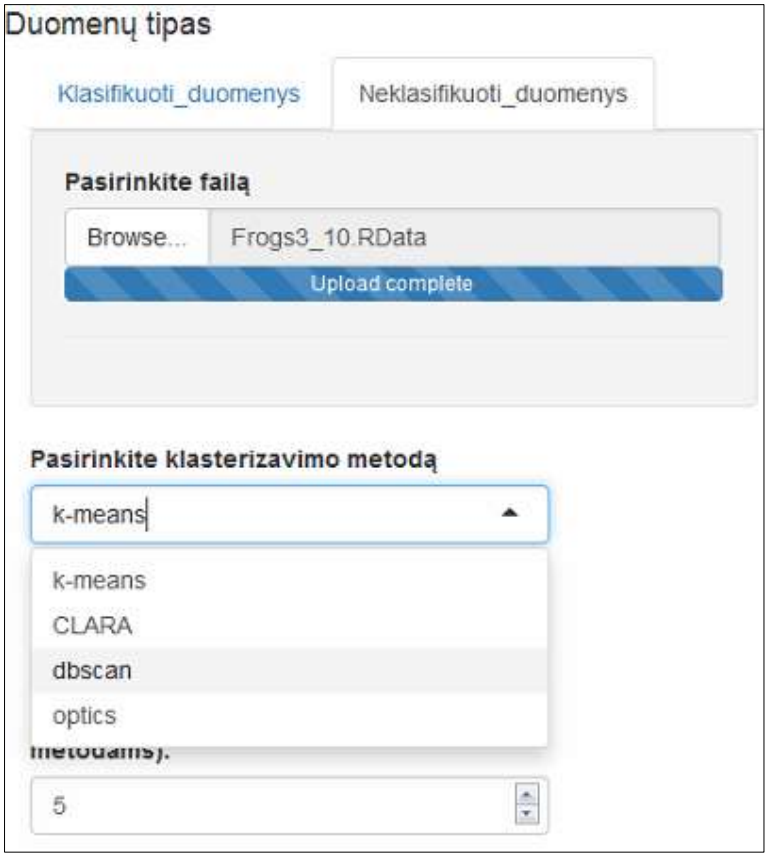
Pradžioje pasirenkamas duomenų tipas – klasifikuoti/neklasifikuoti duomenys. Šiame skyriuje pristatomas neklasifikuotų duomenų atvejis, todėl užtenka pasirinkti vieną failą. Pirmame žingsnyje pasirenkamas skirtukas „Neklasifikuoti duomenys“. Kitame žingsnyje reikia pasirinkti klasifikavimo parametrus (46 pav.).

Jeigu pasirenkamas *K-means* metodas, tuomet užtenka nurodyti norimą gauti klasterių skaičių. Įrankis parodo, kiek objektų pateko į kiekvieną klasterį, pvz. pirmas klasteris turi 606 objektus, antras – 160 ir t.t. (47 pav.).

Jeigu pasirenkamas *dbscan* klasifikavimo metodas, tuomet reikia nurodyti klasterių kiekį, kaimyninių elementų kiekį ir epsilon reikšmę (48 pav.). Epsilon reikšmė

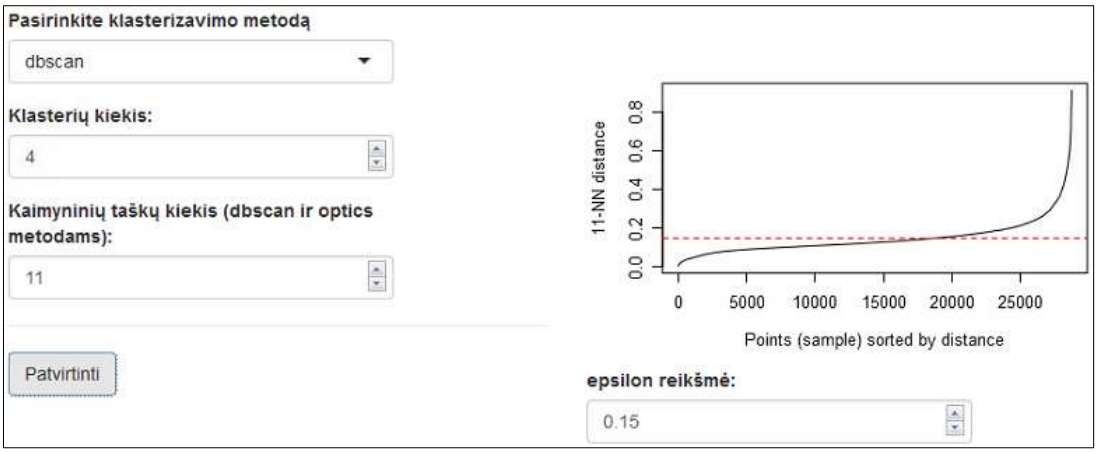
parenkama pagal grafiką – ieškoma taško, kuriame pastebimas didžiausias atstumo tarp kaimyninių elementų pasikeitimas. Šiuo atveju parinkta epsilon reikšmė yra 0.3.

Pasirinkus *dbscan* metodą ir nurodžius 3 klasterius, į pirmą pateko 2120 elementai, į antrą 371, o į trečią 13.



46 pav. Pradinių duomenų užkrovimo langas

47 pav. Klasterių, gautų *K-means* metodu, statistiniai duomenys



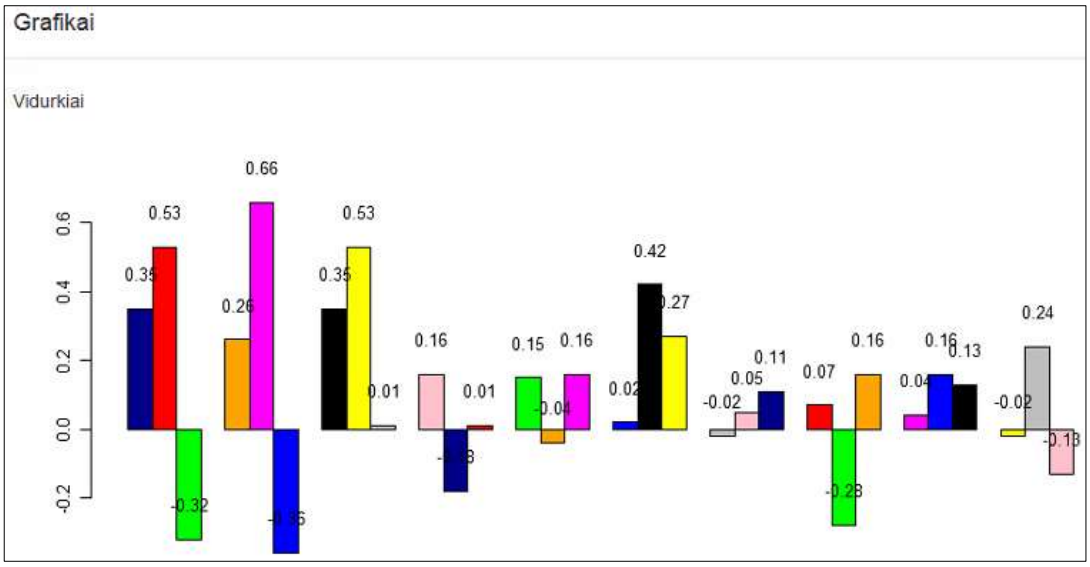
48 pav. *dbscan* metodo parametrų nustatymas

Nepriklausomai nuo klasifikavimo metodo pasirinkimo, visuomet parodomi gautų klasterių elementų statistiniai duomenys: vidurkis, standartinis nuokrypis, minimalios ir maksimalios reikšmės (49 pav.). Paspaudus mygtuką „Rodyti grafiką“ statistiniai rodikliai atvaizduojami grafikuose (50 pav.).

Statistika

```
, , 1

          [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10]
Vidurkis  0.35  0.26  0.35  0.16  0.15  0.02 -0.02  0.07  0.04 -0.02
SD        0.21  0.31  0.15  0.17  0.10  0.15  0.13  0.17  0.18  0.17
Min      -0.29 -0.25 -0.14 -0.41 -0.16 -0.34 -0.37 -0.42 -0.50 -0.44
Max       1.00  1.00  0.79  0.63  0.56  0.70  0.39  0.45  0.48  0.52

, , 2

          [,1] [,2] [,3]  [,4]  [,5] [,6]  [,7]  [,8] [,9] [,10]
Vidurkis 0.53 0.66 0.53 -0.18 -0.04 0.42  0.05 -0.28 0.16  0.24
SD       0.06 0.07 0.06  0.05  0.06 0.05  0.05  0.05 0.05  0.05
Min      0.32 0.47 0.32 -0.37 -0.22 0.27 -0.11 -0.44 0.03  0.01
Max      0.77 1.00 0.74 -0.03  0.17 0.55  0.19 -0.11 0.31  0.41

, , 3

          [,1]  [,2]  [,3]  [,4] [,5] [,6] [,7] [,8] [,9] [,10]
Vidurkis -0.32 -0.36  0.01  0.01 0.16 0.27 0.11 0.16 0.13 -0.13
SD        0.04  0.05  0.04  0.03 0.04 0.03 0.03 0.03 0.02  0.04
Min      -0.40 -0.44 -0.05 -0.04 0.11 0.22 0.03 0.10 0.10 -0.20
Max      -0.26 -0.28  0.07  0.06 0.23 0.33 0.15 0.21 0.15 -0.07
```

49 pav. Duomenų klasterių statistiniai rodikliai



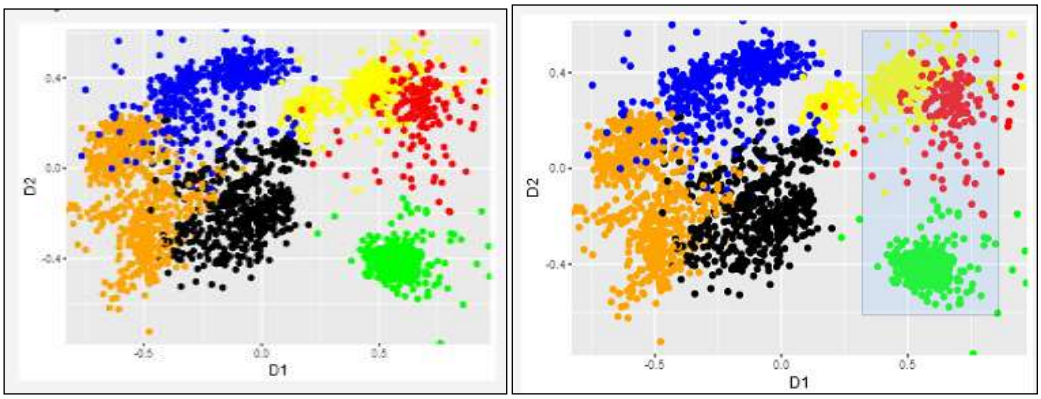
50 pav. Dimensijų vidutinės reikšmės kiekviename klasteryje

4.2.1.3 Daugiapakopis duomenų vizualizavimas ir analizė

Šiuo atveju pradiniam duomenų vizualizavimui iš 6 metodų buvo parinktas PCA:

51 pav. Dimensijų mažinimo metodų pasirinkimas

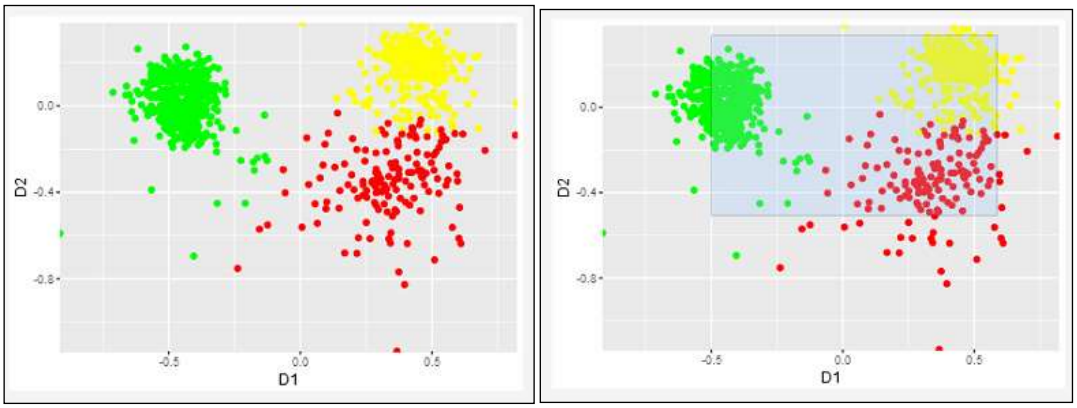
52 paveiksle pateikiami pradiniai duomenys (įeina visi objektai) vizualizuoti PCA metodu. Skirtingi klasteriai nuspalvinti skirtingomis spalvomis. Kai kurie klasteriai turi daugiau objektų (pvz. mėlynas, rudas, juodas), kiti – mažiau (pvz. raudonas, geltonas, žalias). Kai kurie klasteriai persidengia. Įrankis leidžia grafike pasirinkti dalį (jeigu reikia - ir visus) taškų tolimesnei analizei (52 pav., dešinėje).



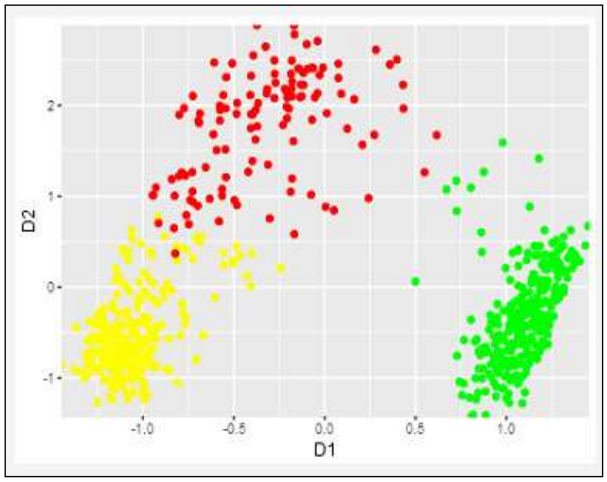
52 pav. Duomenų vizualizavimas PCA metodu

Pasirinkti duomenys gali būti apdoroti kitu dimensijų mažinimo metodu, nei naudotas prieš tai buvusiame žingsnyje. Šiuo atveju buvo panaudotas MDS metodas

(53 pav.). Tokiu principu „priartinant vaizdą“ ir pasirinktus duomenis vizualizuoti norimu metodu galima neribotą kiekį kartų. 54 paveiksle pateikta 53 paveiksle pavaizduotos pasirinktos duomenų srities duomenų vizualizacija, pritaikius ICA dimensijų mažinimo metodą. Grafikuose matoma, kad žalias klasteris aiškiai atsiskiria nuo kitų duomenų. Tuo tarpu raudonas ir geltonas dalinai persidengia.

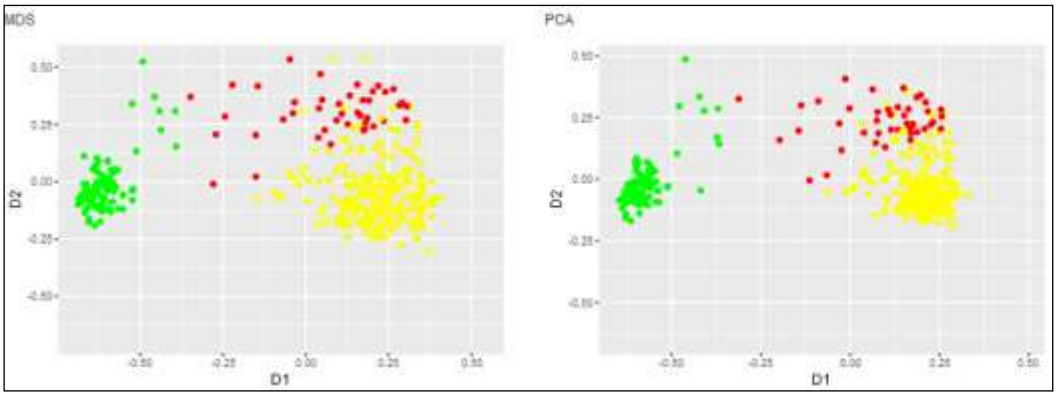


53 pav. Duomenų vizualizavimas MDS metodu

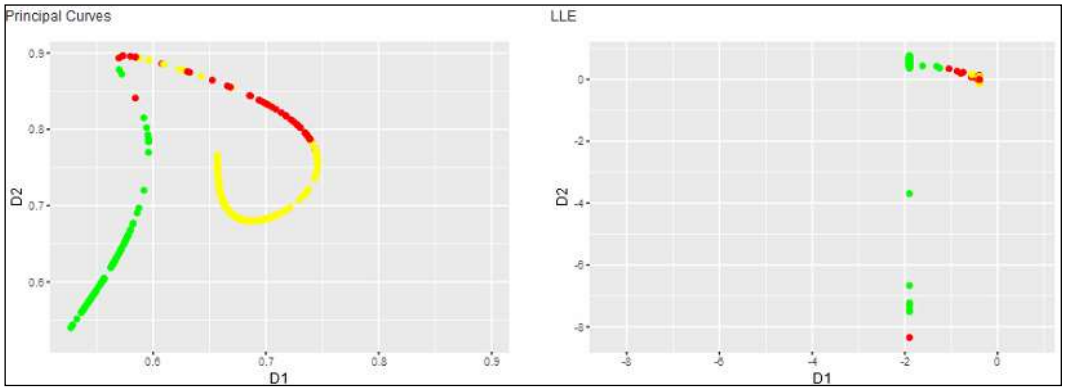


54 pav. Duomenų vizualizavimas ICA metodu

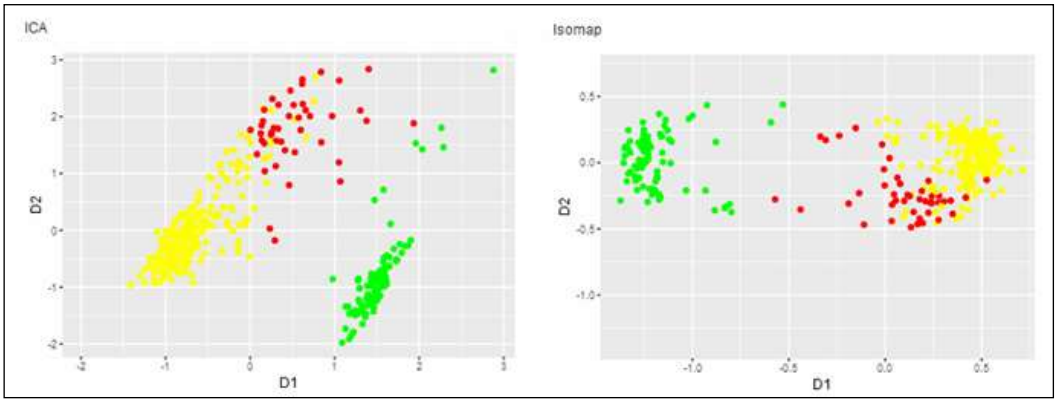
Jeigu vartotojas nėra tikras, kurį metodą taikyti, pasirinktus duomenų objektus galima iš karto vizualizuoti visais 6 metodais. Sistema pateikia vizualizavimo pavyzdžius, o vartotojas pasirenka norimą metodą. 55, 56 ir 57 paveiksluose vizualizuota dalis duomenų (buvo pasirinkta dalis duomenų, gautų praeitame žingsnyje), panaudojant MDS, PCA, ICA, PC, LLE ir Isomap metodus. Taip pat kiekvienam metodui pateikiami tikslumo rodikliai (58 pav). 59 paveiksle pateikiami tikslumo rodikliai grafiniame pavidale.



55 pav. Duomenų vizualizavimas MDS ir PCA metodais



56 pav. Duomenų vizualizavimas Pagrindinių kreivių ir LLE metodais

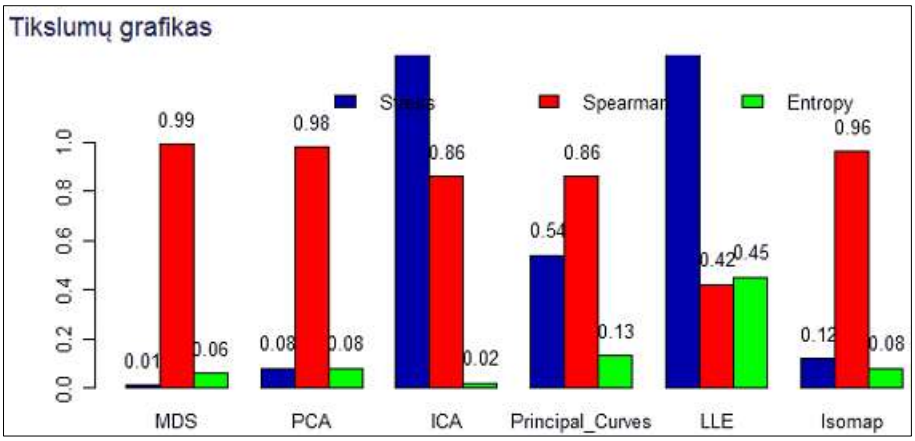


57 pav. Duomenų vizualizavimas ICA ir Isomap (k=10) metodais

Tikslumai:

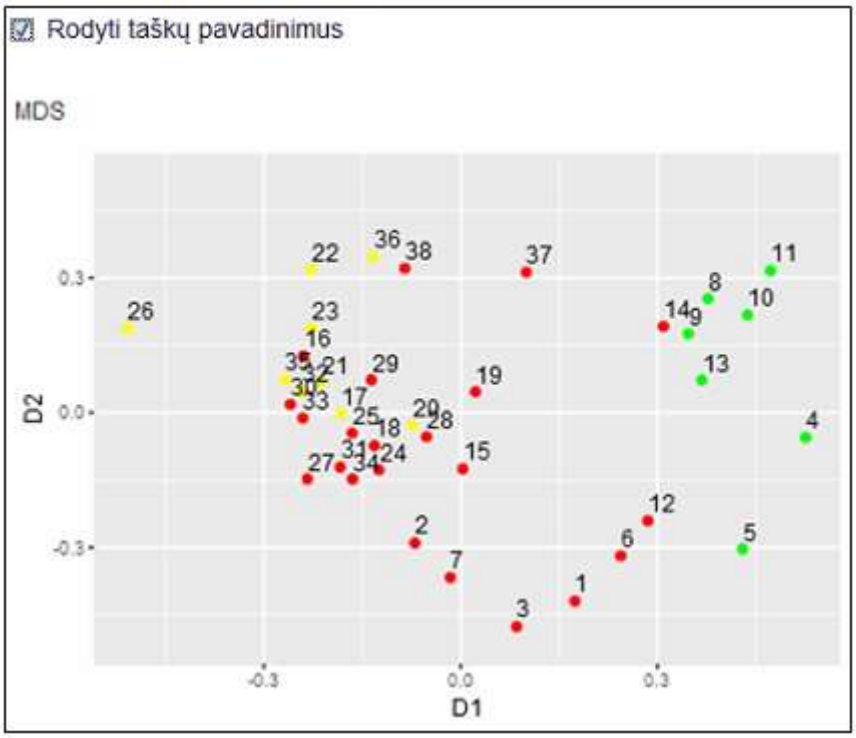
	MDS	PCA	ICA	Principal_Curves	LLE	Isomap
Stress	0.01	0.08	4.63	0.54	3.03	0.12
Spearman	0.99	0.98	0.86	0.86	0.42	0.96
Entropy	0.06	0.08	0.02	0.13	0.45	0.08

58 pav. Metodų tikslumo rodikliai



59 pav. Metodų tikslumo rodikliai grafiniame pavidale

Galima pasirinkti visuose grafikuose matyti kiekvieno objekto ID (pavadinimą). Tai leidžia lengviau surasti, kokiam klasteriui skirtinguose grafikuose (gautuose pritaikius skirtingus metodus) priklauso konkretus objektas (60 pav.). Duomenis atvaizdavus dvimačiame grafike, galima stebėti objektų tarpusavio ryšius bei priklausomybę konkretiems klasteriams. Deja, tokiame grafike nėra galimybės matyti kokius parametrus (pradinių dimensijų reikšmes) turi analizuojamas objektas. Todėl atskiroje lentelėje yra pateikiamas pasirinktų objektų sąrašas su jų parametrų reikšmėmis



60 pav. Atvaizduotų duomenų objektų ID

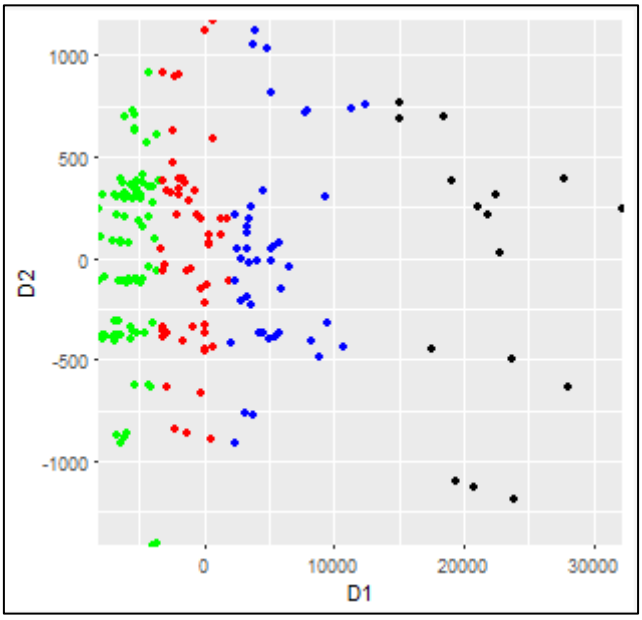
4.2.2 Antro duomenų rinkinio analizė

4.2.2.1 Duomenų rinkinio aprašymas

Šiuo atveju panaudoti duomenys apie automobilius [65]. Rinkinyje yra 205 objektai, kiekvieną jų aprašo 26 atributai: markė, kuro tipas, kėbulo tipas, ilgis, plotis, svoris, variklio darbo tūris ir kiti. Taip pat kiekvienam objektui pateikiamas draudimo rizikos reitingas ir draudimo išmokos dydis palyginus su kitais automobiliais.

4.2.2.2 Duomenų vizualizavimas ir analizė

Pradžioje visi duomenys automatiškai suklasterizuojami į 4 klasterius panaudojant k-means metodą. Pritaikius PCA metodą (Stress: 0,0000176, Spirmeno koeficientas: 0,9998649, Šenono entropija: 0,0040321) gautas toks vaizdas:



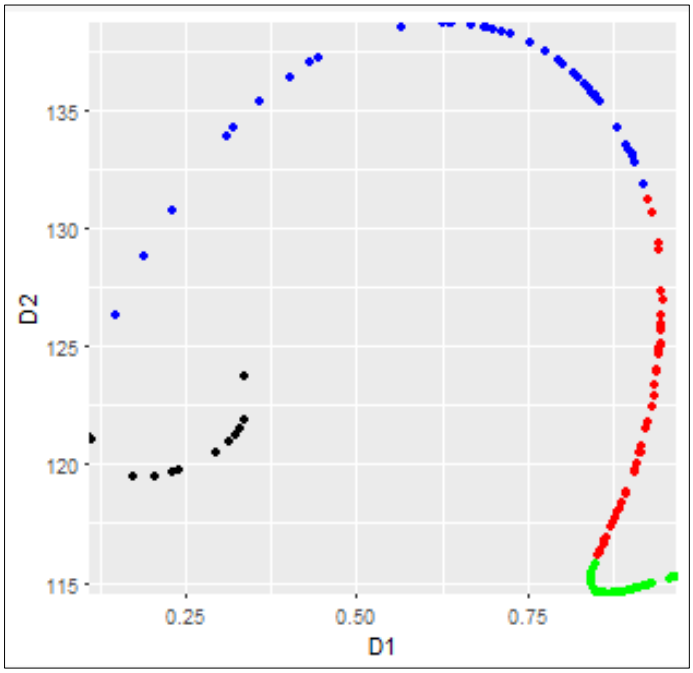
61 pav. Automobilių duomenų vizualizavimas PCA metodu

Aiškiai atsiskyrė visi 4 klasteriai. Į juos pateko atitinkamai 16, 53, 45 ir 91 objektai. Į juodai pažymėtą klasterį pateko objektai, kurių ID: 16, 17, 18, 48, 49, 50, 69, 70, 71, 72, 73, 74, 127, 128, 129, 130. Atlikus pirminę šio klasterio analizę paaiškėjo, kad į jį pateko 7 iš 9 duomenų rinkinyje esančių Mercedes-Benz automobilių, visi 3 Jaguar, visi 4 Porsche, 3 BMW automobiliai.

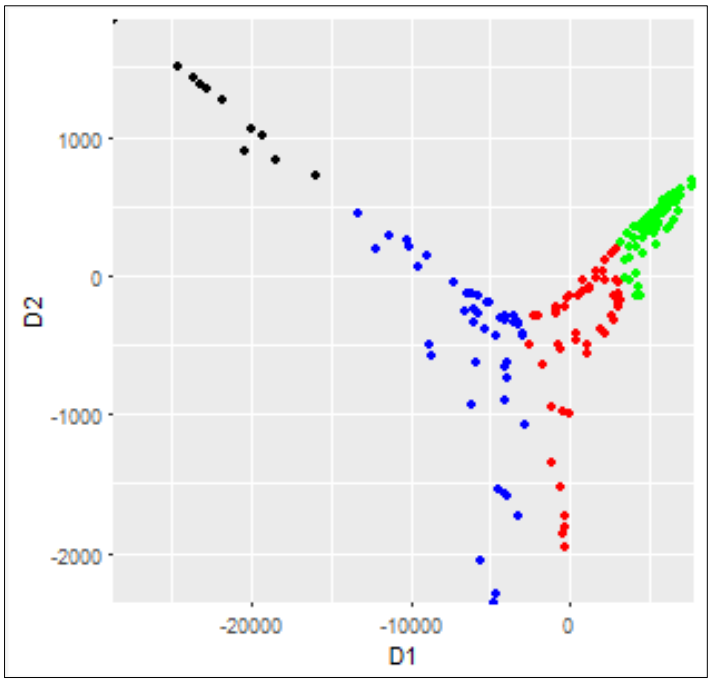
Toliau galima analizuoti, kuo būtent išsiskiria iš kitų, o tarpusavyje yra susiję, šių markių automobiliai. Išanalizavus pateikiamus statistinius rodiklius matosi, kad šio klasterio automobiliai pasižymi 53,21% didesniu variklio darbo tūriu, 37,51% didesniu cilindrų skaičiumi, 43,97% didesne arklio galia ir 92,08% didesne kaina. Įdomu tai,

kad šios grupės automobiliams priskirtas mažiausias rizikos laipsnis (21,21% mažesnis nei vidutinis).

Toliau pateikiami vizualizavimo pavyzdžiai pritaikius pagrindinių kreivių (Stress: 0,9985326, Spirmeno koeficientas: 0,7015576, Šenono entropija: -0,0098832) ir Isomap (Stress: 0,0020828, Spirmeno koeficientas: 0,9976744, Šenono entropija: 0,0038473) metodus.

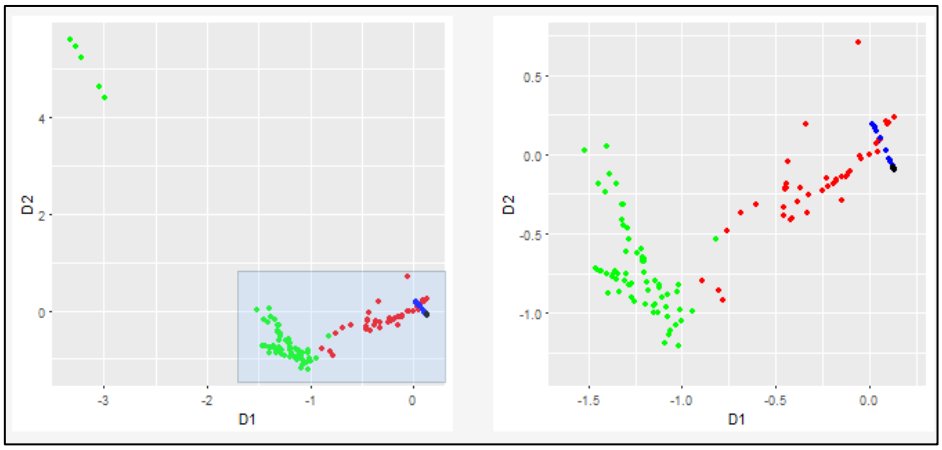


62 pav. Automobilių duomenų vizualizavimas Pagrindinių kreivių metodu



63 pav. Automobilių duomenų vizualizavimas Isomap (k=10) metodu

64 paveiksle pateikiamas duomenų vizualizavimo LLE (Stress: 0,9996529, Spirmeno koeficientas: 0,8715779, Šenono entropija: 0,0003375) metodu pavyzdys. Šiuo atveju dalis taškų yra smarkiai atsiskyrusi, todėl praverčia tam tikros srities priartinimo funkcija.

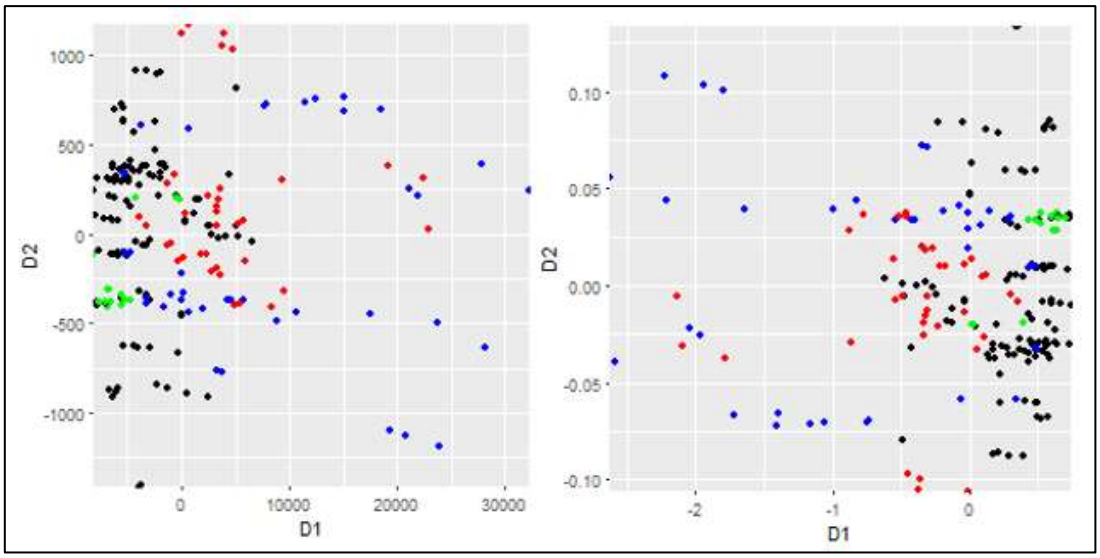


64 pav. Automobilių duomenų vizualizavimas LLE metodu

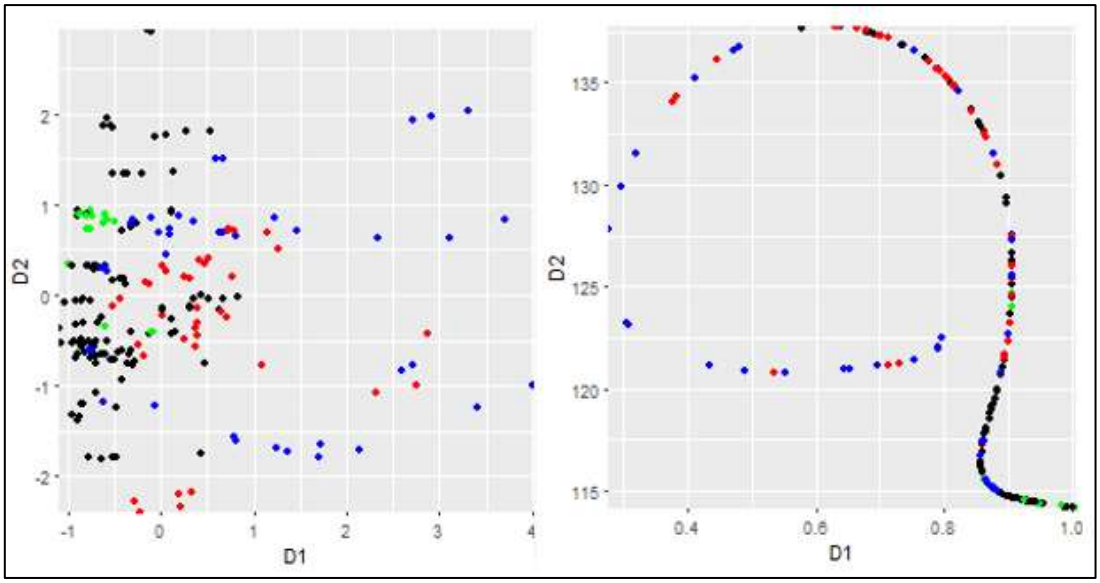
Vertinant vizualiai, visuose grafikuose (61, 62, 63, 64 pav.) matomi aiškiai atsiskiriantys 4 klasteriai. Tačiau vertinant tikslumo matus, PCA ir Isomap metodai davė tikslesnius rezultatus.

Kitame žingsnyje analizei panaudoti jau iš anksto suklasterizuoti automobilių duomenys. Automobiliai buvo suskirstyti pagal automobilių markės kilmę į 4 grupes: JAV, Vokietija, Azijos šalys, kita (Italija, Prancūzija ir t.t).

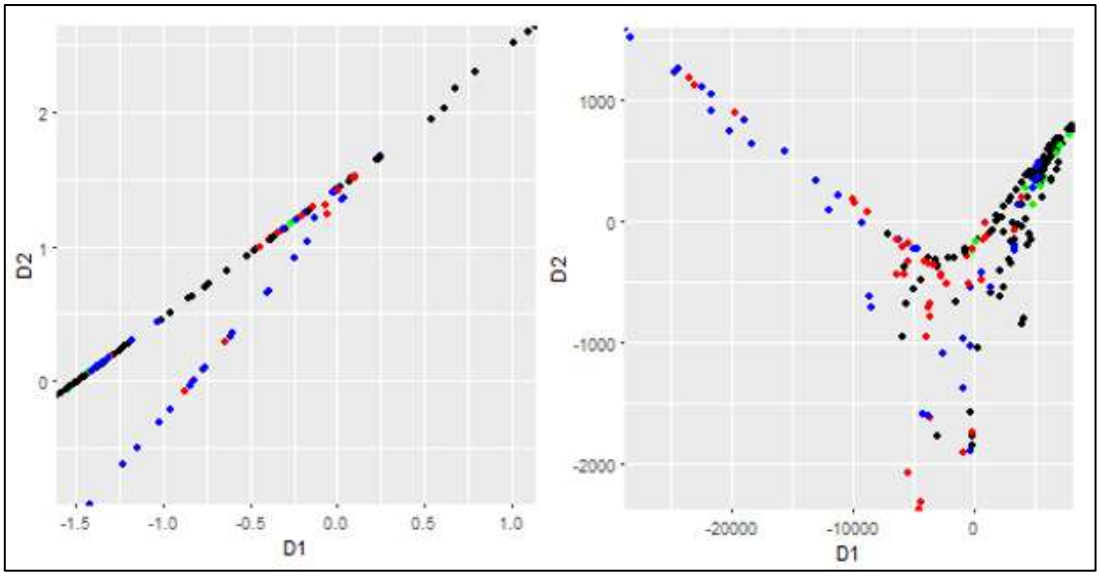
Toliau pateikiami vizualizavimo pavyzdžiai pritaikius skirtingus metodus:



65 pav. Klasterizuotų automobilių duomenų vizualizavimas PCA (kairėje) ir MDS (dešinėje) metodais.

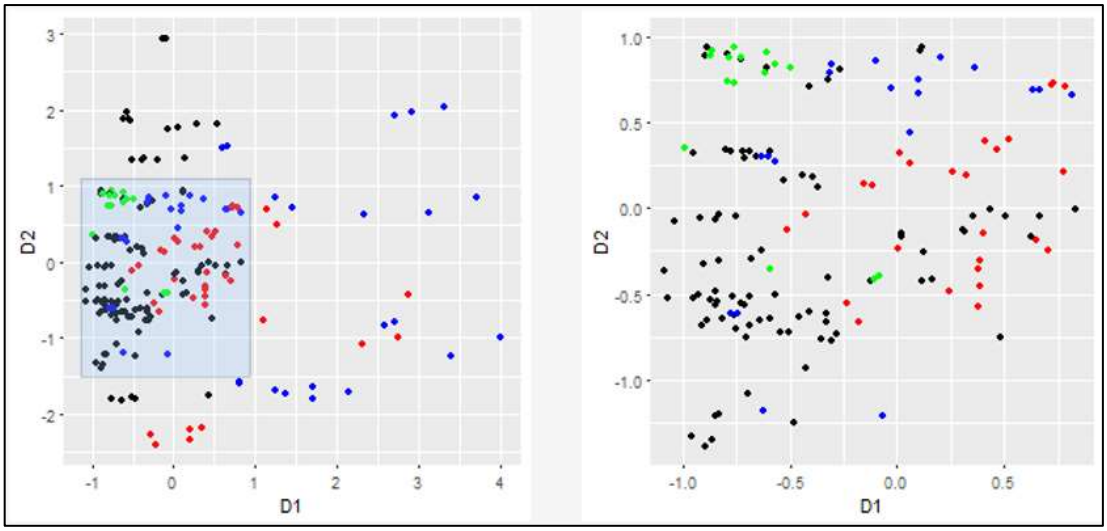


66 pav. Klasterizuotų automobilių duomenų vizualizavimas ICA (kairėje) ir Pagrindinių kreivių (dešinėje) metodais.

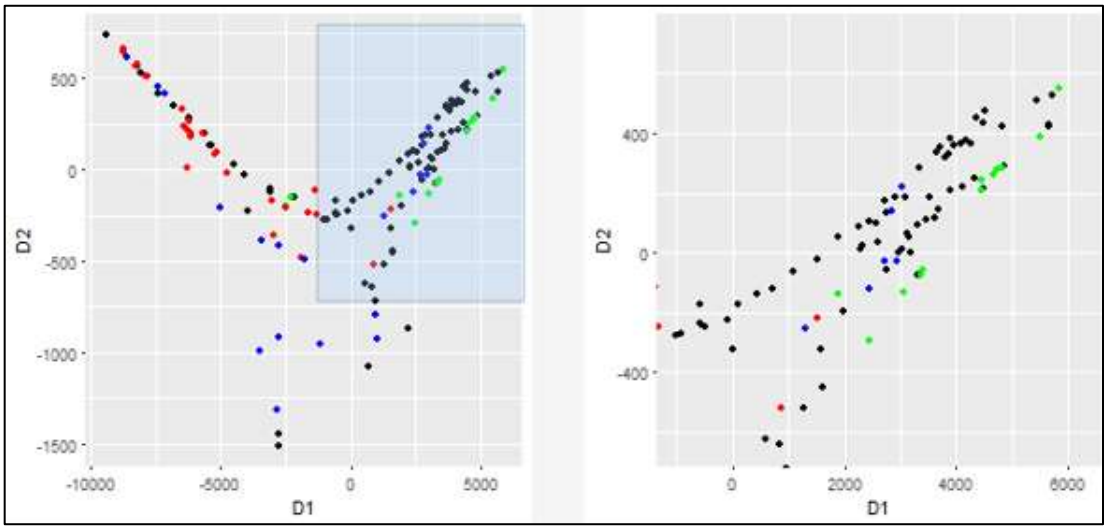


67 pav. Klasterizuotų automobilių duomenų vizualizavimas LLE (kairėje) ir Isomap (k=10) (dešinėje) metodais.

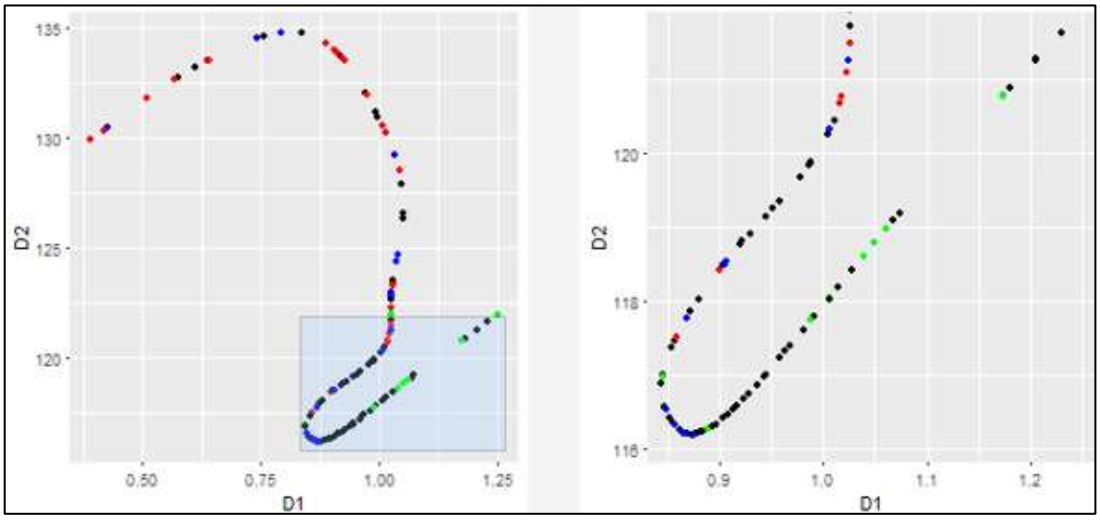
Kituose paveiksluose pavaizduotos vaizdo priartinimo galimybės.



68 pav. ICA metodu vizualizuotų duomenų srities priartinimas



69 pav. Isomap (k=10) metodu vizualizuotų duomenų srities priartinimas



70 pav. Pagrindinių kreivių metodu vizualizuotų duomenų srities priartinimas

4.2.3 Trečio duomenų rinkinio analizė

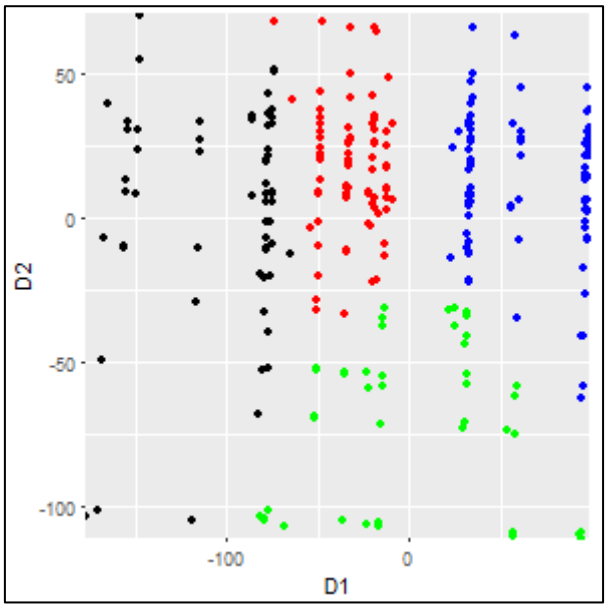
4.2.3.1 Duomenų rinkinio aprašymas

Šiuo atveju panaudotas duomenų rinkinys apie darbuotojų nebuvimo darbe priežastis [63]. Duomenys rinkti vienoje Brazilijos kurjerių įmonėje nuo 2007 liepos iki 2010 liepos. Rinkinyje yra 740 duomenų objektų. Kiekvieną jų nusako 21 atributas: nebuvimo darbe priežastis, mėnuo, savaitės diena, metų laikas, atstumas nuo gyvenamosios vietos iki darbo, amžius, alkoholio ir cigarečių vartojimas, disciplinos pažeidimai, vaikų skaičius, augintinių skaičius, KMI, nebuvimo darbe laikas ir kiti.

4.2.3.2 Duomenų vizualizavimas ir analizė

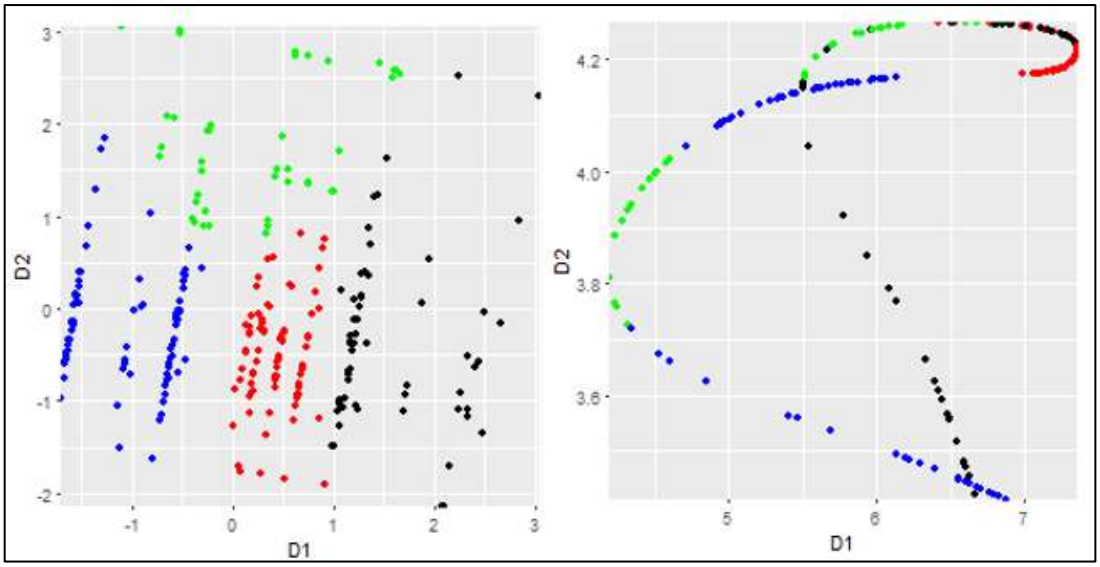
Pradžioje panaudojant *k-means* metodą visi duomenys suklasterizuojami į 4 klasterius, o po to vizualizuojami pritaikius PCA metodą (Stress: 0.0112829, Spirmeno koeficientas: 0.9849337, Šenono entropija: 0.0447027).

Kadangi šiuo atveju yra daug kategorinių duomenų, vaizdas paveiksle yra fragmentiškas.



71 pav. Darbuotojų duomenų vizualizavimas PCA metodu.

Toliau pateikiami vizualizavimo pavyzdžiai pritaikius ICA (Stress: 0,9648462, Spirmeno koeficientas: 0,937401, Šenono entropija: 0,03308564) ir Pagrindinių kreivių (Stress: 0,9815877, Spirmeno koeficientas: 0,3860661, Šenono entropija: 0,1190149) dimensijų mažinimo metodus.



72 pav. Darbuotojų duomenų vizualizavimas ICA (kairėje) ir Pagrindinių kreivių (dešinėje) metodais.

PCA ir ir ICA metodai davė tikslesnius rezultatus nei Pagrindinių kreivių metodas.

Atlikus statistinių rodiklių analizę, pastebėti skirtumai tarp klasterių.

1 klasteris: 89,47% didesnis disciplinos pažeidimų kiekis, 37,7% daugiau vartojančių alkoholį, 109,52% daugiau rūkančiųjų, išlaidos transportui didesnės 36,47%, turima 55,62% daugiau gyvūnų, turima 22,37% daugiau vaikų, darbe nebūta 42,48% daugiau.

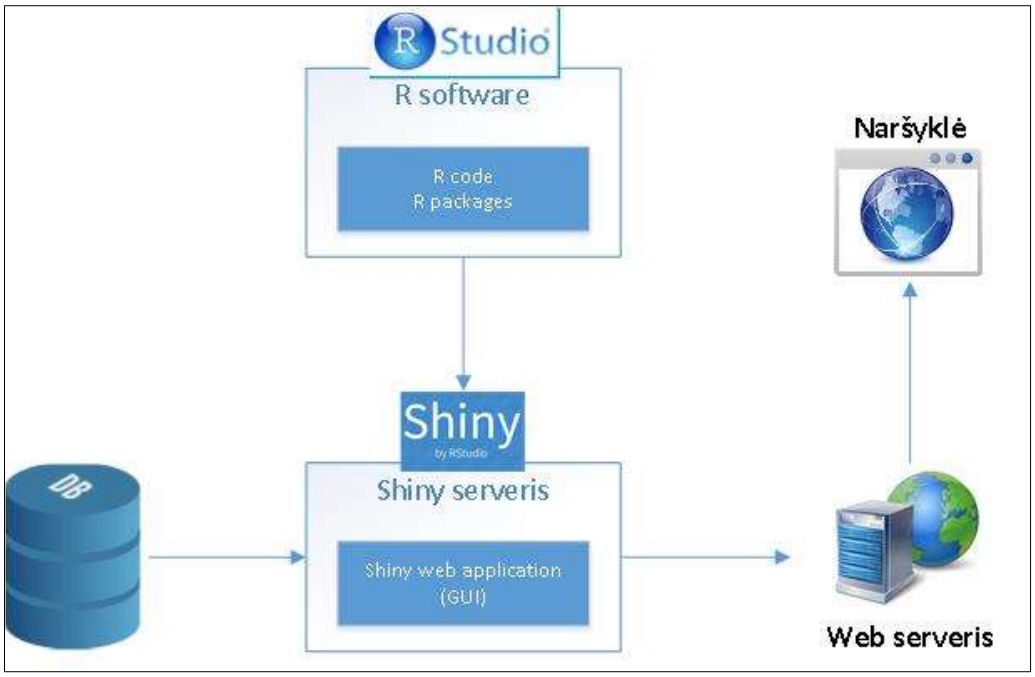
2 klasteris: 18,03% mažiau vartojančių alkoholį, 42,86% mažiau rūkančiųjų, turima 51,6% daugiau vaikų, 36,84% didesnis disciplinos pažeidimų kiekis, tačiau darbe nebūta 45,49% mažiau.

3 klasteris: išlaidos transportui mažesnės 31,21%, 57,89% mažesnis disciplinos pažeidimų kiekis, turima 76,26% mažiau vaikų, turima 91,49% mažiau gyvūnų, darbe nebūta 15,31% mažiau.

4 klasteris: 68,42% didesnis disciplinos pažeidimų kiekis, 22,95% mažiau vartojančių alkoholį, 23,81% mažiau rūkančiųjų, darbo krūvis didesnis 19,43%, darbe nebūta 42,48% daugiau.

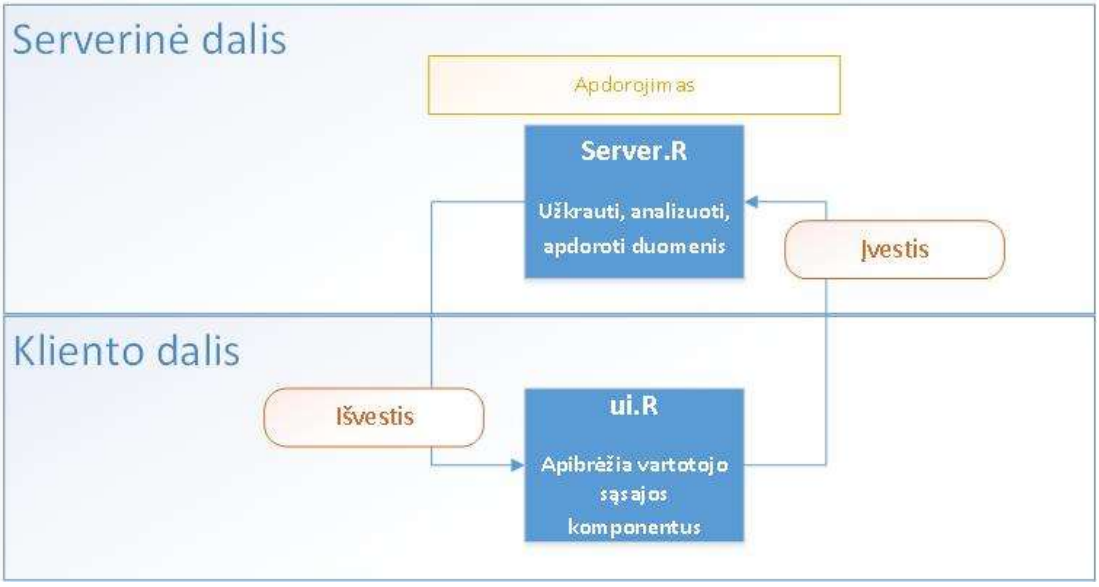
4.3 Įrankio techninis aprašymas

Įrankio programinis kodas parašytas R kalba „R studio“ aplinkoje. Interaktyvios vartotojo sąsajos sukūrimui panaudotas Shiny paketas, todėl įrankis veikia kaip web programėlė [89].



73 pav. Įrankio architektūra

Visas programinis kodas padalintas į dvi pagrindines dalis (failus) – serverio (Server.R) ir kliento (ui.R) (7.2 priedas).



74 pav. Architektūros kliento/serverio dalys

Duomenų vizualizavimo strategiją realizuojantis kodas (duomenų užkrovimas, analizė, dimensijų mažinimas, duomenų vizualizavimas grafikuose) yra serverio dalyje. Dimensijų mažinimui ir statistiniam duomenų apdorojimui panaudoti jau egzistuojantys R paketai (7.1 priedas).

Ši R kodo dalis gali būti lengvai pritaikyta lygiagretiems skaičiavimams panaudojant MPI/CUDA ir skaičiavimus vykdant debesyje (pvz. MS Azure/AWS).

Vartotojo sąsaja (grafikai, mygtukai, pasirenkamieji sąrašai ir kiti komponentai) suprogramuota ui.R dalyje. Kliento pusėje įvesti duomenys pateikiami į serverio pusę, ten yra apdorojami, o gauti rezultatai grąžinami ir išvedami vėl kliento pusėje.

4.4 Skyriaus apibendrinimas

Šiame skyriuje aprašyta daugiapakopio duomenų vizualizavimo metodologija ir ją realizuojantis interaktyvus įrankis. Metodologijos ypatybės ir įrankio galimybės pristatytos aprašant realių duomenų vizualizavimo pavyzdžius.

Pasiūlyta metodologija ir įrankis suteikia daugiau galimybių vizualizuojant didžiuosius duomenis. Ji leidžia duomenis analizuoti įvairiais pjūviais, duomenis analizuoti pažingsniui, kiekviename etape pasirenkant dominančią duomenų aibę ir jai pritaikant geriausiai tinkantį dimensijų mažinimo metodą. Metodų pasirinkimą palengvina pateikiami grafiniai vizualizavimo pavyzdžiai ir tikslumo rodikliai.

5 Bendrosios išvados

Šiuo tyrimu siekta pasiūlyti integralią didelės apimties duomenų vizualios analizės metodologiją, apimančią skirtingų dimensijų mažinimo metodų taikymą, jų statistinių savybių panaudojimą metodų parinkimui, duomenų paruošimą analizei bei jų vizualizavimą. Siekiant išsikelto tikslo darbe buvo gauti šie rezultatai:

1. Apžvelgta didžiųjų duomenų analizės problematika, išanalizuoti duomenų vizualizavimui naudojami projekcijos ir klasterizavimo metodai. Buvo išnagrinėti vizualizavimo metodų, pagrįstų dimensijų mažinimu, tikslumo įvertinimo matai; atlikti metodų greičio ir tikslumo vertinimo tyrimai. Gauti rezultatai patvirtino prielaidą, kad priklausomai nuo duomenų pobūdžio, jų apimties, analizės tikslu gali reikėti naudoti skirtingus metodus. Be pradinės analizės negalima iš anksto pasakyti, koks metodas duos geriausius rezultatus konkrečiu atveju. Todėl reikalinga metodologija, kuri leistų kiekviename analizės etape įvertinti situaciją ir parinkti tinkamą dimensijų mažinimo metodą.
2. Atlikta lyginamoji komercinių ir mokslinėje literatūroje pristatomų duomenų vizualizavimo įrankių analizė parodė, kad egzistuojantys įrankiai išsprendžia tik atskiras su duomenų vizualizavimu susijusias problemas ir turi apribojimų, pvz.: nėra galimybės tolesnei analizei pasirinkti norimas pradinių duomenų dalis ir kiekviename žingsnyje taikyti labiausiai tinkantį dimensijų mažinimo metodą; dimensijų mažinimo metodas yra pritaikomas tik kartą pradiniam duomenų rinkiniui, o vėliau vaizdą galima tik priartinti, be galimybės pasirinktam klasteriui iš naujo pritaikyti kitus dimensijų mažinimo metodus; trūksta interaktyvumo, negalima tiesiogiai priartinti vaizdo ar keisti dimensijų mažinimo metodų; nepateikiamas pradinio duomenų rinkinio vizualizavimas.
3. Sukurta metodologija suteikia daugiau galimybių vizualizuojant didžiuosius duomenis. Ji leidžia duomenis analizuoti įvairiais pjūviais, duomenis analizuoti pažingsniui, kiekviename etape pasirenkant dominančią duomenų aibę ir jai pritaikant geriausiai tinkantį dimensijų mažinimo metodą, atsižvelgiant į dimensijų mažinimo greitį ir tikslumą konkrečiu atveju. Metodų pasirinkimą palengvina pateikiami grafiniai vizualizavimo pavyzdžiai bei greičio ir tikslumo rodikliai.
4. Didžiųjų duomenų vizualizavimui labai svarbus aspektas yra duomenų apdorojimo greitis. Keltas tikslas, kad siūloma metodologija būtų teisinga ne tik

teoriniame lygmenyje, bet taip pat galėtų efektyviai veikti apdorojant realius didelės apimties duomenis. Darbe išanalizuota, kaip lygiagrečiųjų skaičiavimų taikymas gali paspartinti duomenų dimensijų mažinimo bei vizualizavimo užduotis. Pasiūlyta tokia metodologija ir ją realizuojančio įrankio architektūra, kad sukurtas sprendimas leistų panaudoti paskirstytų sistemų išteklius ir debesų kompiuterijos pranašumus. Įrankio prototipo galimybės pristatytos aprašant realių duomenų vizualizavimo atvejus.

6 Literatūra

[1]Ankerst, M., Breunig, M. M., Kriegel, H., Sander, J. OPTICS: Ordering Points To Identify the Clustering Structure. *ACM SIGMOD'99 Int. Conf. on Management of Data*, Philadelphia PA, 1999. http://www.dbs.ifi.lmu.de/Publikationen/Papers/OPTICS.pdf (last accessed 18 March 2018).

[2]Aupetit, M. Sanity check for class-coloring-based evaluation of dimension reduction techniques. *Proceeding BELIV '14 Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization.* Pages 134-141.

[3] Bernatavičienė, J. Vizualios žinių gavybos metodologija ir jos tyrimas. Daktaro disertacija, Vilniaus Gedimino technikos universitetas, Matematikos ir informatikos institutas, 2008.

[4]Bikakis, N. Big Data Visualization Tools. Publication: eprint arXiv:1801.08336, 2018. https://arxiv.org/abs/1801.08336 (last accessed 18 March 2018).

[5]Borgman, Christine, L. Big data, little data, or no data? Scholarship and stewardship to build the UC digital library. *Keynote Presentation. Inaugural 2018 University of California Digital Library Forum (UC DLFx)*, 2018.

[6] Bramer, M. Principles of Data Mining, *Springer,* 2007.

[7]Braun, L., Volke, M., Schlamp, J., Bodisco, A., Carle, G. Flow-inspector: a framework for visualizing network flow data using current web technologies. *Computing*, vol 96, 2014. Pages 15–26.

[8]Cappello, F., Etiemble, D. MPI versus MPI+OpenMP on the IBM SP for the NAS Benchmarks. Supercomputing, ACM/IEEE 2000 Conference. 2000. ISSN: 1063-9535

[9]Chikhi, N. F., Rothenburger, B., Aussenac-Gilles, N. A Comparison of Dimensionality Reduction Techniques for Web Structure Mining. *IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)*, Fremont, CA, 2007. Pages 116-119.

[10] Chorleya, M. J., Walkera, D. W. Performance Analysis of a Hybrid MPI/OpenMP Application on Multi-core Clusters. Journal of Computational Science Volume 1, Issue 3, August 2010, Pages 168–174.

[11] Chudzij, L; Treigys, P. (2014). Saityno paslaugomis grindžiamas daugiamačių duomenų analizės įrankis. *Mokslo taikomųjų tyrimų įtaka šiuolaikinių studijų*

kokybei. VII respublikinės mokslinės-praktinės konferencijos mokslinių straipsnių leidinys. Vilnius: Vilniaus kolegija, p. 23-30.

[12] Claveria, O., Poluzzi, A. Positioning and clustering of the world's top tourist destinations by means of dimensionality reduction techniques for categorical data. *Elsevire, Journal of Destination Marketing & Management*, vol 6, 2017. Pages 22-32.

[13] Cutura, R., Holzer, S., Aupetit, M., Sedlmair, M. VisCoDeR: A Tool for Visually Comparing Dimensionality Reduction Algorithms. https://homepage.univie.ac.at/michael.sedlmair/papers/cutura2018viscoder.pdf (last accessed 18 March 2018).

[14] Čekanavičius, V., Murauskas, G. Statistika ir jos taikymai, II dalis. (2002). Vilnius: TEV. 268 p. ISBN 9955-491-16-7.

[15] Datanovia. Partitional clustering in R: the essentials. https://www.datanovia.com/en/lessons/clara-in-r-clustering-large-applications/ (last accessed 18 March 2018).

[16] Deepak, S., T; Sultana, N. S. (2015). Big data analysis using hace theorem. *International Journal of Advanced Research in Computer Engineering and Technology* 4(1): 18-23.

[17] Diamond, M., Mattia, A. Data Visualization: An Exploratory Study into the Software Tools Used by Businesses. *Journal of Instructional Pedagogies*, vol. 18, 2017. https://eric.ed.gov/?id=EJ1151731 (last accessed 18 March 2018).

[18] Doerr, S., Ariz-Extreme, I., Harvey, M. J., Fabritiis, G. Dimensionality reduction methods for molecular simulations. https://arxiv.org/abs/1710.10629 (last accessed 18 March 2018).

[19] Domeniconi. Comparison of Principal Component Analysis and Random Projection in Text Mining. INFS 795, (2004)

[20] Dumouchel, W. (2002). Data Squashing: Contructing Summary Data Dets. *Handbook of Massive Data Sets*: 579-591. ISBN: 978-1-4613-4882-5

[21] Dunham, M., H. Data Mining. Introductory and Advanced Topics. *Prentice Hall*, ISBN 0-13-088892-3.

[22] Dzemyda, G., Kurasova O., Žilinskas, J. Multidimensional data visualization: methods and applications, 1 ed., New York: Springer, 2012.

[23] Dzemyda, G., Kurasova, O., Marcinkevičiu,s V., Medvedev, V. "Efficient Data Projection for Visual Analysis of Large Data Sets Using Neural Networks," Informatica, vol. 2, no. 4, pp. 507–520, 2011.

[24] Dzemyda, G., Medvedev, V., Lupeikienė, A., Kurasova O., Čaplinskas, A. "Big xMultidimensional Datasets Visualization Using Neural Networks – Efficient Decision Support," Complex Systems Informatics and Modeling Quarterly, vol. 6, pp. 1–11, 2016.

[25] Dzemyda, G.; Kurasova, O.; Žilinskas, J. (2008). Daugiamačių duomenų vizualizavimo metodai. Vilnius: Mokslo aidai.

[26] Eka Sugiyarti et al. (2018). Decision support system of scholarship grantee selection using data mining. *International Journal of Pure and Applied Mathematics*, Volume 119 No. 15 2018, 2239-2249; ISSN: 1314-3395.

[27] Fan, W.; Bifet A. (2012). Mining Big Data: Current Status, and Forecast to the Future. *ACM SIGKDD Explorations Newsletter* 14(2): 1-5.

[28] Fodor, I. K. A survey of dimension reduction techniques. *Center for Applied Scientific Computing*, Lawrence Livermore National Laboratory, 2002.

[29] Fodor, I. K. A survey of dimension reduction techniques. *Center for Applied Scientific Computing,* Lawrence Livermore National Laboratory. June 2002

[30] Frantz, T., L. Blockmap: an interactive visualization tool for big-data networks. *Computational and Mathematical Organization Theory*, vol 24, 2018. Pages 149–168.

[31] Galletta, A., Carnevale, L., Bramanti, A., Fazio, M. An Innovative Methodology for Big Data Visualization for Telemedicine. *IEEE Transactions on Industrial Informatics*, vol 15, 2019. Page(s): 490 – 497.

[32] Gleicher, M., Albers, D., Walker, R., Jusufi, I., Hansen, Ch. D., Roberts, J. C. Visual comparison for information visualization. *Information Visualization*, vol 10(4), 2011. Pages 289–309.

[33] Gui, J., Ji, X., Amos, Ch., I. Efficient survival multifactor dimensionality reduction method for genome-wide association study. *American Association for Cancer Research*, Vol 78, 2018.

[34] Harandi, M., Salzmann, M., Hartley, R. Dimensionality Reduction on SPD Manifolds: The Emergence of Geometry-Aware Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 40 , 2018 . Page(s): 48 – 62.

[35] Hassan, A, Elragal, A. Big Data Visualization Tool: a Best-Practice Selection Model. *Institute of Electrical and Electronics Engineers (IEEE)*, 2017. p. 59-68. http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1072292&dswid=6232 (last accessed 18 March 2018).

[36] Hastie, T., Tibshirani, R., Friedman, J. The Elements of Statistical Learning Data Mining, Inference, and Prediction. *Springer*, Stanford, California, 2008. https://web.stanford.edu/~hastie/ElemStatLearn//printings/ESLII_print10.pdf (last accessed 18 March 2018).

[37] Hauke, J., Kossowski, J. Comparison of values of Pearson's and Spearman's correlation coefficients on the same sets of data. *Quaestiones geographicae*, 30(2), p. 87-93, 2011.

[38] Hausser, J., Strimmer, K. Entropy inference and the James-Stein estimator, with application to nonlinear gene association networks. *Journal of Machine Learning Research*, 2009, 10, 1469-1484.

[39] Hausser, J., Strimmer, K. Entropy inference and the James-Stein estimator, with application to nonlinear gene association networks. *Journal of Machine Learning Research*, 2009, 10, 1469-1484.

[40] Hou, T., Lin, F., Bai, S., Cleves, M.,A., Xu, H., Lou, X. Generalized multifactor dimensionality reduction approaches to identification of genetic interactions underlying ordinal traits. *Genetic Epidemiology*, vol 43, 2019. Pages 24-36.

[41] Huang, T; Lan, L; Fang, X; An, P; Min, J; Wang, F. (2015). Promises and Challenges of Big Data Computing in Health Science. *Big Data Research*: 2(1): 2-11.

[42] Ingram, S., Munzner, T., Irvine, V., Tory, M., Bergner, S., Moller T. DimStiller: Workflows for dimensional analysis and reduction. *2010 IEEE Symposium on Visual Analytics Science and Technology*, Salt Lake City, UT, 2010. Pages 3-10.

[43] Intel Parallel studio. Access: https://software.intel.com/en-us/intel-parallel-studio-xe

[44] Yang, C., Chuang, L., Lin, Y. Multiobjective multifactor dimensionality reduction to detect SNP–SNP interactions. *Bioinformatics*, vol 34, 2018. Pages 2228–2236.

[45] Yongjie, L., Zheng, W., Yang, H. A Hierarchical Visualization Analysis Model of Power Big Data. *IOP Conference Series: Earth and Environmental Science*, 2018,

vol. 108. http://iopscience.iop.org/article/10.1088/1755-1315/108/5/052064/meta (last accessed 18 March 2018).

[46] Jagadish, H. V. (2015). Big Data and Science: Myths and Reality. *Big Data Research* 2(2): 49-52.

[47] Jakimauskas, G. Duomenų tyrybos empirinių Bajeso metodų tyrimas ir taikymas. Daktaro disertacija [interaktyvus]. Prieiga per internetą: <http://www.mii.lt/files/mii_dis_2014_jakimauskas.pdf>

[48] Jin, H., Jespersen, D., etc. High performance computing using MPI and OpenMP on multi-core parallel systems. Parallel Computing 37 (2011) 562–575

[49] Jin, X; Wah, B. W; Cheng, X; Wang, Y. (2015). Significance and Challenges of Big Data Research. *Big Data Research* 2(2): 59-64.

[50] Kammer, D., Keck, M., Gründer, T. Exploring Big Data Landscapes with a Glyph-based Zoomable User Interface. In: Dachselt, R. & Weber, G. (Hrsg.), *Mensch und Computer 2018 - Workshopband. Bonn: Gesellschaft für Informatik* e.V, 2018. https://dl.gi.de/handle/20.500.12116/16838 (last accessed 18 March 2018).

[51] Kammer, D., Keck, M., Gründer, T., Groh, R. Big data landscapes: improving the visualization of machine learning-based clustering algorithms. *Proceeding AVI '18 Proceedings of the 2018 International Conference on Advanced Visual Interfaces Article,* No. 66, 2018.

[52] Kammer, D., Keck, M., Muller, M., Grunder, T., Groh, R.. Exploring Big Data Landscapes with Elastic Displays. *Mensch und Computer 2017 - Workshopband–Spielend einfach interagieren*, 2017. Pages: 381 – 387.

[53] Karbauskaitė, R. Daugiamačių duomenų vizualizavimo metodų, išlaikančių lokalią struktūrą, analizė. Daktaro disertacija, Vytauto Didžiojo univesitetas, Matematikos ir informatikos institutas, 2010.

[54] Kaur, D., Aujla, G. S., Kumar, N., Zomaya, A., Y., Perera, Ch., Ranjan, R. Tensor-Based Big Data Management Scheme for Dimensionality Reduction Problem in Smart Grid Systems: SDN Perspective. *IEEE Transactions on Knowledge and Data Engineering*, vol 30, 2018. Pages: 1985 – 1998.

[55] Khomtchouk, B. B, Hennessy, J. R, Wahlestedt C. Shinyheatmap: Ultra fast low memory heatmap web interface for big data genomics. *PLoS ONE* 2017, 12(5): e0176334. https://doi.org/10.1371/journal.pone.0176334 (last accessed 18 March 2018).

[56] Kim, H., Howland, P., Park, H. Dimension Reduction in Text Classification with Support Vector Machines. *Journal of Machine Learning Research*, vol. 6, p. 37–53. (2005).

[57] Krause, J., Dasgupta, A., Fekete, J., Bertini, E. SeekAView: An Intelligent Dimensionality Reduction Strategy for Navigating High-Dimensional Data Spaces. *LDAV 2016 - IEEE 6th Symposium on Large Data Analysis and Visualization* 2016, Baltimore, MD, United States.

[58] Kuang, L., Yang, L. T., Chen, J., Hao, F., Luo Ch. A Holistic Approach for Distributed Dimensionality Reduction of Big Data. *IEEE Transactions on Cloud Computing*, vol 6, 2018. Page(s): 506 – 518.

[59] Kudyba, S. Big Data, Mining, and Analytics–Components of Strategic Decision Making. *CRC Press Taylor & Francis Group an Auerbach Book.* ISBN 9781466568709. (2014).

[60] Kurasova, O. (2014). Duomenų tyrybos strategijos. *Metodinė priemonė*. Vilnius, 2015

[61] Kurasova, O., Marcinkevičius, V., Medvedev, V., Rapečka, A., Stefanovič, P. (2014). Strategies for big data clustering. *Proceedings of IEEE 26th International Conference on Tools with Artificial Intelligence, ICTAI 2014*, p. 740-747.

[62] Lai, Z., Xu, Y., Yang, J., Shen, L., Zhang, D. Rotational Invariant Dimensionality Reduction Algorithms. *IEEE Transactions on Cybernetics*, vol 47 , 2017. Page(s): 3733 – 3746.

[63] Machine Learning Repository. Absenteeism at work Data Set Description. https://archive.ics.uci.edu/ml/datasets/Absenteeism+at+work (last accessed 18 March 2018).

[64] Machine Learning Repository. Anuran Calls Data Set Description. https://archive.ics.uci.edu/ml/datasets/Anuran+Calls+%28MFCCs%29 (last accessed 18 March 2018).

[65] Machine Learning Repository. Automobile Data Set Description. https://archive.ics.uci.edu/ml/datasets/Automobile (last accessed 18 March 2018).

[66] Mallon, D. A., Taboada, G. L., etc. Performance Evaluation of MPI, UPC and OpenMP on Multicore Architectures. Parallel and Distributed Processing Symposium, Proceedings, 2004.

[67] Marcinkevičius, V. Netiesinės daugiamačių duomenų projekcijos metodų savybių tyrimas ir funkcionalumo gerinimas. Daktaro disertacija, Vytauto Didžiojo universitetas, Matematikos ir informatikos institutas, 2010.

[68] Martišiūtė D. Vaizdų klasterizavimas. Vilniaus universitetas, matematikos ir informatikos fakultetas. Vilnius, 2009.

[69] Medvedev, V. Tiesioginio sklidimo neuroninių tinklų taikymo daugiamačiams duomenims vizualizuoti tyrimai. Daktaro disertacija, Vilniaus Gedimino technikos universitetas, Matematikos ir informatikos institutas, 2008.

[70] Menon, A. K. Random projections and applications to dimensionality reduction. School of Information Technologies, The University of Sydney. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.164.640&rep=rep1&type=pdf (last accessed 2 November 2017).

[71] Message passing interface. Access: https://computing.llnl.gov/tutorials/mpi/

[72] Microsoft MPI. Access: https://msdn.microsoft.com/en-us/library/windows/desktop/bb524831(v=vs.85).aspx

[73] Mininni, P. D., Rosenberg, D., Reddy, R., Pouquet, A. A hybrid MPI-OpenMP scheme for scalable parallel pseudospectral computations for fluid Turbulence. Institute for Mathematics Applied to Geosciences National Center for Atmospheric Research. 2010. Access: arXiv:1003.4322

[74] Mittelstadt, S; Stoffel, A; Keim D. (2014). Methods for Compensating Contrast Effects in Information Visualization. *Computer Graphics Forum*, 33(3): 231-240.

[75] Mizuta, M. Dimension Reduction Methods. Humboldt-Universität Berlin, *Center for Applied Statistics and Economics (CASE)*, 2007, 15.

[76] MPI technology. Access: http://www.mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf

[77] MPICH description. Access: http://www.mpich.org/

[78] Nazemi, M., Eshratifar, A., E., Pedram, M. A Hardware-Friendly Algorithm for Scalable Training and Deployment of Dimensionality Reduction Models on FPGA. Publication: eprint arXiv:1801.04014, 2018. https://arxiv.org/abs/1801.04014 (last accessed 18 March 2018).

[79] Nilashi, M., Ibrahim, O., Bagherifard, K. A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. *Elsevier, Expert Systems with Applications*, vol 92, 2018. Pages 507-520.

[80] OpenMP compilers. Access: http://openmp.org/wp/openmp-compilers/

[81] Paakkonen, P; Pakkala, D. (2015). Reference Architecture and Classification of Technologies, Products and Services for Big Data Systems. *Big Data Research*. p. 1-21.

[82] Paulauskienė, K. Dimensijų mažinimu pagrįstas didelės apimties duomenų vizualizavimas ir projekcijos paklaidos vertinimas. Daktaro disertacija [interaktyvus]. Prieiga per internetą: <https://www.mii.lt/files/doc/lt/doktorantura/apgintos_disertacijos/dmsti_dis_2018_paulauskiene.pdf>

[83] Petrolis, R. Daugiamatės analizės metodai biomediciniinių vaizdų ir signalų analizėje ir veritinime. Daktaro disertacija, Lietuvos sveikatos mokslų universitetas, Kaunas, 2015.

[84] Qin, X., Luo, Y., Tang, N., Li, G. DeepEye: An automatic big data visualization framework. *Big Data Mining and Analytics*, vol 1, 2018. Page(s): 75 – 82.

[85] R package 'clusterGeneration' - Random Cluster Generation (with Specified Degree of Separation). 2015.

[86] R package 'entropy' - Estimation of Entropy, Mutual Information and Related Quantities.

[87] R package 'entropy' - Estimation of Entropy, Mutual Information and Related Quantities. https://cran.r-project.org/web/packages/entropy/entropy.pdf (last accessed 2 November 2017).

[88] R package 'smacof' - Multidimensional Scaling. 2017. https://cran.r-project.org/web/packages/smacof/smacof.pdf (last accessed 2 November 2017).

[89] R Shiny. https://shiny.rstudio.com/ (last accessed 18 March 2018).

[90] Rabenseifner, R., Hager, G., etc. Hybrid MPI/OpenMP Parallel Programming on Clusters of Multi-Core SMP Nodes. Conference: Proceedings of the 17th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, PDP 2009, Weimar, Germany, 18-20 Febuary 2009

[91] Ratner, B. (2012). Statistical and Machine-Learning Data Mining: Techniques for Better Predictive Modeling and Analysis of Big Data. ISBN 9781439860915.

[92] Rifat Chowdhury. Parallel Computing with OpenMP to solve matrix Multiplication. UCONN BIOGRID REU Summer 2010. Department of Computer Science & Engineering. University of Connecticut, Storrs, CT 06269

[93] Rosaria, R. S., Adae, I., Hart, A., Berthold, M. Seven Techniques for Dimensionality Reduction. Knime, 2014. https://www.knime.com/blog/seven-techniques-for-data-dimensionality-reduction (last accessed 2 November 2017).

[94] Ruan, Z., Miao, Y., Pan, L., Xiang, Y., Zhang, J. Big network traffic data visualization. *Multimedia Tools and Applications,* vol 77, 2018. Pages 11459–11487.

[95] Sacha D., Zhang L., Sedlmair M., Lee J. A., Peltonen J., Weiskopf D. Visual Interaction with Dimensionality Reduction: A Structured Literature Analysis. *IEEE Transactions on Visualization and Computer Graphics*, vol 23 , 2017. Page(s): 241 – 250.

[96] Sakalauskas, L. (2009). Duomenų gavyba. *Paskaitų konspektas*. Vilnius, 2009

[97] Santoyo, S. A Brief Overview of Outlier Detection Techniques. (2017). https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561 (last accessed 18 March 2018).

[98] Sedlmair, M., Heinzl, C., Bruckner, S., Piringer, H., Möller T. Visual Parameter Space Analysis: A Conceptual Framework. *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, 2014. Pages.2161-2170.

[99] Sherman, C. (2014). What‘s the Big Deal About Big Data? Online Searcher 38.2. *ProQuest Central*. p. 10-17.

[100] Smirnova, E., Ivanescu, A., Bai, J., Crainiceanu, C. P. A practical guide to big data. *Elsevier Statistics & Probability Letters*, vol 136, 2018, Pages 25-29.

[101] Soille, P., Burger, A., Marchi, D., Kempeneers, P., Rodriguez, D., Syrris, V., Vasilev, V. A versatile data-intensive computing platform for information retrieval from big geospatial data. *Elsevier, Future Generation Computer Systems*, vol 81, 2018. Pages 30-40.

[102] Sorzano, C. O. S., Vargas, J., Montano, A. P. A survey of dimensionality reduction techniques, 2014. https://arxiv.org/abs/1403.2877 (last accessed 2 November 2017).

[103] Stock ratios. Access: < http://finviz.com/>.

[104] Support Vector Machine – Regression [interaktyvus]. Prieiga per internetą: <http://www.saedsayad.com/support_vector_machine_reg.htm>.

[105] Tang, B., Shepherd, M., Milios, E., Heywood, M. I. Comparing and Combining Dimension Reduction Techniques for Efficient Text Clustering. *Proceedings of the Workshop on Feature Selection for Data Mining: Interfacing Machine Learning and*

Statistics in conjunction with the 2005 SIAM International Conference on Data Mining, April 23, 2005, Newport Beach, CA.

[106] Wolf, F., Mohr, B. Automatic performance analysis of hybrid MPI/OpenMP applications. Journal of Systems Architecture 49 (2003) 421–439

[107] Zhang, L., Stoffel, A., Behrisch, M. Mittelstädt, S. Schreck, T., Pompl, R., Weber, S., Last, H., Keim, D. (2012). Visual analytics for the big data era – a comparative review of state-of-the-art commercial systems. *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, p. 173-182.

[108] Zhong, X., Enke, D. Forecasting daily stock market return using dimensionality reduction. *Elsevier Expert Systems with Applications*, vol 67, 2017. Pages 126-139.

[109] Zubova, J., Kurasova, O., Liutvinavicius, M. Dimensionality Reduction Methods: The Comparison Of Speed And Accuracy. *Information Technology And Control,* vol 47, No 1 (2018).

[110] Zubova, J., Kurasova, O., Liutvinavicius, M. Parallel computing for dimensionality reduction. Information and Software Technologies, Springer-Verlag. p. 230-241, ISBN 978-3-319-46254-7. (2016).

[111] Zubova, J., Kurasova, O., Liutvinavicius, M. Parallel computing for dimensionality reduction. Information and Software Technologies, Springer-Verlag. p. 230-241, ISBN 978-3-319-46254-7. (2016).

[112] Žilinskas A., Žilinskas, J. "Parallel hybrid algorithm for global optimization of problems occurring in MDS-based visualization," Computers & Mathematics with Applications, vol. 52, no. 1, pp. 211–224, 2006.

[113] Žilinskas A., Žilinskas, J. "Two level minimization in multidimensional scaling," Journal of Global Optimization, vol. 38, no. 4, pp. 581–596, 2007.

[114] Žilinskas, A., Žilinskas, J. "Branch and bound algorithm for multidimensional scaling with city-block metric," Journal of Global Optimization, vol. 43, no. 2–3, pp. 357–372, 2009.

[115] Žilinskas, J. "Parallel branch and bound for multidimensional scaling with cityblock distances," Journal of Global Optimization, vol. 54, no. 2, pp. 261–274, 2012.

7 Priedai

7.1 R paketai

```
library('smacof')
library('ica')
library('lle')
library('RDRToolbox')
library('princurve')
library('entropy')
library(ggplot2)
library('cluster')
library('dbscan')
library("fpc")
library(shinyBS)
```

7.2 R kodo fragmentai

ui.R kodo fragmentas:

```
ui <- fluidPage(

…

#Radio buttonai metodo pasirinkimui
  radioButtons("radio", label = h4("Dimensionality
reduction methods"),
              choices = list("MDS smacof" = 1, "PCA" = 2,
"ICA" = 3, "Pricipal Curves" = 4, "LLE" = 5, "Isomap" = 6),
              selected = 1, inline=T),
       hr(),
       #----------------------------------
       #Action button vizualizavimo veiksmo patvirtinimui
        actionButton("do", "Pradinis vizualizavimas"),
       #-------------------------------------
       #Action button vizualizavimo keitimo veiksmo
patvirtinimui
       actionButton("do2", "Vizualizuoti pasirinktu
metodu"),
       #-------------------------------------

       #Action button vizualizavimui pagal visus metodus
       actionButton("do3", "Vizualizuoti visais
metodais"),


----
```

Server.R kodo fragmentas:

```
server <- function(input, output, session) {

……

observeEvent(input$do,{
  Metodai(Metodo_kint)
  Radio_kint_2 <<- input$radio

  if(Radio_kint_2 == "1"){
   #---MDS smacof metodas (duomenis reikia pateikti
kaip data.frame)
   source('MDS_smacof_Shiny_2.R')
   Failas_viz <<- MDS_Shiny
  } else if(Radio_kint_2 == "2"){
   #---PCA metodas (duomenis reikia pateikti kaip
data.frame)
   source('PCA_Shiny.R')
   Failas_viz <<- Rez_PCA_Shiny
  } else if(Radio_kint_2 == "3"){
   #---ICA metodas (duomenis reikia pateikti kaip
data.frame)
   source('ICA_Shiny.R')
   Failas_viz <<- Rez_ica_Shiny
  } else if(Radio_kint_2== "4"){
   #---PrincipalCurves metodas (duomenis reikia
pateikti kaip data.frame)
   source('PrinCurve_Shiny.R')
   Failas_viz <<- Rez_prinCur_Shiny
  } else if(Radio_kint_2 == "5"){
   #---LLE metodas (duomenis reikia pateikti kaip
data.frame)
   source('LLE_Shiny.R')
   Failas_viz <<- LLE_ats_Shiny
  } else if(Radio_kint_2 == "6"){
   #---Isomap metodas (duomenis reikia pateikti kaip
data.frame)
   source('Isomap_Shiny.R')
   Failas_viz <<- Rez_Isomap_Shiny
  }

……
```