VILNIAUS UNIVERSITETAS

KOTRYNA PAULAUSKIENĖ

DIMENSIJŲ MAŽINIMU PAGRĮSTAS DIDELĖS APIMTIES DUOMENŲ VIZUALIZAVIMAS IR PROJEKCIJOS PAKLAIDOS VERTINIMAS

Daktaro disertacija,
Fiziniai mokslai, informatika (09 P)

Vilnius, 2018

Disertacija rengta 2011–2017 metais Vilniaus universitete.

Mokslinė vadovė:

prof. dr. Olga Kurasova (Vilniaus universitetas, fiziniai mokslai, informatika – 09 P).

PADĖKA

Nuoširdžiai dėkoju mokslinei vadovei prof. dr. Olgai Kurasovai už besąlygišką atsidavimą, mokslines konsultacijas, visapusę pagalbą ir palaikymą rengiant disertaciją.

Dėkoju recenzentams dr. Viktorui Medvedevui ir prof. dr. Juliui Žilinskui už skirtą laiką ir vertingas pastabas bei patarimus, dėl kurių disertacija tapo kokybiškesnė.

Esu dėkinga prof. habil. dr. Mindaugui Blozneliui už laiku išsakytas vertingas pastabas rengiant disertaciją.

Už naudingus patarimus ir draugišką pagalbą norėčiau padėkoti kolegai dr. Tadui Mineikiui.

Nuoširdžiai dėkoju vyrui Robertui, dukroms Salomėjai ir Marijai už kantrybę, supratingumą ir palaikymą.

Taip pat dėkoju visiems, kurie tiesiogiai ar netiesiogiai prisidėjo prie šio darbo rengimo.

Kotryna Paulauskienė

Santrauka

Šioje disertacijoje sprendžiami uždaviniai, susiję su didelės apimties duomenimis, t. y. projekcijos paklaidos apskaičiavimas analizuojant didelės apimties duomenų aibes ir didelės apimties duomenų aibės vizualizavimas išvengiant duomenų aibės taškų persidengimo projekcijos erdvėje.

Darbo tikslas – sukurti didelės apimties duomenų projekcijos paklaidos apskaičiavimo būdus ir pasiūlyti duomenų vizualizavimo strategiją didelės apimties duomenims vizualizuoti.

Šioje disertacijoje pasiūlyti du projekcijos paklaidos apskaičiavimo būdai, tinkami didelės apimties duomenų aibėms. Vienas iš jų grindžiamas duomenų aibės imties sudarymu, antrasis – duomenų aibės dalijimu į dalis. Pasiūlyti projekcijos paklaidos apskaičiavimo būdai leidžia sutaupyti skaičiavimo laiką ir kompiuterio operatyviąją atmintį bei leidžia projekcijos paklaidą apskaičiuoti didelės apimties duomenų aibėms. Disertacijoje pasiūlyta nauja vizualizavimo strategija, leidžianti vizualizuoti didelės apimties duomenų aibes, išvengti duomenų aibės taškų persidengimo ir išlaikyti bendrą duomenų struktūrą. Vizualizavimo strategija sudaryta iš dviejų etapų: duomenų aibės imties sudarymas; imties taškų vizualizavimas be persidengimo. Visi disertacijoje pasiūlyti sprendimai pritaikyti sprendžiant realų duomenų analizės uždavinį. Atlikta išsami įvairių dimensijos mažinimo metodų (tarp jų klasikiniai gerai žinomi metodai ir metodai, kuriuose projekcija randama remiantis valdymo taškais), sprendžiant projekcijos paieškos uždavinį, lyginamoji analizė.

Tyrimų rezultatai publikuoti 6 moksliniuose leidiniuose: 3 periodiniuose recenzuojamuose mokslo žurnaluose ir 3 pateikiami konferencijos pranešimų medžiagoje. Šie rezultatai pristatyti ir aptarti 3 nacionalinėse ir 3 tarptautinėse konferencijose.

Disertaciją sudaro 6 skyriai ir literatūros sąrašas. Visa disertacijos apimtis – 119 puslapių, juose pateikti 25 paveikslai ir 19 lentelių. Disertacijoje remtasi 89 literatūros šaltiniais.

Summary

In this dissertation, the issues regarding massive data are being solved, i.e. projection error calculation for massive data sets and massive data sets visualization without point overlapping in projection space.

The goal of this research is to develop projection error evaluation approaches for massive data as well as to propose visualization approach for massive data.

In this thesis two ways to evaluate projection error for massive data sets are proposed. One of them is based on building the sample of the data set, the second one on dividing the data set into the smaller data sets. Both proposed ways of projection error evaluation are suitable for massive data sets and allow us to decrease computation time as well as to reduce the usage of computer operating memory. In this dissertation new approach of massive data visualization was proposed. The proposed visualization approach consist of two stages: selection of data subset; visualization of the projection of the data subset. This approach allows us to visualize data without points overlapping and keeps the structure of the data. All proposed approaches in this dissertation were applied to solve real world data tasks. Comprehensive analysis of various dimensionality reduction techniques was performed while solving the dimensionality reduction problem. Analysis included various classic dimensionality methods and methods which are based on control point's selection.

The main results of the dissertation were published in 6 research papers: 3 papers are published in periodicals, reviewed scientific journals and 3 papers are published in conference proceedings. The main results have been presented and discussed at 3 national and 3 international conferences.

The dissertation consists of 6 chapters and the list of references The scope of the work is 119 pages including 25 figures and 19 tables. The list of references consists of 89 sources.

Žymėjimai

d	Projekcinės erdvės, į kurią atvaizduojamas m-matis vektorius, dimensijų skaičius $d,\ d < m$
$d(X_k, X_l)$	Euklido atstumas tarp taškų X_k ir X_l
δ_{ij}	i-ojo ir j-ojo objektų skirtingumas
E_{MDS}	Paklaidos funkcija, minimizuojama daugiamačių skalių metodu
E_{Stress}	Projekcijos paklaida
E_{KT}	Konigo topologijos išlaikymo matas
k	Valdymo taškų skaičius
τ	Klasterių skaičius
m	Vektoriaus komponenčių skaičius; objektą apibūdinančių požymių (parametrų) skaičius
n	Analizuojamų objektų (vektorių) skaičius
N	Taškų kandidatų skaičius
$\mathbb{R}^d$	d-matė Euklido erdvė
S	Silueto koeficientas
ρ	Koreliacijos koeficientas
$X = \{X_1, \dots, X_n\}$	Taškų rinkinys daugiamatėje erdvėje
$X_i = (x_{i1}, x_{i2}, \dots, x_{im})$	Taško koordinatės daugiamatėje erdvėje, $X_i \in \mathbb{R}^m$
$Y = \{Y_1, \dots, Y_n\}$	Taškų rinkinys mažesnės dimensijos erdvėje
$Y_i = (y_{i1}, y_{i2}, \dots, y_{id})$	Taško koordinatės, vektoriaus X_i transformacija į mažesnio skaičiaus dimensijų erdvę $\mathbb{R}^d, d < m$

Santrumpos

LAMP	Lokaliosios afiniosios daugiamatės projekcijos metodas (angl. *local affine multidimensional projection)*
MDS	Daugiamačių skalių metodas (angl. *multidimensional scaling)*
PCA	Pagrindinių komponenčių metodas (angl. *principal component analysis*)
PLMP	Dalinai tiesinės daugiamatės projekcijos metodas (angl. *part-linear multidimensional projection)*
RBF	Radialinės bazinės funkcijos (angl. *radial basis functions)*
ROLS	Reguliarizuotų ortogonalių mažiausių kvadratų metodas (angl. *regularized orthogonal least squares method)*
RP	Atsitiktinė projekcija (angl. *random projection)*

Turinys

Įvadas

1.1 Tyrimo sritis ir problemos aktualumas

Mokslo, inžinerijos, telekomunikacijų, finansų, medicinoje ir kitose srityse nuolat susiduriama su didelės apimties duomenų aibėmis. Vystantys technologijoms kaupiamų duomenų apimtys sparčiai didėja. Nors objektų skaičius didelis, kiekvieną iš jų nusako ir daug požymių, tačiau analizuojant daugiamačius duomenis dažnai ne visi būna svarbūs. Dimensijos mažinimo metodai leidžia mums geriau suprasti daugiamačius duomenis. Metodų tikslas – pateikti objektus, apibūdinančius duomenis mažesnės dimensijos erdvėje (projekcijos erdvėje), taip, kad būtų kiek galima tiksliau išlaikyti tam tikrą duomenų struktūrą ir būtų lengviau apdoroti ir vizualizuoti didelės dimensijos duomenis. Duomenų vizualizavimas leidžia geriau suprasti turimus duomenis, pastebėti išskirtinumus, grupavimosi tendencijas, tarpusavio ryšius, t. y. atskleisti duomenų struktūrą. Analizuojant daugiamačius didelės apimties duomenis nėra tikslinga vizualizuoti keliasdešimt tūkstančių ar milijoną taškų sklaidos diagramoje, nes jie gali susitelkti ir vienas kitą perdengti. Yra sukurta įvairių duomenų vizualizavimo sistemų, tačiau dauguma iš jų duomenis leidžia vizualizuoti tik agreguotus arba juos vizualizuoti pagal keletą daugiamačių duomenų požymių. Vis dėlto norint atsižvelgti į visus duomenų aibės požymius ir matyti ne agreguotus duomenis, o identifikuoti kiekvieno taško poziciją, trūksta vizualizavimo būdų. Be to, taikant vizualizavimo metodus, pagrįstus duomenų dimensijos mažinimu, reikia įvertinti gautos projekcijos kokybę. Dažniausiai dimensijų mažinimo metodas turi savo kriterijų, pagal kurį ieškoma optimali projekcija. Gauta projekcija gali būti vertinama taikant tą patį kriterijų. Tačiau norint įvertinti keliais metodais gautas projekcijas, naudojami kiti nuo metodo nepriklausantys matai, atspindintys įvairias duomenų ypatybes. Dažniausiai dimensijų mažinimo metodų rezultatams tyrimuose vertinti naudojama projekcijos paklaida. Projekcijos paklaidos reikšmėms skaičiuoti dažnai naudojami atstumai tarp taškų. Nagrinėjant

didelės apimties duomenų aibes kyla projekcijos paklaidos įvertinimo problema, kadangi apskaičiuoti ją naudojamos didelės apimties atstumų matricos, o šioms apskaičiuoti gali pritrūkti personalinio kompiuterio operatyviosios atminties. Nors ir analizuojant didelės apimties duomenų aibes dimensija tam tikrais dimensijos mažinimo metodais gali būti sumažinama labai greitai, tačiau dėl anksčiau minėtos priežasties projekcijos paklaidos įvertinimas trunka labai ilgai arba reikalauja daug kompiuterio operatyvios atminties.

Taigi šioje disertacijoje sprendžiamos šios pagrindinės problemos:

1. Projekcijos paklaidos apskaičiavimas analizuojant didelės apimties duomenų aibes.
2. Didelės apimties duomenų aibės vizualizavimas išvengiant duomenų aibės taškų persidengimo projekcijos erdvėje.

1.2 Tyrimo objektas

Disertacijos tyrimo objektas:

- didelės apimties daugiamačiai duomenys;
- dimensijų mažinimo metodai didelės apimties daugiamačiams duomenis vizualizuoti ir projekcijos paklaidų įvertinimas.

1.3 Darbo tikslas ir uždaviniai

Darbo tikslas – sukurti didelės apimties duomenų projekcijos paklaidos apskaičiavimo būdus ir pasiūlyti duomenų vizualizavimo strategiją didelės apimties duomenims vizualizuoti.

Siekiant tikslo būtina spręsti šiuos uždavinius:

- atlikti dimensijų mažinimo metodų, skirtų daugiamačiams duomenims vizualizuoti, ir projekcijos kokybės įvertinimo būdų analitinę apžvalgą;
- pasiūlyti daugiamačių duomenų projekcijos į mažesnio matmenų skaičiaus erdvę apskaičiavimo būdus, leidžiančius projekcijos paklaidą vertinti didelės apimties duomenims;

- eksperimentiškai palyginti pasiūlytus didelės apimties duomenų projekcijos apskaičiavimo būdus su jau žinomais būdais;
- pasiūlyti ir ištirti didelės apimties duomenų aibės vizualizavimo strategiją, leidžiančią išvengti duomenų aibės taškų persidengimo ir išlaikyti bendrą duomenų struktūrą;
- pasiūlytus sprendimus pritaikyti realių duomenų vizualiosios analizės uždaviniui.

1.4 Tyrimo metodai

Analizuojant dimensijos mažinimo ir duomenų vizualizavimo srities mokslinius ir eksperimentinius pasiekimus naudoti informacijos paieškos, sisteminimo, analizės, lyginamosios analizės ir apibendrinimo metodai. Remiantis eksperimentinio tyrimo metodu, atlikta statistinė duomenų ir tyrimų rezultatų analizė, o šios rezultatams įvertinti naudotas apibendrinimo metodas.

1.5 Darbo mokslinis naujumas

1. Pasiūlyti du projekcijos paklaidos apskaičiavimo būdai, tinkami didelės apimties duomenų aibėms. Vienas iš jų grindžiamas duomenų aibės imties sudarymu, antrasis – duomenų aibės dalijimu į dalis.
2. Pasiūlyta nauja vizualizavimo strategija, leidžianti vizualizuoti didelės apimties duomenų aibes, išvengti duomenų aibės taškų persidengimo ir išlaikyti bendrą duomenų struktūrą.
3. Atlikta išsami įvairių dimensijos mažinimo metodų, sprendžiant projekcijos paieškos uždavinį, lyginamoji analizė.

1.6 Ginamieji teiginiai

1. Pasiūlyti projekcijos paklaidos apskaičiavimo būdai yra tinkami apskaičiuoti projekcijos paklaidą didelės apimties duomenų aibėms.
2. Pasiūlyta nauja vizualizavimo strategija yra tinkama didelės apimties duomenų aibėms vizualizuoti, išvengti duomenų aibės taškų persidengimo ir išlaikyti bendrą duomenų struktūrą.

1.7 Darbo rezultatų praktinė reikšmė

Pasiūlyti projekcijos paklaidos apskaičiavimo būdai leidžia sutaupyti skaičiavimo laiką ir kompiuterio operatyviąją atmintį bei leidžia projekcijos paklaidą apskaičiuoti didelės apimties duomenų aibėms. Pasiūlyta didelės apimties duomenų aibių vizualizavimo strategija leidžia vizualizuoti didelės apimties duomenų aibes, išlaikyti duomenų struktūrą ir išvengti taškų persidengimo. Pasiūlytas duomenų aibės imties sudarymo būdas gali būti naudojamas ne tik didelės apimties duomenims vizualizuoti, bet ir sprendžiant duomenų analizės uždavinius įvairiose srityse. Visi disertacijoje pasiūlyti būdai gali būti taikomi sprendžiant realius duomenų analizės uždavinius.

1.8 Darbo rezultatų aprobavimas

Tyrimų rezultatai publikuoti 6 moksliniuose leidiniuose: 3 periodiniuose recenzuojamuose mokslo žurnaluose, iš jų 1 leidinyje, referuojamame „Clarivate Analytics Web of Science“ duomenų bazėje ir turinčiame citavimo indeksą, ir 3 pateikiami konferencijos pranešimų medžiagoje.

Tyrimų rezultatai pristatyti ir aptarti šiose nacionalinėse ir tarptautinėse konferencijose Lietuvoje ir užsienyje:

1. XVI mokslinė kompiuterininkų konferencija „Kompiuterininkų dienos 2013“, 2013 m. rugsėjo 19–21 d., Šiauliai, Lietuva. Pranešimo pavadinimas: „Duomenų tyrybos sistemų galimybių tyrimas įvairių apimčių duomenims analizuoti“.
2. 19-oji tarpuniversitetinė magistrantų ir doktorantų konferencija „Informacinė visuomenė ir universitetinės studijos“ (IVUS 2014), 2014 m. balandžio 24 d., Kaunas, Lietuva. Pranešimo pavadinimas: „Dimensijų mažinimo metodų tyrimas įvairių apimčių duomenims analizuoti“.
3. Lietuvos matematikų draugijos 55-oji konferencija, 2014 m. birželio 26–27 d., Vilnius, Lietuva. Pranešimo pavadinimas: „Dimensijų mažinimo metodais gautų projekcijų įvertinimas“.

4. 8-th International Workshop „Data Analysis Methods for Software Systems“, 1–3 December, 2016, Druskininkai, Lithuania. Pranešimo pavadinimas: „Massive Data Visualization via Selecting a Data Subset“.
5. 25-th International Conference on Computer Graphics, Visualization and Computer Vision 2017, May 29 – June 2, 2017, Pilzen, Czech Republic. Pranešimo pavadinimas: „A new dimensionality reduction-based visualization approach for massive data“.
6. 6-th International Conference on Advanced Technology & Sciences, 12–15 September, 2017, Riga, Latvia. Pranešimo pavadinimas: „Improvement of projection error evaluation for massive data sets“.

1.9 Disertacijos struktūra

Disertaciją sudaro 6 skyriai ir literatūros sąrašas. Disertacijos skyriai: Įvadas, Dimensijos mažinimo ir vizualizavimo metodų apžvalga, Projekcijos paklaidos apskaičiavimas ir strategija didelės apimties duomenims vizualizuoti, Eksperimentinių tyrimų rezultatai, Pasiūlytų sprendimų taikymas meteorologinių duomenų aibės analizei, Bendrosios išvados. Be to, disertacijoje pateiktas naudotų žymėjimų ir santrumpų sąrašai. Visa disertacijos apimtis – 119 puslapių, juose pateikti 25 paveikslai ir 19 lentelių. Disertacijoje remtasi 89 literatūros šaltiniais.

2 Dimensijos mažinimo ir vizualizavimo metodų apžvalga

Duomenų kiekis pasaulyje auga sparčiai. Dažnai duomenys nusakomi požymiais, kurių taip pat yra daug. Dimensijos mažinimo metodai leidžia geriau suprasti daugiamačius duomenis ir įžvelgti juose naudingos informacijos.

Pastaruoju metu duomenų vizualizavimo tematika plačiai nagrinėjama mokslinėje literatūroje visame pasaulyje. Šioje srityje dirba nemažai ir Lietuvos mokslininkų. Per pastarąjį dešimtmetį sėkmingai apginta daug disertacijų, kuriose nagrinėjami dimensijos mažinimo metodai ir duomenų vizualizavimas. Šioje disertacijoje taip pat sprendžiami panašūs dimensijos mažinimu pagrįsto vizualizavimo ir su projekcijos paklaidos apskaičiavimu susiję uždaviniai. J. Bernatavičienės disertacijoje [1] taip pat nagrinėjami daugiamačiai duomenys ir jų vizualizavimo metodai. J. Bernatavičienė disertacijoje siūlo 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*). Šioje disertacijoje nagrinėjami kiti baziniais vektoriais (šioje disertacijoje vadinamais valdymo taškais) paremti dimensijos mažinimo metodai ir nagrinėjamų duomenų aibių apimtys yra dešimtimis, o kai kuriais atvejais ir šimtais kartų didesnės už nagrinėtųjų J. Bernatavičienės disertacijoje.

R. Karbauskaitė disertacijoje [2] nagrinėjo daugiamačių duomenų vizualizavimo algoritmus ir metodus, išlaikančius lokalią struktūrą, ir daugiamačių duomenų projekcijų mažesnės dimensijos erdvėje vertinimo kriterijus. Ištirti ir lyginti trys topologijos išlaikymo matai: Spirmeno koeficientas, Konigo matas ir kaimynystės klaidos, tinkami analizuoti daugdaros topologijos išlaikymui po jos transformavimo į mažesnės dimensijos

erdvę. Šioje disertacijoje taip pat tirti įvairūs projekcijos kokybės įvertinimo matai.

V. Medvedev daktaro disertacijoje [3] nagrinėja tiesioginio sklidimo dirbtinius neuroninius tinklus, skirtus daugiamačiams duomenims vizualizuoti. Saviorganizuojantys neuroniniai tinklai taip pat gali būti naudojami duomenims klasterizuoti ir vizualizuoti. P. Stefanovič disertacijos [4] tyrimo objektas – duomenų klasterizavimas, klasifikavimas ir vizualizavimas taikant saviorganizuojančius neuroninius tinklus ir jų kokybės vertinimas. Daugiamačių duomenų vizualizavimą taikant neuroninius tinklus tyrinėja O. Kurasova su bendraautoriais [5], [6]. V. Marcinkevičiaus disertacijos [7] tyrimo objektas – daugiamačiai duomenys, jų atvaizdavimas netiesiniais daugiamačių skalių algoritmais ir saviorganizuojančiais neuroniniais tinklais, projekcijos kokybės vertinimas. N. Galiauskas disertacijoje [8] nagrinėja daugiamačių duomenų vizualizavimo problematiką spręsdamas minimizavimo uždavinį, kylantį iš daugiamačių skalių metodo su miesto kvartalo atstumais.

A. Žilinskas ir J. Žilinskas tyrinėja daugiamačių skalių metodą su miesto kvartalų metrika (angl. *city-block*) ir ieško daugiamačių skalių paklaidos (įtempimo, tikslo) funkcijos globalaus minimumo [9], [10], [11], [12]. G. Dzemyda, O. Kurasova ir J. Žilinskas monografijoje [13] nagrinėja įvairius daugiamačių duomenų vizualizavimo metodus, jų modifikacijas ir taikymus, tačiau nagrinėjami metodai nėra pagrįsti valdymo taškais ir analizuojamos duomenų apimtys nėra didelės.

Nors visuose minėtuose darbuose nagrinėjami daugiamačiai duomenys, dimensijos mažinimo ir vizualizavimo metodai, tačiau nė viename iš jų nėra nagrinėjami didelės apimties duomenys ir jų vizualizavimas, bei projekcijos paklaidos apskaičiavimas didelės apimties duomenims.

Toliau šiame skyriuje pateikti didelės apimties duomenų apibrėžimai, aprašytos šiuolaikinės technologijos, taikomos didelės apimties duomenų aibėms. Taip pat aprašyti įprasti dimensijos mažinimo metodai ir metodai, paremti valdymo taškais, ir projekcijos kokybės įvertinimo matai. Taip pat

pateikta duomenų tyrybos sistemų lyginamoji analizė. Pateiktos apžvalgos ir analizės publikuotos autorės darbuose [A1], [A3], [B1], [B2].

2.1 Didelės apimties duomenų apibrėžtis

Dažnai kyla klausimas, kokie duomenys gali būti įvardijami kaip didelės apimties. Vienareikšmį atsakymą į šį klausimą sunku rasti. Duomenys, kurie prieš kelerius metus buvo didelės apimties, atsiradus greitesniems duomenų apdorojimo įrenginiams ir metodams, laikomi nedidelės apimties. Tarptautinė duomenų korporacija (angl. *International Data Corporation*) prognozuoja, kad 2020 m. visa pasaulio duomenų apimtis sudarys 44 zeta baitus [14].

Prieš du dešimtmečius atsirado terminas „*big data*“, kurį vieni iš pirmųjų pavartojo mokslininkai iš Nacionalinės aeronautikos ir kosmoso administracijos (angl. *National Aeronautics and Space Administration*, NASA) aprašydami duomenų vizualizavimo uždavinį [15]. Viena iš didžiųjų duomenų (angl. *big data*) apibrėžčių yra ta, kad tai duomenys, kurių apimtis tokia didelė, kad jiems saugoti ir apdoroti nėra tinkami įprasti saugojimo ir apdorojimo metodai ir sistemos. Šiuo metu kyla daug diskusijų, kas yra didieji duomenys, tačiau sutinkama, kad tai duomenys, kurie yra tokie dideli, kad jiems apdoroti nepakanka vieno kompiuterio aparatinės įrangos [16]. 2001 m. Doug Laney nurodė tris svarbiausius didžiųjų duomenų aspektus, anglų kalba apibūdinamus trimis V raidėmis (angl. *3 V's*). Didžiuosius duomenis nusako trys charakteristikos: didelė apimtis (angl. *high volume*), didelė sparta (angl. *high velocity*), didelė įvairovė (angl. *high variety*) [17]. Paskui pasiūlytos dar dvi charakteristikos, tai vertingumas (angl. *value*) ir teisingumas (angl. *veracity*) [18]. Didžiųjų duomenų apibrėžtis nuolat plečiama.

Kitas terminas, susijęs su didele duomenų apimtimi, yra „*massive data"*. Čia įprastai nėra atsižvelgiama į kitas didžiųjų duomenų charakteristikas, tokias kaip sparta, įvairovė, o tik apsiribojama didelės apimties charakteristika. Šiame darbe tokie duomenys vadinami didelės apimties.

2.2 Dimensijos mažinimo metodai

Iš pradžių apibrėšime pagrindines sąvokas ir žymėjimus, vartojamus šioje disertacijoje. Nagrinėjant daugiamačius duomenis, jų projekciją, duomenų vizualizavimo metodus, sutinkamos *objekto* ir *požymių* sąvokos. *Objekto* (angl. *object*) sąvoka gali apimti įvairius daiktus, reiškinius, įrenginius, augalus, gyvūnus, žmones ir kt. Objektai, sudarantys konkrečią analizuojamų objektų aibę, nusakomi bendrais požymiais. Požymiai gali būti vadinami ir parametrais, ypatybėmis (angl. *features, parameters, attributes*). Objektų tokioje aibėje skaičius n yra baigtinis. Tam tikras visų požymių reikšmių rinkinys nusako vieną konkretų analizuojamos aibės objektą $X_i = (x_{i1}, x_{i2}, \dots, x_{im}),\ i \in \{1, \dots, n\}$, čia m yra požymių skaičius, i yra objekto eilės numeris. Kai objektą $X_i = (x_{i1}, x_{i2}, \dots, x_{im})$ apibūdina daugiau kaip vienas požymis, duomenys, charakterizuojantys nagrinėjamą objektą, yra daugiamačiai. $X_i = (x_{i1}, x_{i2}, \dots, x_{im})$ dažnai dar vadinami vektoriais ar taškais, parametrai $x_1, x_2, \dots, x_m$ – požymiais [19]. Kai požymiai įgyja tam tikras skaitines reikšmes, tai analizuojami duomenys yra matrica:

$$X = \{X_1, \dots, X_n\} = \{x_{ij}, i = 1, \dots, n, j = 1, \dots, m\},$$

kurios i-oji eilutė yra vektorius $X_i \in \mathbb{R}^m$, čia $X_i = (x_{i1}, x_{i2}, \dots, x_{im})$, $i \in \{1, \dots, n\}$, n – analizuojamų objektų (vektorių) skaičius.

Duomenų analizė padeda aptikti paslėptas duomenų struktūras, koreliacijas, daryti įžvalgas, padedančias priimti svarbius sprendimus. Kuo didesnė duomenų apimtis (objektų ir požymių skaičius), tuo sunkiau tuos duomenis suprasti žvelgiant į duomenų lentelę. Vienas iš būdų, leidžiantis geriau (giliau) suvokti duomenis yra *vizualizavimas*, t. y. grafinis informacijos pateikimas. Duomenų vizualizavimo būdai pateikia duomenis grafiškai ir leidžia geriau juos suprasti. Praktikoje dažnai susiduriama su daugiamačiais duomenimis. Norint šiuos duomenis vizualizuoti dvimatėje arba trimatėje erdvėje, reikia sumažinti jų pradinę dimensiją. Dimensijos mažinimo metodai, dar vadinami projekcijos metodais, m-matės erdvės taškus $X_1, \dots, X_n$ transformuoja į d-matės erdvės taškus $Y_1, \dots, Y_n,\ d < m$. Šių metodų tikslas –

pateikti objektus, apibūdinančius daugiamačius duomenis, mažesnės dimensijos erdvėje taip, kad būtų kiek galima tiksliau išlaikyti tam tikrą duomenų struktūra, ir būtų lengviau apdoroti ir interpretuoti didelės dimensijos duomenis. Yra tokių dimensijos mažinimo metodų, kuriuose vietoje aibės $X = \{X_1, \dots, X_n\}$ gali būti panaudota objektų artimumų (angl. *proximity*) matrica $D = \{\delta_{ij}, i, j = 1, \dots, n\}$. Paprasčiausias variantas būtų Euklido atstumas tarp dviejų taškų $X_k = (x_{k1}, x_{k2}, \dots, x_{km})$ ir $X_l = (x_{l1}, x_{l2}, \dots, x_{lm})$, apskaičiuojamas pagal šią formulę:

$$d(X_k, X_l) = \sqrt{\sum_{j=1}^{m} \left(x_{kj} - x_{lj}\right)^2}, \; k, l = 1, \dots, n.$$

Šioje disertacijoje nagrinėjamos artimumų matricos, sudarytos iš Euklido atstumų, tačiau gali būti naudojamos ir kitos metrikos, tokios kaip Minkovskio, miesto kvartalo, Čebyševo [19].

Radus daugiamačių duomenų projekciją (dvimatėje ar trimatėje erdvėje) ir juos vizualizavus daug lengviau suvokti duomenų struktūrą ir jų ryšį.

Šiame poskyryje apžvelgiami dažniausiai nagrinėjami ir taikomi įprasti dimensijos mažinimo metodai ir pastaruoju metu literatūroje pasiūlyti nauji dimensijos mažinimo metodai, paremti valdymo taškais (angl. *control points*) [20], [21].

2.2.1 Įprasti dimensijos mažinimo metodai

Daugiamačių skalių metodas. Vienas iš populiariausių dimensijos mažinimo metodų – daugiamačių skalių metodas (angl. *multidimensional scaling*, MDS) [22]. Šis metodas ir jo realizacijos plačiai naudojami daugiamačiams duomenims analizuoti [13], [19]. Taikant MDS metodą, ieškomos daugiamačių duomenų projekcijos mažesnės dimensijos erdvėje, siekiama išlaikyti analizuojamos aibės objektų artimumus – panašumus arba skirtingumus [13], [19]. Pradiniai MDS metodo duomenys – kvadratinė simetrinė matrica, kurios elementai nusako artimumą tarp analizuojamų objektų. Paprasčiausias variantas – Euklido atstumų tarp objektų matrica. MDS metodo tikslas yra rasti optimalų daugiamačius objektus atitinkančių taškų

vaizdą mažesnės dimensijos erdvėje. Tarkime, kiekvieną m-matį tašką $X_i \in \mathbb{R}^m, i \in \{1, \dots, n\}$ atitinka mažesnės dimensijos taškas $Y_i \in \mathbb{R}^d, d < m$, čia n – taškų skaičius. Atstumą tarp taškų X_i ir X_j pažymėkime $d(X_i, X_j)$, o atstumą tarp taškų Y_i ir Y_j – $d(Y_i, Y_j)$, $i, j = 1, \dots, n$. Taikant MDS metodą, atstumai $d(Y_i, Y_j)$ bandomi priartinti prie atstumų $d(X_i, X_j)$, t. y. minimizuojama paklaidos (dar vadinama *Stress*) funkcija:

$$E_{\text{MDS}} = \sum_{i<j} \left(d(X_i, X_j) - d(Y_i, Y_j) \right)^2. \qquad (1)$$

Šioje disertacijoje *Stress* funkcijai minimizuoti taikomas SMACOF (angl. *Scaling by MAjorizing a COmplicated Function*) algoritmas – vienas iš populiariausių minimizavimo algoritmų daugiamatėms skalėms [13], [22]. Šis algoritmas remiasi tikslo funkcijos mažoravimu. Čia mažiausiųjų kvadratų paklaidos funkcijos *Stress* minimizavimas pakeistas paprastesniu pagalbinės funkcijos iteraciniu minimizavimu. Šis metodas užtikrina paklaidos funkcijos E_{MDS} monotonišką konvergavimą [22].

SMACOF algoritmas:

1 žingsnis: inicializuojami dvimačiai aibės $Y = \{Y_1, Y_2, \dots, Y_n\}$ vektoriai. Pradinė iteracija lygi $t = 0$, čia t yra iteracijos numeris;

2 žingsnis: apskaičiuojama projekcijos paklaida $E_{\text{MDS}}(Y(t))$ pagal (1) formulę;

3 žingsnis: iteracijų numeris t padidinamas vienetu;

4 žingsnis: apskaičiuojami dvimačiai vektoriai pagal formulę $Y(t+1) = n^{-1} B(Y(t)) Y(t)$, čia matricos $B(Y(t))$ elementai apskaičiuojami pagal formules:

$$b_{ij} = \begin{cases} \frac{-d(X_i, X_j)}{d(Y_i, Y_j)}, \text{kai } i \neq j \text{ ir } d(Y_i, Y_j) \neq 0 \\ 0, \qquad \text{kai } i \neq j \text{ ir } d(Y_i, Y_j) = 0 \end{cases},$$

$$b_{ii} = -\sum_{j=1,\ j \neq i}^{m} b_{ij}, \text{kai } i = j.$$

5 žingsnis: apskaičiuojamas $E_{\text{MDS}}(Y(t))$ pagal (1) formulę. Jeigu $E_{\text{MDS}}(Y(t-1)) - E_{\text{MDS}}(Y(t)) < \varepsilon$ arba t yra lygus maksimaliam

iteracijų skaičiui, iteracinis procesas stabdomas (ε yra maža teigiama konstanta). Priešingu atveju, algoritmas kartojamas nuo *3 žingsnio.*

MDS vienos iteracijos skaičiavimų sudėtingumas SMACOF algoritme yra $O(mn^2)$, čia m – požymių skaičius, n – objektų skaičius. Jeigu analizuojamos didelės apimties duomenų aibės, svarbus MDS faktorius yra skaičiavimo laikas. Yra būdų MDS skaičiavimo laikui sumažinti. Vienas iš jų – klasterizuojant sumažinti analizuojamos duomenų aibės objektų skaičių n ir analizuoti gautą mažesnę duomenų aibę.

Pagrindinių komponenčių analizė. Pagrindinių komponenčių analizės (angl. *principal component analysis,* PCA) metodo tikslas – sumažinti duomenų dimensiją atliekant tiesinę transformaciją ir atsisakant dalies po transformacijos gautų naujų komponenčių, kurių dispersijos yra mažiausios [23], [24], [25], [26]. Šį metodą sudaro duomenų kovariacinės matricos tikrinių reikšmių λ ir tikrinių vektorių E skaičiavimai. Kiekvienas tikrinis vektorius yra vadinamas pagrindine komponente. Pagrindinės komponentės yra lygties $CE = \lambda E$ sprendinys E. 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 pat kaip 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 $X_i \in \mathbb{R}^m$ 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.

Nepriklausomų komponenčių analizė. Literatūroje nurodoma, kad daugiamačių taškų dimensijai mažinti gali būti naudojamas nepriklausomų komponenčių analizės metodas (angl. *independent component*

analysis, ICA) [27], [28], nors tiesioginė šio metodo paskirtis nėra dimensijos mažinimas. ICA – tai metodas, transformuojantis pagrindines komponentes į statistiškai nepriklausomas komponentes. Pagrindinis tikslas – rasti komponentes, kurios būtų maksimaliai nepriklausomos ir nebūtų pasiskirsčiusios pagal normalųjį skirstinį [29]. Detalesnė informacija apie nepriklausomų komponenčių analizės metodą pateikta darbuose [30], [31]. Hyvarinen ir Oja [32] pasiūlė greitą fiksuoto taško algoritmą nepriklausomoms komponentėms rasti (angl. *FactICA algorithm*). Nepriklausomų komponenčių modelis užrašomas kaip m tiesinių nepriklausomų komponenčių mišinių $x_1, \dots, x_m$ [32]:

$$x_j = a_{j1}s_1'' + a_{j2}s_2'' + \cdots + a_{jm}s_m'',$$

čia s_i'' – nepriklausomos komponentės (realus signalas), a_{ji} – sumaišymo elementai, x_j – fiksuojamas signalas.

Kaip matrica šis modelis užrašomas taip:

$$X = AS''.$$

Nepriklausomos komponentės yra latentiniai kintamieji, kurie negali būti stebimi tiesiogiai. Taigi turint atsitiktinį vektorių X, turime rasti sumaišymo matricos A ir vektoriaus S'' įverčius.

ICA daroma prielaida, kad komponentės s_i'' yra statistiškai nepriklausomos ir nėra pasiskirsčiusios pagal normalųjį skirstinį. Taip pat priimama, kad nežinoma sumaišymo matrica A yra kvadratinė. Radus matricą A, galima apskaičiuoti jos atvirkštinę matricą A^{-1} ir taip rasti nepriklausomas komponentes:

$$S'' = A^{-1}X.$$

Rengiant pradinius duomenis ICA atliekami du žingsniai: centravimas ir balinimas.

FastICA algoritmas yra pagrįstas fiksuotojo taško paieška iteraciniu metodu (angl. *fixed-point iteration*). Iteracijomis randami svertiniai sumaišymo matricos koeficientai, atskiriantys ieškomą signalą nuo signalo, sumaišyto su

triukšmu. Straipsnyje [32] pateikiamas *FastICA* algoritmas nepriklausomoms komponentėms išskirti.

Pagrindinė ICA metodo problema ta, kad tarp nepriklausomų komponenčių nėra tvarkos, tokios kaip PCA metode. Pastaruoju norint rasti pagrindines komponentes užtenka rasti kovariacinės matricos pirmąsias didžiausias tikrines reikšmes ir jas atitinkančius tikrinius vektorius. Nepriklausomų komponenčių metodu pirmosios išskirtos nepriklausomos komponentės nebūtinai bus reikšmingesnės už vėliau išskirtas. Išskirtas nepriklausomas komponentes galima rūšiuoti, apskaičiuojant sumaišymo matricos A stulpelių vektorių normas arba pasinaudojant viena iš negausiškumo skaitinių charakteristikų [27]. Darbe [33] siūloma nepriklausomas komponentes rūšiuoti atsižvelgiant į jų negentropijos koeficiento aproksimaciją:

$$J(IC_i) = \left(\frac{1}{12}\right)[\kappa_i^3]^2 + \left(\frac{1}{48}\right)[\kappa_i^4 - 3]^2,$$

čia IC_i – i-toji nepriklausoma komponentė, κ_i^3 ir κ_i^4 yra trečiasis ir ketvirtasis momentai. Yra ir kitas būdas nepriklausomoms komponentėms pasirinkti, t. y. taikant ICA metodą balinimo metu galima atlikti dimensijos mažinimą, o tada išskirti nepriklausomas komponentes.

Atsitiktinės projekcijos metodas. Pastaruoju metu dažnai naudojamas dimensijos mažinimo metodas yra atsitiktinės projekcijos metodas (angl. *random projection,* RP). Šiuo metodu pradiniai duomenys iš m-matės erdvės transformuojami į d-matę erdvę ($d \ll m$) taikant atsitiktinių reikšmių matricą R, sudarytą iš d eilučių ir m stulpelių, kurios stulpeliuose esantys vektoriai yra vienetinio ilgio [34].

Pradinių duomenų $X_{n\times m}$ projekcija $Y_{n\times d}$ iš m-matės erdvės į mažesnę d-matę erdvę užrašoma taip:

$$Y_{n\times d} = X_{n\times m}R_{m\times d}.$$

Pagrindinė atsitiktinės projekcijos idėja yra kilusi iš Johnson-Lindenstrauss lemos [35], teigiančioje, kad sekai n taškų iš m-matės Euklido erdvės egzistuoja tiesinė transformacija į mažesnės dimensijos d-matę erdvę

($d \geq O(\varepsilon^{-2} \log n)$), tokia, kurioje atstumai tarp taškų yra apytiksliai išlaikomi, t. y. neiškreipiami daugiau už dydį $1 \pm \varepsilon$.

Remiantis Johnson-Lindenstrauss lema Dasgupta ir Gupta savo darbe [36] formuluoja šį teiginį:

Tarkime, turime taškų matricą $X \in \mathbb{R}^{n \times m}$. Tada bet kokiam $\varepsilon > 0$ ir $d \geq 4\left(\frac{\varepsilon^2}{2} - \frac{\varepsilon^3}{3}\right)^{-1} \log n$ egzistuoja projekcija $f: \mathbb{R}^m \to \mathbb{R}^d$, tokia, kad visiems $u, v \in X$ turime $(1-\varepsilon)\|u-v\|^2 \leq \|f(u)-f(v)\|^2 \leq (1+\varepsilon)\|u-v\|^2$.

Standartinės atsitiktinės projekcijos metodu (angl. *conventional random projection*) pradinių duomenų matrica dauginama iš atsitiktinės matricos $R_{m \times d}$, kurios elementai r_{ij} yra atsitiktiniai dydžiai ($r_{ij} \sim \mathcal{N}(0,1)$).

Achlioptas [37] duotai pradinių duomenų matricai $X \in \mathbb{R}^{n \times m}$ ir pasirinktiems $\varepsilon, \beta > 0$, $d = \frac{4+2\beta}{\frac{\varepsilon^2}{2} - \frac{\varepsilon^3}{3}} \log n$, kur parametras ε reguliuoja pageidaujamą tikslumą išlaikydamas atstumus, parametras β – projekcijos sėkmės tikimybę, siūlo tokias matricos R elementų r_{ij} reikšmes:

$$r_{ij} = \sqrt{3} \cdot \begin{cases} +1 \text{ su tikimybe } \frac{1}{6} \\ 0 \text{ su tikimybe } \frac{2}{3}, \\ -1 \text{ su tikimybe } \frac{1}{6} \end{cases}$$

arba

$$r_{ij} = \begin{cases} +1 \text{ su tikimybe } \frac{1}{2} \\ -1 \text{ su tikimybe } \frac{1}{2} \end{cases}.$$

Li ir kt. [38] siūlo labai retą atsitiktinę projekciją (angl. *very sparse random projection*) ir tokias matricos R elementų r_{ij} reikšmes:

$$r_{ij} = \sqrt{s} \cdot \begin{cases} +1 \text{ su tikimybe } \frac{1}{2s} \\ 0 \text{ su tikimybe } 1 - \frac{1}{2s}, \\ -1 \text{ su tikimybe } \frac{1}{2s} \end{cases}$$

čia $s = \sqrt{m}$ arba $s = \frac{m}{\log m}$. Achlioptas pasiūlyto atveju $s = 1$ arba $s = 3$.

2.2.2 Valdymo taškais paremti dimensijos mažinimo metodai

Šiame disertacijos skyrelyje nagrinėjami valdymo taškais (angl. *control points*) paremti metodai, taikantys Euklido atstumų matricą ne tarp visų duomenų aibės taškų, bet tik tarp dalies taškų. Taip sutaupomi skaičiavimo kaštai. Iš pradžių atrenkama dalis duomenų aibės taškų, vadinamų valdymo taškais, paskui randama jų vieta mažesnės dimensijos erdvėje ($\mathbb{R}^d, d = 2$). Informacija, gauta iš valdymo taškų, naudojama likusiųjų taškų projekcijai rasti. J. Bernatavičienės disertacijoje [1] nagrinėjamas santykinių daugiamačių skalių metodas, paremtas baziniais vektoriais (angl. *basic vectors*). J. Bernatavičienė siūlo bazinių vektorių atrinkimo būdus ir nustato optimalų bazinių vektorių skaičių (mažesnėms duomenų aibėms (iki 3000 vektorių) tikslinga imti nuo 700 iki 1000 bazinių vektorių, o didelėms duomenų aibėms – nuo 900 iki 1500). Šioje disertacijoje aptariami trys pastaruoju metu pasiūlyti ir valdymo taškais paremti metodai dimensijai mažinti: dalinai tiesinė daugiamatė projekcija [21], lokalioji afinioji daugiamatė projekcija [20] ir metodas, paremtas radialinių bazinių funkcijų teorija ir valdymo taškais [39]. Kitaip nei J. Bernatavičienė, apie dalinai tiesinės daugiamatės projekcijos ir lokaliosios afiniosios daugiamatės projekcijos metodus rašę autoriai yra nustatę, kad geriausia pasirinkti $k = \sqrt{n}$ valdymo taškų, kur n – taškų skaičius duomenų aibėje, nes taip gaunama pusiausvyra tarp skaičiavimo kaštų ir taškų atvaizdavimo kokybės.

Dalinai tiesinė daugiamatė projekcija. Dalinai tiesinės daugiamatės projekcijos (angl. *part-linear multidimensional projection*, PLMP) metodu siūloma naudoti dalį atstumų tarp duomenų aibės taškų [21]. Tarkime, $X = \{X_1, \dots, X_n\}$ yra duomenų aibė, sudaryta iš taškų $X_i \in \mathbb{R}^m$, reikia surasti tiesinę transformaciją Φ: $\mathbb{R}^m \to \mathbb{R}^d$, $d < m$, kuri tenkintų šią išraišką:

$$\Phi = \underset{\widehat{\Phi} \in \mathfrak{L}_{m,d}}{\operatorname{argmin}} \left\{ \frac{1}{D} \sum_{ij} \left(d(X_i, X_j) - d\left(\widehat{\Phi}(X_i), \widehat{\Phi}(X_j)\right)\right)^2 \right\},$$

čia $\mathfrak{L}_{m,d}$ – tiesinės transformacijos iš $\mathbb{R}^m$ į $\mathbb{R}^d$ erdvę, $d(X_i, X_j)$ ir $d\left(\widehat{\Phi}(X_i), \widehat{\Phi}(X_j)\right)$ yra atstumai tarp taškų pradinėje ir sumažintos dimensijos erdvėje, o $D = \sum_{ij}(d(X_i, X_j))^2$.

Tiesiogiai ieškoti Φ esant didelėms n reikšmėms – sudėtinga, tad ieškoma aproksimacija. Iš pradžių iš aibės X atrenkama dalis taškų $X' = \{X'_1, \dots, X'_k\}$, $k \ll n$, k – valdymo taškų skaičius. Taškai vaizduojami sumažintos dimensijos erdvėje $\mathbb{R}^d$. Tegu Y'_i yra taško X'_i projekcija erdvėje $\mathbb{R}^d$. Projekcija Y'_i, minimizuojanti Φ, turėtų tenkinti lygybę:

$$\Phi(X'_i) = \mathrm{Y}'_i, i = 1, \dots, k.$$

Ši lygybė leidžia apskaičiuoti aproksimuojančią Φ transformaciją, kai nagrinėjama kiekvienos matricos Φ' eilutės sandauga su taškais iš X'. Nagrinėkime matricos Φ' pirmos eilutės sandaugą su taškais $X'_i, i = 1, \dots, k$:

$$\phi'_{11}x'_{11} + \dots + \phi'_{1m}x'_{1m} = y'_{11},$$

$$\phi'_{11}x'_{21} + \dots + \phi'_{1m}x'_{2m} = y'_{21},$$

$$\vdots$$

$$\phi'_{11}x'_{k1} + \dots + \phi'_{1m}x'_{km} = y'_{k1},$$

čia $\phi'_{i1}, i = 1, \dots, m$ yra matricos Φ' pirmos eilutės elementai. $x'_{j1}, \dots, x'_{jm}$ yra taško X'_j koordinatės erdvėje $\mathbb{R}^m$ ir y'_{j1} yra pirmoji Y'_j koordinatė erdvėje $\mathbb{R}^d$. Taip sudaroma tiesinių lygčių sistema $L\phi = b$, kur L yra $(k \times m)$ matrica su elementais $x'_{j1}, \dots, x'_{jm}$ j-ojoje eilutėje, ϕ' – transponuota pirmoji matricos Φ' eilutė, b – vektorius, sudarytas iš taškų X'_j pirmosios koordinatės reikšmių, $j = 1, \dots, k$.

Tarkime, kad k yra didesnis už m, tada pirmoji matricos Φ' eilutė ϕ' gali būti aproksimuota šia lygtimi:

$$L^T L\phi' = L^T b.$$

Visa Φ aproksimacija gaunama kartojant nurodytus skaičiavimus su kiekviena matricos Φ' eilute, kurių yra d. Taikant PLMP metodą taškų atvaizdavimo kokybė priklauso nuo taškų $X' = \{X'_1, \dots, X'_k\}$ parinkimo. Šį

metodą sukūrę autoriai valdymo taškams parinkti taiko atsitiktinį taškų parinkimą [21].

Lokali afinioji daugiamatė projekcija. Taikant lokaliosios afiniosios daugiamatės projekcijos metodą (angl. *local affine multidimensional projection,* LAMP), iš pradžių reikia parinkti dalį duomenų aibės taškų, vadinamų valdymo taškais, ir rasti jų vietą mažesnės dimensijos erdvėje $(\mathbb{R}^d, d = 2)$ [20]. Paskui informacija, gauta iš valdymo taškų, naudojama ortogonalių afiniųjų atvaizdavimų šeimai sudaryti po vieną kiekvieno taško projekcijai.

Tegu X_i yra taškas iš duomenų aibės $X = \{X_1, \dots, X_n\}, X \in \mathbb{R}^m$, o X_i' yra i-tasis taškas iš aibės X'. $\mathrm{X}' = \{X_1', \dots, X_k'\}$ – valdymo taškai, atrinkti iš aibės X, o juos atitinkantys taškai sumažintos dimensijos erdvėje žymimi $Y' = \{Y_1', \dots, Y_k'\}$. Tada LAMP metodas tašką X_i atideda mažesnės dimensijos erdvėje, randama geriausia afinioji transformacija $f_X(p) = pM + t$, minimizuojanti funkciją:

$$\sum_i \alpha_i \|f_X(X_i) - Y_i\|^2, \text{ esant apribojimams } M^T M = I,$$

čia matrica M ir vektorius t yra nežinomi. Svoriai α_i randami iš lygties:

$$\alpha_i = \frac{1}{\left\|X_i' - X_i\right\|^2}$$

Radus dalines išvestines pagal t ir jas prilyginus nuliui, t gali būti išreikštas taip:

$$t = \tilde{Y} - \tilde{X}M, \tilde{X} = \frac{\sum_i \alpha_i X_i'}{\alpha}, \tilde{Y} = \frac{\sum_i \alpha_i Y_i'}{\alpha},$$

čia $\alpha = \sum_i \alpha_i$. Taigi dabar minimizavimo uždavinys gali būti užrašytas taip:

$$\min_M \sum_i \alpha_i \left\|\widehat{X_i'}M - \widehat{Y_i'}\right\|^2 \text{ esant apribojimams } M^T M = I,$$

čia $\widehat{X_i'} = X_i' - \tilde{X}$ ir $\widehat{Y_i'} = \mathrm{Y}_i' - \tilde{Y}$. Kaip matrica minimizavimo uždavinys užrašomas taip:

$$\min_M \|AM - B\|_F \text{ esant apribojimams } M^T M = I,$$

kur $\|\cdot\|_F$ – Frobenijaus norma, o matricos A ir B apibrėžiamos taip:

$$A = \begin{bmatrix} \sqrt{\alpha_1}\,\widehat{X_l'} \\ \sqrt{\alpha_2}\,\widehat{X_2'} \\ \vdots \\ \sqrt{\alpha_k}\widehat{X_k'} \end{bmatrix},\ B = \begin{bmatrix} \sqrt{\alpha_1}\widehat{Y_1'} \\ \sqrt{\alpha_2}\widehat{Y_2'} \\ \vdots \\ \sqrt{\alpha_k}\widehat{Y_k'} \end{bmatrix}.$$

Minimizavimo uždavinio sprendinys: $M = UV$, $A^T B = UDV$, kur UDV yra ypatingųjų reikšmių dekompozicija. Radus matricą M, galima rasti taško X_i projekciją Y_i:

$$Y_i = f_X(X_i) = \left(X_i - \tilde{X}\right)M + \tilde{Y}.$$

Dimensijos mažinimas naudojant radialines bazines funkcijas, kai valdymo taškai randami reguliarizuotų ortogonalių mažiausių kvadratų metodu. Šis metodas, paremtas radialinėmis bazinėmis funkcijomis (angl. *radial basis functions*, RBF), projekcijos paieškai pasiūlytas Amorimo ir kt. [39]. Tarkime, turime duomenų aibę $X = \{X_1, \dots, X_n\} \in \mathbb{R}^m$ su n taškų. Tegu $X_k' = \{X_1', \dots, X_k'\} \subset X, k \ll n$, yra valdymo taškų aibė, kurių projekcija $Y_k' = \{Y_1', \dots, Y_k'\} \in \mathbb{R}^d$, $d < m$ yra randama taikant bet kokį pasirinktą dimensijos mažinimo metodą ($d = 2$). RBF metodu randama funkcija $g: R^m \to R^d$, išreiškiama formule:

$$g(X_i) = \sum_{X_i \in X} \lambda_i \phi(\|X_i - X_i'\|). \qquad (2)$$

Funkcija s interpoliuoja kiekvieno valdymo taško poziciją, t. y. $g(X_i) = Y_i$, $i = 1, \dots, k$. Funkcija $\phi'': \mathbb{R}_+ \to \mathbb{R}$ yra vadinama RBF branduoliu (angl. *kernel*). Yra daug funkcijų, kurios gali būti naudojamos kaip RBF branduoliai, daugiau informacijos apie tai pateikiama [39].

Koeficientai λ_i randami taip, kad būtų tenkinama interpoliavimo sąlyga. Sprendžiama tiesinių lygčių sistema su k lygčių $g(X_i') = Y_i', i = 1, \dots, k$. Ši sistema kaip matrica užrašoma taip:

$$\Phi''\lambda = Y', \qquad (3)$$

čia Φ'' interpoliavimo matrica dydžio $k \times k$, $\Phi_{ij}'' = \phi_{ij}''\left(\left\|x_i - x_j\right\|\right)$; Y_k' ir λ yra dvimačiai vektoriai, kiekvienas stulpelis priskiriamas vienai iš projekcijos dimensijų. Tegu $\phi_{ij} = \left(\left\|X_i' - X_j'\right\|\right)$, $\lambda_i = (\lambda_i^1, \lambda_i^2), Y_i' = (y_{i1}, y_{i2})$, tada (3) lygtis gali būti perrašyta:

$$\begin{bmatrix} \phi''_{11} & \cdots & \phi''_{1k} \\ \vdots & \ddots & \vdots \\ \phi''_{k1} & \cdots & \phi''_{kk} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{bmatrix} = \begin{bmatrix} Y'_1 \\ \vdots \\ Y'_k \end{bmatrix}. \qquad (4)$$

Kai randami koeficientai λ, funkcija g yra visiškai apibrėžta ir gali būti naudojama likusiems duomenų aibės taškams aproksimuoti.

Toliau pateikiamas valdymo taškų parinkimas, pagrįstas ortogonaliųjų mažiausių kvadratų metodu. Amorimas ir kiti [39] pasiūlė *ortogonalių mažiausių kvadratų* metodą taikyti valdymo taškams parinkti (angl. *orthogonal least squares method,* OLS). OLS metodas yra naudojamas RBF centrams rasti [40]. Pažymėtina, kad RBF yra tiesinės regresijos modelis. Tarkime, turime N taškų kandidatų $\{X'_i, Y'_i\}, i = 1, \dots, N$, kurie galėtų būti valdymo taškai, kur Y'_i yra valdymo taško X'_i projekcija. Pirmasis žingsnis – atsitiktinai parinkti *N* taškų kandidatų ir rasti jų projekciją sumažintos dimensijos erdvėje. Valdymo taškai pradedami parinkti su tuščia valdymo taškų (regresorių) aibe, o paskui atrenkama po vieną valdymo tašką iš taškų kandidatų aibės. Kiekvienas parinkimas atliekamas taip, kad kvadratinė paklaida $e^T e$ būtų sumažinta. Taikant OLS metodą, transformuojantį ϕ''_i į ortogonalių bazinių vektorių rinkinį, galima apskaičiuoti valdymo taškų individualų įnašą.

Straipsnyje [39] pasiūlytas reguliarizuotų ortogonalių mažiausių kvadratų metodas (angl. *regularized orthogonal least squares method*, ROLS). Kiekvienam taškui kandidatui apskaičiuojama paklaida:

$$E_{\text{error}_i} = \frac{\sum_{i=1}^{k} (W_i^T W_i + \beta) g_i^2}{Y_i'^T Y_i'},$$

čia β yra reguliarizacijos parametras, Y'_i – valdymo taško projekcija, $W = \Phi'' A'$, čia A' yra viršutinio trikampio matrica su vienetais diagonalėje ir $W = [W_1, \dots, W_N]$ su ortogonaliais stulpeliais, tenkinančiais sąlygą, kad $W_i^T W_j = 0$, jei $i \neq j$.

Kiekviename parinkimo žingsnyje taškas X_i, susijęs su vektoriumi W_i ir didžiausia paklaidos E_{error_i} reikšme, įtraukiamas į valdymo taškų aibę. Taip pat apskaičiuojama projekcijos paklaida E_{Stress}, kurios išraiška pateikta formulėje (5). Paklaida E_{error_i} mažiausia, kai visi taškai kandidatai yra

valdymo taškai, tačiau projekcijos paklaida E_{Stress} nebūtinai yra mažiausia (žr. 5 formulę). Tikslas – atrinkti baigtinį geriausiai atspindinčių (paaiškinančių) visą duomenų aibę ir labiausiai sumažinančių projekcijos paklaidą E_{Stress} valdymo taškų skaičių. ROLS algoritmą sukūrę autoriai rekomenduoja, kad šis algoritmas stabdomas turėtų būti tada, kai randamas maksimalus valdymo taškų skaičius arba iteracija su mažiausia projekcijos paklaida E_{Stress}. Detalus valdymo taškų parinkimas taikant ROLS algoritmą pateikiamas darbe [39].

2.3 Projekcijos kokybės nustatymo būdai

Kai sumažinama daugiamačių duomenų dimensija taikant kelis dimensijos mažinimo metodus, reikia įvertinti gautos projekcijos kokybę. Projekcijos kokybės matai atspindi įvairiomias duomenų ypatybes, tad tikslinga projekciją vertinti ne pagal vieną, o pagal kelis matus. Dimensijų mažinimo metodais gautos projekcijos kokybei įvertinti literatūroje siūlomi ir dažnai taikomi šie projekcijos kokybės matai:

- projekcijos paklaida (angl. *Stress function*) [22];
- Spirmeno rho koeficientas (angl. *Spearman's rho*) [41];
- Konigo topologijos išlaikymo matas (angl. *Konig's topology measure)* [41];
- silueto koeficientas (angl. *silhouette*) [42];
- Renyi entropijos koeficientas (angl. *Renyi entropy*) [43].

MDS metodas remiasi normuotąja projekcijos paklaida (E_{Stress}), taip pat gali būti taikoma skirtingais dimensijos mažinimo metodais gautiems rezultatams lyginti. Projekcijos paklaida apskaičiuojama pagal šią formulę [22]:

$$E_{\text{Stress}} = \frac{\sum_{ij}\left(d(X_i,X_j)-d(Y_i,Y_j)\right)^2}{\sum_{ij}\left(d(X_i,X_j)\right)^2}. \qquad (5)$$

Projekcijos paklaida E_{Stress} – tai matas, rodantis, kaip tiksliai išlaikomi atstumai tarp taškų pereinat iš didesnės dimensijos erdvės į mažesnės

dimensijos erdvę. Kuo mažesnė projekcijos paklaida, tuo geriau atstumai išlaikomi.

Straipsnyje [41] projekcijos kokybei įvertinti naudojamas Spirmeno rho koeficientas, apskaičiuojamas pagal šią formulę:

$$\rho = 1 - \frac{6}{(n')^3 - n'} \sum_{k'=1}^{n'} (r'_X(k') - r'_Y(k'))^2, \quad (6)$$

čia r'_X ir r'_Y – atstumų tarp taškų rangai pradinėje ir sumažintos dimensijos erdvėse, n – taškų skaičius, $n' = n(n-1)/2$. Statistikoje Spirmeno rho koeficientas – tai koreliacijos koeficientas, apskaičiuotas ne pačioms kintamųjų reikšmėms, o jų rangams. Spirmeno rho koeficientas taikomas siekiant įvertinti santykinį atstumų išlaikymą tarp taškų, kai pereinama iš m-matės į d-matę erdvę. Spirmeno rho koeficientas – skaičius, priklausantis intervalui $[-1; 1]$. Vertinant projekciją pagal šį koeficientą, geriausiai santykiniai atstumai tarp taškų išlaikomi tada, kai jo reikšmė yra 1.

Konigo topologijos išlaikymo matas pagrįstas atstumų tarp taškų tvarkos išlaikymu m-matėje ir d-matėje erdvėse. Šis matas turi du valdymo parametrus – artimiausių kaimynų skaičiai μ ir ϑ ($\mu < \vartheta$). Artimiausiems kaimynams nustatyti taikomas Euklido atstumas. Konigo topologijos išlaikymas kiekvienam i-ajam taškui ir j-ajam kaimynui apskaičiuojamas pagal šią formulę [41]:

$$E_{KT}^{ij} = \begin{cases} 3, \text{kai } r_X(i,j) = r_Y(i,j) \\ 2, \text{kai } r_X(i,j) = r_Y(i,l), l \in (1, \dots, \mu), i \neq l \\ 1, \text{kai } r_X(i,j) = r_Y(i,t), t \in (\mu+1, \dots, \vartheta), \mu < \vartheta \\ 0, \text{kitais atvejais} \end{cases},$$

čia naudojami šie žymėjimai: $X_{ij}, j = 1, \dots, \mu$, X_{ij} – pradinės erdvės taškai; μ yra m-mačio taško X_i artimiausių kaimynų skaičius, tenkinantys nelygybę $\|X_i - X_{ij_1}\| \leq \|X_i - X_{ij_2}\|, j_1 < j_2$; $Y_{ij}, j = ,1, \dots, \vartheta$, Y_{ij} – sumažintos dimensijos erdvės taškai; ϑ yra yra d-mačio taško Y_i artimiausių kaimynų skaičius; $r_X(i,j)$ yra m-mačio taško X_i j-otojo kaimyno X_{ij} eilės numeris; $r_Y(i,j)$ yra d-mačio taško Y_i j-otojo kaimyno Y_{ij} eilės numeris.

Visas Konigo topologijos išlaikymo matas apskaičiuojamas pagal formulę:

$$E_{\text{KT}} = \frac{1}{3\mu \times m} \sum_{i=1}^{\mu} \sum_{j=1}^{m} E_{\text{KT}}^{ij}. \qquad (7)$$

Konigo topologijos išlaikymo matas taikomas įvertinti kaimyninių taškų eilės numerių pradinėje ir projekcijos erdvėje tvarkai. Koeficientas E_{KT} yra intervale $[0; 1]$. Tiksliausiai kaimynystė išsaugoma, kai koeficientas lygus 1.

Silueto koeficientas pasiūlytas klasterizavimo algoritmų kokybei nustatyti ir leido įvertinti sanglaudą ir atskyrimą tarp suklasterizuotų duomenų aibės taškų [44]. Jis rodo, kaip tiksliai kiekvienas taškas priskirtas klasteriui. Darbe [20] šis koeficientas taikomas projekcijos kokybei įvertinti. Silueto koeficiento vidurkis visai duomenų aibei apskaičiuojamas pagal šią formulę [42]:

$$S = \frac{1}{n} \sum_{X_i \in X} \frac{b_{X_i} - a_{X_i}}{max(a_{X_i})}, \qquad (8)$$

čia n – taškų skaičius aibėje. Taško X_i sanglauda a_{X_i} apskaičiuojama išvedus vidurkį iš taško X_i ir taškų, priklausančių tam pačiam klasteriui, atstumų skirtumų. Atskyrimas b_{X_i} yra mažiausias vidutinis atstumas tarp taško X_i ir taškų, priklausančių kitiems klasteriams. Silueto koeficiento reikšmės yra intervale $[-1; 1]$; kuo didesnė koeficiento S reikšmė, tuo geresnė sanglauda ir atskyrimas. Šioje disertacijoje siūloma silueto koeficientus apskaičiuoti m-mačių ir d-mačių taškų aibėms, paskui vertinti gautų koeficientų skirtumą.

Straipsnyje [43] analizuojant skaitmeninius vaizdus siūloma Euklido atstumų matricai taikyti Renyi entropijos koeficientą, apskaičiuojamą pagal šią formulę:

$$H_\alpha(p) = \frac{1}{1-\alpha} \ln \sum_{i=1}^{n} p_i^\alpha, \qquad (9)$$

čia $\alpha \geq 0$, p_i – tikimybė. Straipsnio [43] autoriai naudoja $\alpha = 2$ reikšmę, tad šioje disertacijoje taip pat pasirinkta ši reikšmė. Entropija yra informacijos teorijoje naudojamas dydis, apibūdinantis vidutinį informacijos kiekį, kurį teikia vienas pranešimas. Renyi entropijos koeficientai apskaičiuojami m-mačių taškų ir sumažintos dimensijos d-mačių taškų duomenų aibėms. Paskui siūloma vertinti šių koeficientų skirtumą, rodantį, kaip pakito informacijos kiekis, kai buvo sumažinta duomenų aibės dimensija.

Šiame poskyryje aprašytų projekcijos kokybės įvertinimo matų charakteristikos pateikiamos 2.1 lentelėje.

2.1 lentelė. Projekcijos kokybės įvertinimo matų charakteristikos

Projekcijos kokybės nustatymo matas	**Žymė-jimas**	**Kam skirtas?**	**Reikšmės**
Projekcijos paklaida	E_{Stress}	Parodo, kaip tiksliai išlaikomi **atstumai** tarp taškų pereinant iš m-matės erdvės į d-matę erdvę.	Geriausiai išlaikomi atstumai tarp taškų tada, kai gaunama kuo mažesnė projekcijos paklaida.
Spirmeno rho koeficientas	ρ	Parodo, kaip išlaikomi **santykiniai atstumai** tarp taškų pereinant iš m-matės erdvės į d-matę erdvę.	Priklauso intervalui $[-1; 1]$. Geriausiai santykiniai atstumai tarp taškų išlaikomi tada, kai ρ koeficiento reikšmė yra 1.
Konigo topologijos išlaikymo matas	E_{KT}	Parodo, kaip išlaikoma **kaimyninių taškų eilės numerių** m-matėje ir d-matėje erdvėje tvarka.	Priklauso intervalui $[0; 1]$. Tiksliausiai kaimynystė išsaugoma, kai koeficientas lygus 1.
Silueto koeficientas	S	Parodo sanglaudą ir atskyrimą tarp suklasterizuotų duomenų aibės taškų. Silueto koeficientų, apskaičiuotų daugiamačiams ir sumažintos dimensijos taškams, skirtumas parodo, ar taškų **klasterizavimo rezultatai** sutampa.	S priklauso intervalui $[-1; 1]$. Geriausia sanglauda ir atskyrimas, kai koeficientas lygus 1.
Renyi entropijos koeficientas	$H_\alpha(p)$	Parodo informacijos kiekį duomenyse. Renyi entropijos koeficientų, apskaičiuotų daugiamačiams ir sumažintos dimensijos taškams, skirtumas parodo, **kaip pakito** informacijos kiekis duomenyse.	–

Paminėtina, kad dauguma iš šių matų reikšmėms skaičiuoti remiasi atstumais tarp taškų. Nagrinėjant didelės apimties duomenų aibes kyla projekcijos įvertinimo problema, mat skaičiavimuose naudojamos didelės apimties atstumų matricos, kurioms apskaičiuoti gali pritrūkti personalinio kompiuterio operatyviosios atminties [45].

2.4 Duomenų tyrybos sistemos dimensijai mažinti

Šiame poskyryje aprašyto tyrimo tikslas – nustatyti, kokių apimčių duomenis per priimtiną laiką gali ištirti populiarios duomenų tyrybos sistemos, sprendžiančios dimensijos mažinimo uždavinį. Nagrinėjama dimensijos mažinimo algoritmų greitaveika taikant skirtingos apimties duomenų aibes.

Darbe nagrinėjamos ir lyginamos trys atvirojo kodo duomenų tyrybos sistemos:

- WEKA (*Waikato Environment for Knowledge Analysis*) [46];
- KNIME (*Konstanz Information Miner*) [47];
- ORANGE (*Data Mining Fruitful and Fun*) [48].

Tai vienos iš populiariausių duomenų tyrybos sistemų. Šios sistemos realiems uždaviniams spręsti taikomos darbuose [49], [50], [51]. Dėl savo nesunkiai suvokiamų principų šios duomenų tyrybos sistemos tapo populiarios įvairiose srityse. Taip pat būtent dėl šių priežasčių minėtos sistemos pasirinktos šiai analizei. Šiose sistemose realizuotų klasifikavimo algoritmų teisingas duomenų klasifikavimas tiriamas darbe [52], sistemų analizė atlikta darbe [53], tačiau nėra nustatyta, kokių apimčių duomenis sistemos pajėgios apdoroti ir analizuoti. Atvirojo kodo duomenų tyrybos sistemų taikymo sritys, vartotojų grupės, realizuoti algoritmai, vizualizavimo būdai ir kitos ypatybės vertinamos darbe [54], bet analizė, taikant įvairias duomenų aibes, nėra atlikta. Darbe [55] autoriai teigia, kad WEKA, KNIME, ORANGE sistemos veikia su vidutinio dydžio duomenų aibėmis, tačiau nenurodyta, kokie duomenys įvardijami kaip vidutinio dydžio.

Šiame tyrime atvirojo kodo sistemos lyginamos su MATLAB – specialios paskirties kompiuterine programa, skirta automatizuoti mokslinius skaičiavimus. Tyrime skaičiavimai atlikti naudojant MATLAB.

Eksperimentiniams tyrimams pasirinkti du klasikiniai dimensijos mažinimo metodai: pagrindinių komponenčių analizė (PCA) [25], [26] ir daugiamačių skalių metodas (MDS) [22].

Eksperimentai atlikti kompiuteriu, kurio pagrindinės ypatybės yra šios: operatyvioji atmintis (RAM) 12 GB, procesorius Intel i5-3317U, šio taktinis dažnis 1,7 GHz (Max Turbo dažnis 2,6 GHz). Šiame kompiuteryje veikia Windows 8 operacinė sistema.

Eksperimentiniu tyrimu siekiama išnagrinėti sistemų galimybes analizuoti įvairaus dydžio duomenis ir nustatyti, kokių apimčių duomenų analizė negalima šiomis sistemomis. Šiam tikslui naudotos ne etaloninės duomenų aibės, skirtos duomenų tyrybos algoritmams vertinti, o dirbtinai sugeneruotos įvairių apimčių duomenų aibės, kurių požymių reikšmės tolygiai pasiskirsčiusios intervale (0; 1). Dirbtinai generuojant duomenų aibes galima rinktis norimos apimties požymių ir objektų skaičių. Analizuojant MDS metodą požymių skaičius parinktas – 20 ir 100, o objektų skaičius – 1000, 5000, 10000, 15000 ir 20000. PCA analizei parinktas požymių skaičius – 20 ir 100, objektų – 100000–5000000. Jei pasirenkamas kitas požymių ir / ar objektų skaičius, rezultatų skaitinių išraiškų absoliutūs dydžiai pakistų, tačiau išliktų toks pat santykis tarp skirtingomis sistemomis gautų rezultatų. Atliekant tyrimą pasirinkta projekcinės erdvės dimensija yra $d = 2$. Tyrime nagrinėjami projekcijos metodai turi tam tikrus valdymo parametrus ir nustatymus, pateikiamus 2.2 lentelėje. Joje pateikiamos ir apibendrintos metodų pagrindinės ypatybės. Tyrime lyginamos dvi daugiamates skales įgyvendinančios funkcijos, viena iš jų yra standartinė MATLAB funkcija *mdscale,* kita funkcija vadinama *smacof* [22], ši įgyvendinta ir taikyta straipsnyje [56]. Šiame tyrime naudojama Sammono paklaida (angl. *Sammon's Stress*) E_S. Tai matas, rodantis, kaip tiksliai išlaikomi atstumai tarp taškų

pereinant iš didesnio skaičiaus matmenų erdvės į mažesnio skaičiaus matmenų erdvę [57].

2.2 lentelė. Metodų valdymo parametrai ir kiti nustatymai

Metodas	**Pagrindinės metodo ypatybės**	**Sistema**	**Valdymo parametrai ir kiti nustatymai**
PCA	Vertinamos dispersijos.	MATLAB	Pagrindinių komponenčių skaičius – 2.
		KNIME	
		WEKA	
		ORANGE	
MDS	Vertinami visų taškų tarpusavio artimumai. Paprasčiausias variantas – Euklido atstumas.	MATLAB (funkcija *smacof*)	SMACOF algoritmas sukurtas naudojant MATLAB ciklą *for*, 100 iteracijų, pradinės taškų reikšmės parenkamos atsitiktinai iš intervalo $(0,1)$.
		MATLAB (funkcija *mdscale*)	Minimizuojama Sammono paklaida, standartinė MATLAB funkcija *mdscale*, 200 iteracijų, tolerancija 10^{-4}, pradinis taškų parinkimas – atsitiktinis.
		KNIME	Minimizuojama Sammono paklaida, 20 epochų, pradinis taškų parinkimas – atsitiktinis.
		WEKA	Sistemoje nėra šio metodo.
		ORANGE	Sammono paklaida, maksimalus žingsnių skaičius – 5000, tolerancija 10^{-3}, pradinis taškų parinkimas – atsitiktinis.

Kai dimensija mažinama MDS metodu, analizuojant įvairios apimties duomenų aibes, greičiausiai skaičiavimo rezultatai gaunami MATLAB programa su *mdscale* funkcija, tačiau šiai sistemai negana kompiuterio operatyviosios atminties analizuojant 20000 objektų aibę. ORANGE sistema skaičiavimas trunka vidutiniškai 1,5 karto ilgiau už skaičiavimus MATLAB programa su *mdscale* funkcija, be to, ORANGE sistema nėra pajėgi apdoroti 10000 objektų ir didesnių duomenų aibių. KNIME sistema ir MATLAB

programa su funkcija *smacof* sumažina dimensiją duomenų aibei, sudarytai iš 10000 objektų ir 20 požymių, tačiau tai trunka apie 7 val. Dėl ilgo skaičiavimo laiko šios sistemos su dar didesnėmis duomenų aibėmis nebuvo išbandytos. Gauti eksperimento rezultatai leidžia teigti, kad nagrinėjamos sistemos KINIME ir ORANGE nėra pajėgios apdoroti didelės apimties duomenų aibių, taikant MDS metodą, t. y. pritrūksta kompiuterio operatyviosios atminties arba skaičiavimo laikas yra per ilgas (daugiau kaip 7 val.).

Kai dimensija mažinama PCA metodu analizuojant 100000 objektų ir 100 požymių duomenų aibę ir duomenų aibę, sudarytą iš 250000 objektų ir 20 požymių, taikant visas sistemas trunkama iki minutės. WEKA ir ORANGE sistemoms nepakanka kompiuterio operatyviosios atminties analizuojant duomenų aibę, sudarytą iš 500000 objektų ir 100 požymių. MATLAB programai negana kompiuterio operatyviosios atminties analizuojant duomenų aibę, sudarytą iš 3000000 objektų ir 100 požymių. Tačiau kai požymių skaičius 20, taikant MATLAB programą dimensija sumažinama mažiau kaip per minutę analizuojant 3000000–5000000 objektų duomenų aibes. O KNIME sistema gali apdoroti 3000000–5000000 objektų ir 100 požymių duomenų aibes. Pavyzdžiui, duomenų aibės iš 5000000 objektų ir 100 požymių dimensija sumažinama per 14 min. KNIME sistema. Galima teigti, kad MATLAB ir KNIME sistemos gali apdoroti gana didelės apimties duomenų aibes, kai taikomas PCA metodas. O ORANGE ir WEKA sistemos nėra tinkamos didelės apimties duomenų aibėms.

2.5 Technologijos, skirtos didiesiems duomenims apdoroti

Naujai sukuriamų ir surenkamų duomenų kiekis auga intensyviai. Sparčiai vystantis internetinėms technologijoms, duomenų kaupimo ir saugojimo galimybėms, didelės apimties duomenų daugėja visose mokslo ir inžinerijos srityse [58]. Paskirstytaisiais ir lygiagrečiaisiais skaičiavimais apdorojami didelės apimties duomenys. Kai duomenims apdoroti nepakanka įprastų sistemų, galima taikyti debesų kompiuteriją (angl. *cloud computing*).

Skaičiavimo užduotys perkeliamos, taip taupomos informacinių technologijų išlaidos ir ištekliai [59]. Debesų kompiuterija – paradigma, kai per tinklo prieigą galima naudotis visais kompiuteriniais ištekliais, o jie valdomi su minimaliu paslaugų tiekėjo įsikišimu.

Skaičiavimams gali būti taikoma atvirojo kodo programinė įranga *Hadoop* (https://hadoop.apache.org/), skirta didelės apimties duomenims apdoroti ir analizuoti. Viena iš pagrindinių *Hadoop* dalių – paskirstyta failų sistema HDFS (angl. *Hadoop Distributed File System*). Šia sistema duomenys dalijami į mažesnes vienodas dalis (blokus), tolygiai paskirstomas po kompiuterio klasterio kompiuterius. *Hadoop MapReduce* – paskirstytų didelės apimties duomenų apdorojimo programinės įrangos karkasas, leidžiantis atlikti paskirstytuosius skaičiavimus ir lygiagretinimą kompiuterių klasteriuose [60]. Įvairioms duomenų tyrybos sritims sukurta kartu su *Hadoop* veikianti *Mahout* biblioteka, šioje įgyvendinti populiarūs klasterizavimo, klasifikavimo, dimensijos mažinimo algoritmai. Tačiau šių algoritmų nėra daug ir dėl *MapReduce* specifikos ne visada lengvai ir efektyviai galima taikyti esamus duomenų tyrybos algoritmus [61]. *Apache Spark* (https://spark.apache.org/) yra didelės spartos atviro kodo programinė įranga paskirstytos atminties lygiagretiems skaičiavimams atlikti su didelės apimties duomenimis [62], [63]. Darbuose [16], [64] analizuojamos atvirojo kodo kompiuterio mokymo (angl. *machine learning*) bibliotekos (Mahout, MLlib, H_2O, SAMOA), skirtos darbui su didelės apimties duomenų aibėmis. Darbe [65] pasiūlytas holistinis būdas, efektyvus paskirstytam dimensijos mažinimui, kai dirbama su didžiaisiais duomenimis. Kiekvienas įrankis turi savo privalumų ir trūkumų, taigi sprendžiant konkretų uždavinį reikia pasirinkti tinkamą įrankį.

Šiuo metu sukurta nemažai komercinių įrankių, skirtų didiesiems duomenims apdoroti, tačiau kyla problema, kad nedaug iš jų gali taikyti dimensijos mažinimo metodus. *Microsoft Azure* (https://azure.microsoft.com/en-us/) įrankis realizuoja klasifikavimo, klasterizavimo, regresijos, dimensijos mažinimo algoritmus, vienas iš jų – pagrindinių komponenčių analizė. *Tableau* programinis įrankis

(http://www.tableau.com) leidžia bet kokio dydžio duomenis vizualizuoti vartotojui patrauklia forma. *Datameer* (http://www.datameer.com) produktas skirtas didelės apimties duomenų integracijai / paruošimui, analizei ir vizualizavimui. *Datameer Smart Analytics* leidžia pasirinkti šiuos analizės algoritmus: klasterizavimo, sprendimų medžių, stulpelių priklausomybės ir rekomendacijų, t. y. iš istorinių duomenų pateikiama prognozė. *Zementis Universal* PMML (angl. *Predictive model markup language*) papildinys leidžia integruoti įvairius statistinių paketų, tokių kaip SAS, SPSS, R, KNIME ir kt., prognozavimo modelius. *Pentaho* (http://www.pentaho.com) įrankis skirtas didelės apimties duomenis išskleisti, parengti, analizuoti ir vizualizuoti. SQL serverio analizės paslaugos (angl. *SQL Server Analysis Services* (*SASS*) (https://msdn.microsoft.com/en-us/library/bb510516.aspx) turi nemažai įdiegtų duomenų tyrybos algoritmų: klasifikavimo, klasterizavimo, sprendimų medžio, regresijos, dirbtinių neuroninių tinklų ir kt.

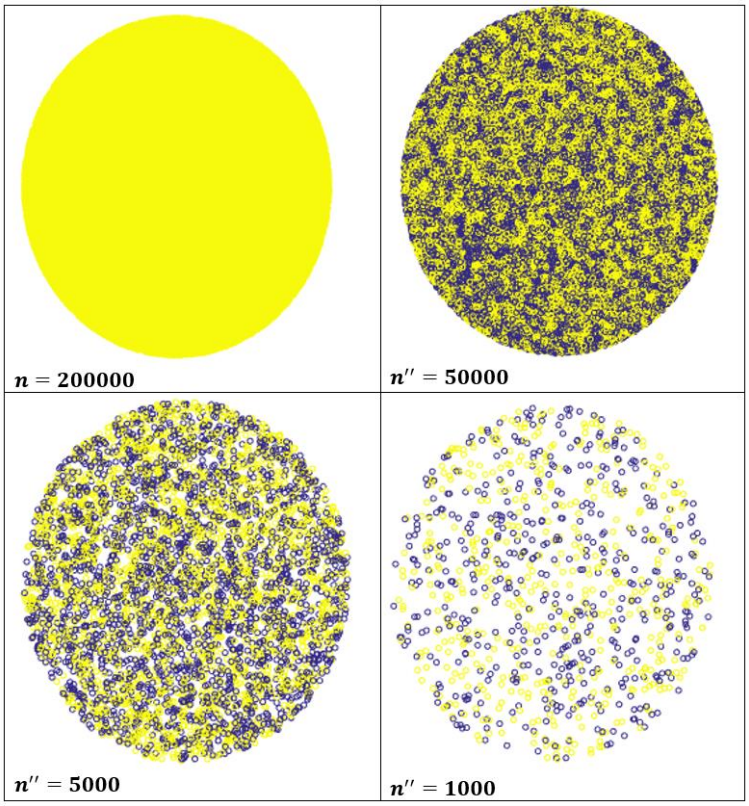
WEKA programa šiuo metu turi įrankių (angl. *distributedWekaBase*) lygiagretiems skaičiavimams *Hadoop* karkase, tačiau dauguma iš WEKA programos klasifikavimo ir regresijos algoritmų, nors ir gali būti taikomi Hadoop karkase, negali būti išlygiagretinti. Šie algoritmai apmokyti kaip ansambliai (angl. *ensemble*): mažesni duomenų aibės poaibiai apmokomi individualiai, tačiau užuot sujungti į galutinį modelį mažinimo etapu, jie sujungiami taikant balsavimo technikas (angl. *voting techniques*) [16]. Ansamblių teorija paremti algoritmai jungia kelis metodus, taip gaunami tikslesni rezultatai.

Kaip matyti, nedaugelis produktų, skirtų didelės apimties duomenims apdoroti, turi integruotus klasikinius gerai žinomus dimensijos mažinimo metodus. Be to, šioms minėtoms technologijoms reikalingas specifinis išmanymas, taip pat nėra vartotojui draugiškos aplinkos paprastai valdyti lygiagrečiuosius ir paskirstytuosius skaičiavimus, ypač kai kalbama apie duomenų dimensijos mažinimą.

2.6 Duomenų vizualizavimo įrankiai

Didelės apimties duomenų vizualizavimo įrankiai naudojami palengvinti žinių gavybos procesui ir atitinkamos srities įžvalgoms daryti. Dauguma iš duomenų aibių nusakomos daugiau kaip dviem ar trimis požymiais, o tokius duomenis vizualizuoti daug sudėtingiau nei dvimačius ar trimačius duomenis. Vizualizuoti daugiamačius duomenis galima dviem būdais: naudoti tiesioginį vizualizavimą arba vizualizuoti duomenų aibės projekciją. Tiesioginio vizualizavimo metodais kiekvienas daugiamačio objekto požymis pateikiamas tam tikra vizualia forma. Tiesioginio vizualizavimo pavyzdžiai – taškiniai grafikai, apžiūros grafikai, Andrews kreivės, lygiagrečios koordinatės, spindulinis vizualizavimas ir kt. [66]. Projekcijos metodais daugiamačius duomenų objektus atitinkantys vektoriai pateikti mažesnio skaičiaus matmenų erdvėje. Pastaruoju metu sukurta daug įrankių duomenims vizualizuoti. Darbe [67] pateikiama lyginamoji *Tableau*, *Spotfire*, *QlikView*, *JMP (SAS)*, *Cognos (IBM)*, *SQL Server BI (Microsoft)*, *Business Objects (SAP)*, *Teradata*, *PowerPivot (Microsoft)* ir kitų sistemų analizė. Tik trys iš šių sistemų turi įdiegtus dimensijos mažinimo metodus (pagrindinių komponenčių, daugiamačių skalių ir neuroninių tinklų). Kai vizualizuojamų duomenų labai daug, o norima pamatyti lokalią taškų struktūrą, gali būti taikomas vaizdo priartinimas ir išdidinimas [68]. Sklaidos diagrama – paprastas intuityvus būdas vizualizuoti dvimačius duomenis. Ji gali parodyti ryšį, koreliaciją tarp dviejų dimensijų. Suprantama, kad sumažinus didelės apimties duomenų aibės daugiamačių taškų dimensiją iki vaizdo dimensijos $d = 2$, o dvimačius taškus atvaizdavus sklaidos diagramoje, šių negalėsime atskirti ir identifikuoti. Rodyti per daug taškų beprasmiška, nes tai tik apsunkina duomenų suvokimą. Persidengiančių taškų problema pavaizduota 2.1 paveiksle. Iš pradžių atvaizduojama visa duomenų aibė, sudaryta iš 200000 taškų. Vaizde matyti skritulys, jį sudarantys taškai pažymėti geltona spalva. Iš pradinės duomenų aibės atsitiktinai parinkus ir atvaizdavus 50000 taškų pastebėtina, kad po geltona spalva pažymėtais taškais yra kitos klasės taškai, pažymėti mėlyna

spalva. Atvaizdavus 5000 taškų dar esti persidengiančių taškų, o atvaizdavus imtį, sudarytą iš 1000 taškų, matyti, kad persidengiančių taškų labai sumažėja. Taigi būtina ieškoti būdų spręsti šią problemą ir vizualizuoti duomenis be persidengimo. Persidengiančių taškų problema vizualizuojant duomenis taip pat sprendžiama ir darbuose [69], [70], [71], [72].



2.1 pav. Persidengiančių taškų vizualizavimo pavyzdys (kiekvieno paveikslėlio kairiajame kampe pateikiama: n – duomenų aibės dydis arba n'' – duomenų aibės imties dydis)

Vienas iš plačiausiai paplitusių būdų, susijęs su tankiai išsidėsčiusiais duomenimis, yra tankio įverčių naudojimas [71]. Deja, šis metodas dažniausiai neįtraukia išskirčių, kai nedidelis taškų kiekis su mažu tankiu yra nutolęs nuo likusiųjų duomenų aibės taškų. Kai dirbama su persidengiančiais taškais, siūloma įvesti papildomą dimensiją (dvimačiu atveju trečiąją) ir joje atvaizduoti taškų tankumą ar taškų skaičių [70], tačiau šis metodas nėra tinkamas [71] didelės apimties duomenų aibėms. Apibendrintoje sklaidos diagramoje (angl. *generalized scatter plot*) siūloma atsižvelgti į taškų tankumą. Tankiau išsidėsčiusiems taškams skiriama daugiau erdvės, o rečiau

išsidėsčiusiems taškams – mažiau [69]. Nors šiuo būdu pateikiami labiau matomi tankiai išsidėstę išsidėstę taškai ir neprarandamos išskirtys, tačiau iš atvaizduotų duomenų sunku įžvelgti, kurie taškai buvo tankiai išsidėstę, o kurie ne. Yra tyrimų, nagrinėjančių sklaidos diagramos suvokimą pagal tam tikras ypatybes, tokias kaip simbolio dydžio atskyrimas [73], šviesumas [74], simbolio kontrastas [75]. Nors tai nėra tiesiogiai susiję su persidengiančių taškų suvokimu, tačiau norint visiškai suvokti sklaidos diagramas reikėtų atsižvelgti į minėtus aspektus [71]. Yra nemažai ir atviro kodo ir komercinių duomenų vizualizavimo įrankių, tačiau daugelis iš jų neturi dimensijos mažinimo algoritmų, be to, atvaizduojami tik keletas daugiamačių taškų požymių arba agreguoti duomenys (susumuoti, suvidurkinti, minimalios / maksimalios reikšmės) [76], [77], [78]. Duomenims vizualizuoti gali būti naudojami šie įprasti grafikų tipai: sklaidos diagrama, linijinė diagrama, stulpelinė diagrama, skritulinė diagrama, histograma ir kt. [79]. Nedaugelis vizualizavimo įrankių yra pritaikyti prasmingai ir kokybiškai atvaizduoti informaciją [80].

Literatūros analizė rodo, kad taškų persidengimo problema yra aktuali ir nėra tinkamo vizualizavimo būdo atvaizduoti taškus be persidengimo, kuris padėtų suvokti didelės apimties duomenis ir leistų identifikuoti kiekvieno taško vietą (poziciją) tarp kitų duomenų aibės taškų.

2.7 Antrojo skyriaus apibendrinimas

Šiame skyriuje pateiktos didžiųjų duomenų ir didelės apimties duomenų apibrėžtys. Taip pat atlikta dimensijos mažinimo metodų apžvalga. Aprašyti dažniausiai taikomi dimensijos mažinimo metodai ir metodai, paremti valdymo taškais. Atlikta projekcijos kokybės įvertinimo matų, atspindinčių įvairias duomenų ypatybes, analitinė apžvalga. Dauguma iš nagrinėjamų matų reikšmėms skaičiuoti naudoja atstumus tarp taškų. Nagrinėjant didelės apimties duomenų aibes kyla projekcijos paklaidos įvertinimo problema, kadangi jos skaičiavimuose naudojamos didelės apimties atstumų matricomos, o šioms apskaičiuoti gali nepakakti kompiuterio operatyviosios atminties. Taigi reikia

tinkamo būdo apskaičiuoti projekcijos paklaidą, kai dirbama su didelės apimties duomenų aibėmis.

Taip pat šiame skyriuje apžvelgtos duomenų tyrybos sistemos. Nustatyta, kokių apimčių duomenis per ne per ilgą laiką geba apdoroti populiarios duomenų tyrybos sistemos, kai sprendžiamas dimensijos mažinimo uždavinys. Atlikta didžiųjų duomenų apdorojimo technologijų ir vizualizavimo apžvalga parodė, kad nėra būdo atvaizduoti didelės apimties duomenų aibės taškus be persidengimo sklaidos diagramoje ir išlaikyti bendrą duomenų struktūrą.

3 Projekcijos paklaidos apskaičiavimas ir strategija didelės apimties duomenims vizualizuoti

Šiame skyriuje pasiūlytas projekcijos paklaidos apskaičiavimo būdas didelės apimties duomenų aibėms reikalauja mažiau kompiuterio operatyviosios atminties ir trumpesnio skaičiavimo laiko. Taip pat pasiūlyta vizualizavimo strategija leidžia vizualizuoti didelės apimties duomenų aibes, išvengti duomenų aibės taškų persidengimo ir išlaikyti duomenų struktūrą.

Visi šioje disertacijoje pasiūlyti ir sudaryti algoritmai realizuoti MATLAB aplinkoje. Ši programa pasirinkta dėl daug joje realizuotų dimensijos mažinimo algoritmų ir kitų duomenų analizės metodų. Sukurtas papildomas dimensijos mažinimo metodų įrankis MATLAB *Toolbox for Dimensionality Reduction* (https://lvdmaaten.github.io/drtoolbox/). Be to, viena iš pagrindinių MATLAB ypatybių yra ta, kad programa naudoja procesoriui optimizuotas bibliotekas greitiems veiksmų su matricomis ir vektoriais skaičiavimams atlikti. Tai ypač svarbu, kai dirbama su dimensijos mažinimo metodais, kuriose naudojamos atstumų matricos. Disertacijoje pateikti siūlymai gali būti pritaikyti ir kitoms programavimo kalboms su analogiškomis funkcijomis.

Skyriuje pateikti rezultatai publikuoti autorės darbuose [A2], [B3], [C1], [C2]. Eksperimentinių tyrimų rezultatai pateikti 4 skyriuje.

3.1 Projekcijos paklaidos apskaičiavimas

Šiame poskyryje pasiūlyti projekcijos paklaidos apskaičiavimo būdai didelės apimties duomenų aibėms. Projekcijos paklaida dažniausiai taikoma tyrimuose dimensijų mažinimo metodų rezultatams vertinti.

Skirtingų duomenų rezultatams lyginti literatūroje dažnai naudojama MDS metodo normuotoji projekcijos paklaida [20], [21], [22]. Šioje disertacijoje dimensijos mažinimo metodu gautai projekcijai įvertinti nagrinėjama projekcijos paklaida (angl. *Stress function*), apskaičiuojama pagal (5) formulę.

Yra keletas būdų, kaip skaičiuoti projekcijos paklaidą MATLAB aplinkoje:

- skaičiuojant projekcijos paklaidą naudoti ciklą *for*, jame paklaida skaičiuojama panariui kiekvienam duomenų aibės taškui pagal (5) formulę;
- arba taikyti programos MATLAB ciklą *pdist* pradinei ir sumažintos dimensijos duomenų aibėms, o paskui pritaikyti (5) formulę.

Funkcija *pdist* apskaičiuoja Euklido atstumus tarp visų objektų porų iš matricos X dydžio $n \times m$. Matricos X eilutės atitinka objektus, o stulpeliai – požymius. Funkcijos rezultatas – atstumų matrica Δ, kurios elementas (i, j) (kur $i < j$) atitinka atstumą tarp duomenų aibės objektų i ir j. Papildoma funkcija *pdist2* apskaičiuoja Euklido atstumus tarp kiekvienos objektų poros iš matricų X (dydžio $n \times m$) ir Z (dydžio $n \times m$). Matricų X ir Z eilutės atitinka objektus, o stulpeliai – požymius. Funkcijos rezultatas – atstumų matrica Δ dydžio $n \times n$, šios elementas (i, j) yra lygus atstumui tarp objekto i iš matricos X ir objekto j iš matricos Z.

Iš didelės apimties duomenų aibių nustatyta, kad pirmuoju būdu skaičiavimai trunka ilgai, pavyzdžiui, 250000 objektų ir 3 požymių duomenų aibės skaičiavimams prireikia apie 2 valandų personaliniu kompiuteriu, kurio parametrai nurodyti 4.1 poskyryje. Antruoju – skaičiavimai atliekami labai sparčiai, tačiau nagrinėjant didesnes negu 30000 objektų duomenų aibes nepakanka esamos 12 GB kompiuterio operatyviosios atminties.

Šioje disertacijoje siūlomi du projekcijos paklaidos apskaičiavimo būdai didelės apimties duomenims:

- *1-asis būdas*: projekcijos paklaidą vertinti ne visai duomenų aibei, o tik jos imčiai;
- *2-asis būdas*: projekcijos paklaidą skaičiuoti visai duomenų aibei, tačiau duomenų aibę skaičiavimų metu padalyti į dalis.

3.1.1 Projekcijos paklaida duomenų aibės imčiai

Pirmasis šiame darbe siūlomas būdas projekcijos paklaidai apskaičiuoti grindžiamas sudaryta duomenų aibės imtimi. Statistikoje dažnai tiriama ne visa

populiacija, o tik jos dalis – imtis. Iš sudarytos tinkamos imties galima daryti patikimas išvadas apie visą populiaciją. Siūloma tai pritaikyti ir projekcijos paklaidai vertinti. Duomenų aibės imtis gali būti randama šiais populiariais imties sudarymo būdais: vienas iš jų yra sudaryti atsitiktinę imtį, kitas – duomenų aibę suskirstyti į sluoksnius (stratus), suskaičiuoti, kiek taškų priskirta kiekvienam sluoksniui, o tada iš kiekvieno sluoksnio atrinkti proporcingą dalį taškų imčiai sudaryti.

Apskaičiuojama sudarytos duomenų aibės imties projekcijos paklaida pagal (5) formulę taikant programos MATLAB funkciją *pdist* pradinei ir sumažintos dimensijos duomenų aibėms.

3.1.2 Duomenų aibės dalijimas į dalis

Antruoju siūlomu būdu projekcijos paklaida apskaičiuojama dalijant duomenų aibę į dalis. Šis būdas sudarytas iš šių žingsnių:

1 žingsnis: pradinė ir sumažintos dimensijos duomenų aibės padalijamos į atitinkamai mažesnes dalis;

2 žingsnis: kiekvienai duomenų aibės daliai taikant MATLAB funkciją *pdist* apskaičiuojami Euklido atstumai;

3 žingsnis: kiekvienai duomenų aibės daliai apskaičiuojamas (5) formulės skaitiklis ir vardiklis;

4 žingsnis: visų galimų aibių dalių porų kombinacijų atstumai skaičiuojami taikant *pdist2* funkciją;

5 žingsnis: visoms galimoms poroms apskaičiuojamas (5) formulės skaitiklis ir vardiklis;

6 žingsnis: paklaidos funkcija apskaičiuojama susumavus skaitiklius, gautus *3* ir *5 žingsniuose*, ir padalinus iš vardiklių, gautų *3* ir *5 žingsniuose*, sumos.

3.1. paveiksle pateiktas projekcijos paklaidos apskaičiavimo duomenų aibę dalijant į dalis algoritmo pseudokodas.

```
Input: data – pradinė duomenų aibė (daugiamačiai taškai);
       proj – duomenų aibės projekcija (sumažintos dimensijos taškai);
       A – matrica, sudaryta iš dviejų stulpelių (pirmojo (antrojo) elementai
            nurodo duomenų aibės taško indeksą) atitinkančius duomenų
aibės         dalies pradžią (pabaigą);
       groups – duomenų aibės dalių skaičius.
Output: Stress – projekcijos paklaida.
BEGIN
//Kiekvienai duomenų aibės daliai
FOR i=1:groups
      data_temp=pdist(data(A(i,1):A(i,2),:))
      proj_temp=pdist(proj(A(i,1):A(i,2),:))
      nomin_temp(i)=sum((data_temp-proj_temp).^2)
      denom_temp(i)=sum(data_temp.^2)
END
//Taškams iš dviejų duomenų aibės dalių
numerator=0; denominator=0
FOR i=1:groups
FOR j=i+1:groups
      data=(pdist2(data(A(i,1):A(i,2),:),data(A(j,1):A(j,2),:)))
      proj=(pdist2(proj(A(i,1):A(i,2),:),proj(A(j,1):A(j,2),:)))
      numerator= numerator+sum(sum((data-proj).^2))
      denominator=denominator+sum(sum(data.^2))
END
END
//Apskaičiuojama projekcijos paklaida
Stress=(numerator+sum(nomer_temp))/(denominator+sum(denom_temp))
END
```

3.1 pav. Pseudokodas, kai projekcijos paklaida apskaičiuojama duomenų aibę dalijant į dalis

Dalijant duomenų aibę į dalis skaičiavimai atliekami mažesnės apimties duomenų aibėms, taip skaičiavimams atlikti programai pakaks kompiuterio operatyviosios atminties. Be to, kiekvienos dalies atstumams tarp taškų skaičiuoti naudojamos funkcijos *pdist*, *pdist2*, šių greitaveika yra didelė palyginus su ciklo naudojimu atstumų matricoms rasti. Pasiūlyti projekcijos apskaičiavimo būdai eksperimentiškai ištirti ir gauti rezultatai pateikti 4.4 poskyryje.

3.2 Pasiūlytas būdas duomenims vizualizuoti

Didelės apimties duomenų aibių analizei reikia būdo, kaip parodyti esminę duomenų informaciją. Nėra būtina atvaizduoti visų duomenų aibės taškų, nes šie persidengs ir vaizdas bus neinformatyvus. Pats paprasčiausias būdas – iš

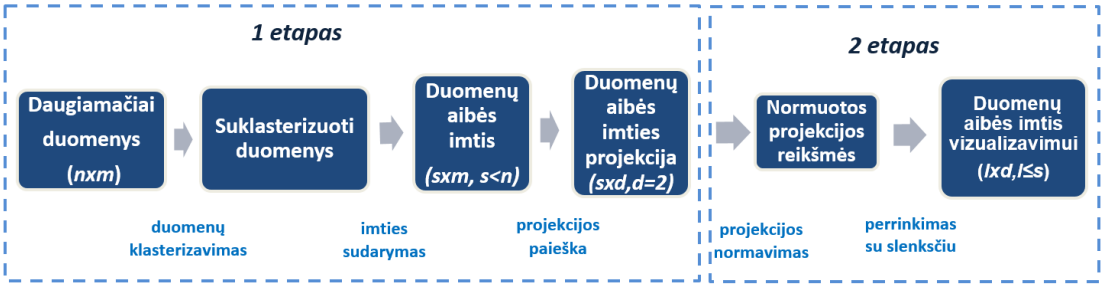
pradžių suklasterizuoti duomenis, o paskui imti klasterių atstovus, tačiau taip gali būti prarandamos išskirtys (angl. *outliers*) arba retų klasterių atstovai ir duomenų struktūra nebus išlaikyta, o vaizdas – neinformatyvus. Išskirtys gali turėti svarbios informacijos, tad vizualizuojant svarbu neprarasti tokių taškų. Vizualizuojant didelį taškų kiekį sklaidos diagramoje dažniausiai gaunamas neinformatyvus vaizdas, t. y. taškai susitelkia ir vienas kitą perdengia. Tampa sudėtinga įžvelgti prasmingos informacijos, kai duomenų skaičius ypač didelis. Nebūtina vizualizuoti milijono taškų, jei galima naudojant imtį, sudarytą iš duomenų aibės taškų, atspindėti pagrindines duomenų aibės ypatybes. Kita problema yra ta, kad ne visi dimensijos mažinimo metodai tinkami didelės apimties duomenų aibėms. Pagrindinis MDS trūkumas yra tas, kad reikia didelių kompiuterio operatyviosios atminties resursų, be to, skaičiavimai trunka ilgai [45], [81]. Taigi šioje disertacijoje siūloma vizualizuoti ne visą duomenų aibę, bet tik jos imtį. Daugiamačių taškų imties duomenų aibės projekciją galima rasti greitai, o paskui jos dvimačius taškus vizualizuoti sklaidos diagramoje.

Šiame darbe pasiūlytą duomenų vizualizavimo strategiją sudaro šie pagrindiniai etapai (žr. 3.2. paveikslą):

1 etapas. Duomenų aibės imties sudarymas ir imties projekcijos radimas.

2 etapas. Imties projekcijos taškų vizualizavimas be taškų persidengimo.

Vizualizavimo strategija pateikiama 3.2 paveiksle. Tarkime, turime daugiamačių taškų duomenų aibę $X \in \mathbb{R}^m$ su n taškų. Tegu $X_S = \{x_1, \dots, x_s\} \subset X, s < n$ yra duomenų aibės imtis. Sumažintos dimensijos imties taškai $Y_S = \{y_1, \dots, y_s\} \in \mathbb{R}^d, s < n,\ d = 2$ randami taikant pasirinktą dimensijos mažinimo metodą. Vizualizavimo imtis yra $Y_L = \{y_1, \dots, y_l\} \subset Y_S, l \leq s < n$. Vizualizuojama imtis Y_L sudaryta iš tokio paties ar mažesnio taškų kiekio už imties projekciją Y_S, kadangi 2 etape pašalinamas taškų persidengimas.

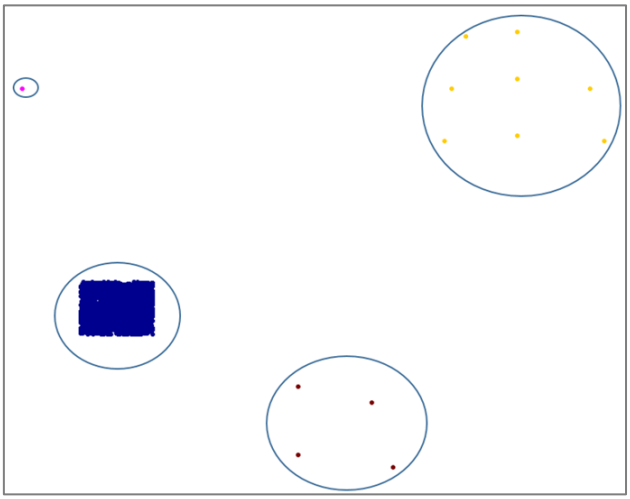


3.2 pav. Vizualizavimo strategija analizuojant didelės apimties duomenų aibes

3.2.1 Duomenų aibės imties sudarymas (1 etapas)

Svarbus uždavinys – tinkamai sudaryti duomenų aibės imtį. Pats paprasčiausias būdas – suklasterizuoti duomenis ir tada išrinkti atstovus iš kiekvieno klasterio, tačiau gali būti neatskleista retų klasterių struktūra ar prarandamos išskirtys. Duomenų klasterizavimas – tai procesas, kai duomenys suskirstomi į klasterius taip, kad skirtumai klasterių viduje būtų kuo mažesni, o tarp klasterių – kuo didesni [82]. Duomenų aibės imtis gali būti sudaroma šiais būdais: atsitiktinė imtis, stratifikuota (sluoksninė) imtis, sistemingoji imtis, klasterizuota (lizdinė) imtis ir kt. [83]. Tačiau šiais atvejais tik keletas retų klasterių atstovų arba ne visi atokiau esantys taškai (išskirtys) parenkami į imtį, nors gali būti svarbūs analizei ir rezultatams. Išskirtys gali būti apibrėžiamos kaip netipiniai ir reti stebėjimai, nutolę nuo likusiųjų taškų [84]. Išskirtys gali turėti ir svarbios informacijos [85]. Be to, yra svarbių taškų, kurie yra retai išsidėstę ir gali telktis į atskirus klasterius. Ypač tai pasakytina apie medicininius duomenis, kur stebimi pacientai, ar apie kitas tyrimų sritis, kur maži klasteriai (grupelės) gali būti esminiai ir ypatingi, kadangi domina kiekvienas stebėjimas su savo individualiais požymiais. Sukčiavimo aptikimo analizei reti įvykiai gali būti įdomesni (svarbesni) už reguliarius įvykius [82]. Retų klasterių atstovai, išskirtys dažnai suteikia reikšmingos informacijos apie tiriamus duomenis. 3.3 paveiksle pateikiamas tankiai ir retai išsidėsčiusių taškų pavyzdys. Šioje disertacijoje siekiama vizualizuoti visus arba didžiąją dalį retai išsidėsčiusių taškų, sudarančių atskirus klasterius, ir taškus, nutolusius nuo daugumos taškų. Pateiktame pavyzdyje matyti, kad iš geltonai, raudonai

pažymėtų taškų, sudarančių atskirus retus klasterius, į imtį siekiama įtraukti kuo daugiau taškų. O iš mėlynai pažymėtų taškų klasterio į imtį galima įtraukti daug mažiau taškų, nes šie stipriai susitelkę, vienas kitą perdengia ir yra panašūs. Iš purpurine spalva pažymėto taško(-ų) klasterio tikimasi įtraukti bent vieną atstovą (tašką).



3.3 pav. Tankiai ir retai išsidėsčiusių taškų klasterių pavyzdys

Matyti, kad į duomenų aibės imtį svarbu įtraukti kuo daugiau taškų iš retų klasterių, kurie nebūtų atrinkti, jei imtis būtų sudaroma standartiniais būdais. Šioje disertacijoje siūlomas duomenų aibės imties sudarymo būdas atsižvelgia į duomenų aibės taškų tankumą. Tam yra naudojamas duomenų aibės klasterizavimas, t. y. vertinama atstumų suma nuo kiekvieno taško iki klasterio centro ir taškų skaičius klasteryje.

Pasiūlytas duomenų aibės imties sudarymo būdas susideda iš žingsnių:

1 žingsnis: daugiamačiai taškai iš matricos X, kurios dydis $n \times m$, suklasterizuojami į τ klasterių (klasteriais galima laikyti žinomas klases, tokiu atveju klasterizavimas gali būti netaikomas);

2 žingsnis: kiekvienam i-tajam klasteriui apskaičiuojama atstumų nuo kiekvieno taško X_j^i iki klasterio centro M_i atstumų suma $D_i = \sum_{j=1}^{N_i} d(X_j^i, M_i)$, čia N_i yra i-tojo klasterio taškų skaičius, $i = 1, \dots, \tau$;

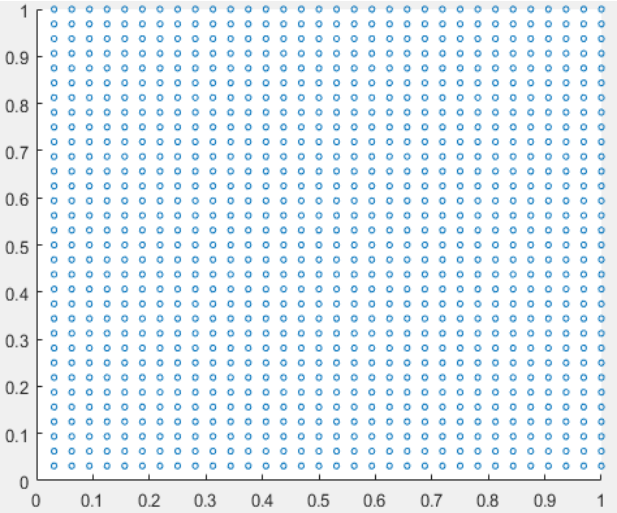
3 žingsnis: parenkamas imties dydis s, t. y. taškų kandidatų skaičius, kurie bus vizualizuojami;

4 žingsnis: apskaičiuojamas santykis $r_i = D_i/N_i$, nurodantis, kiek taškų turėtų būti atrenkama į imtį iš kiekvieno klasterio. Jei $r_i = 0$, į imtį įtraukiamas bet kuris vienas i-ojo klasterio taškas. Atrenkamų į imtį taškų skaičius iš kiekvieno klasterio apskaičiuojamas pagal formulę: $N_i' = \frac{r_i \times s}{\omega}$, čia $\omega = \sum_{i=1}^{\tau} r_i$, ir s – duomenų aibės imties dydis. Kuo didesnis i-tojo klasterio santykis r_i, tuo daugiau taškų iš šio klasterio reikia atrinkti į imtį;

5 žingsnis: sudaroma duomenų aibės imtis X_S dydžio $s \times m$;

6 žingsnis: randama duomenų aibės imties X_S projekcija Y_S, kurios dydis yra $s \times d$.

Kyla klausimas, kaip parinkti imties dydį s. Kadangi taškai bus pavaizduoti kompiuterio ekrane, tikslinga šį dydį susieti su monitoriaus raiška. Tarkime, turime kompiuterio monitorių, kurio raiška yra 1280×1024 pikselių, ir norime, kad gautas daugiamačių taškų projekcijos vaizdas užimtų penktadalį ekrano. Jei vienas taškas sudarytas iš 256 pikselių, kaip rezultatą bus galima pavaizduoti 1024 taškus. Taigi *3 žingsnyje* siūloma rinktis imties dydį $s = 1024$. Šis faktas iliustruojamas 3.2 paveiksle, kuriame dvimačiai taškai yra tolygiai pasiskirstę kvadrate. Matyti, kad taškai nepersidengia, kai jų yra 1024, tačiau analizuojant realius duomenis, taškai nebus taip tolygiai pasiskirstę, priešingai, koncentruosis į grupes ir nebus taip plačiai ir retai išsidėstę. Vizualizuoti 1024 realios aibės taškai būtų labiau susitelkę nei 3.4 paveiksle ir tikėtina, kad dalis iš jų persidengtų. Taškų persidengimas šalinamas kitu siūlomos vizualizavimo strategijos etapu (žr. 3.2.2 skyrelį).



3.4 pav. Taškų vizualizavimas intervale [0,1], imties dydis: $s = 1024$

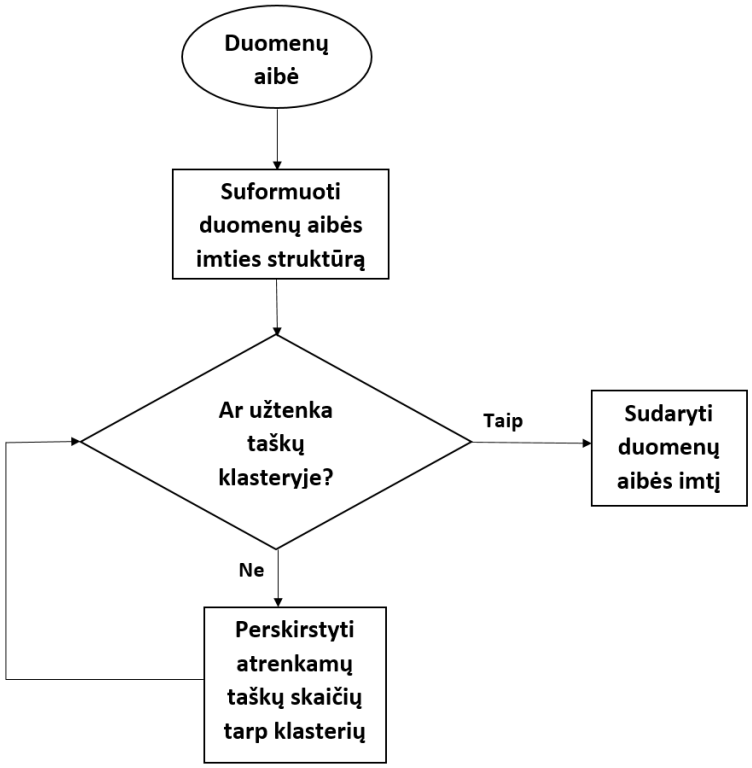
Gali būti atvejų, kai *4 žingsnyje* apskaičiuotas išrenkamų taškų skaičius N_i' yra didesnis už faktinį taškų skaičių N_i klasteryje. Tokiu atveju siūloma į imtį įtraukti visus klasterio taškus ir padidinti iš kitų klasterių įtraukiamų į imtį taškų skaičių pagal jų koeficientą r_i. Padidėjimas yra lygus $\delta = N_i' - N_i$. 3.1 lentelėje pateikiamas taškų, įtraukiamų į imtį, perskirstymo pavyzdys, t. y. iš pirmo ir antro klasterio išrenkamų taškų skaičius yra didinamas pagal jų koeficientus r_i, kai analizuojama *Random3* duomenų aibė ($n = 2515, m = 10,$ $\tau = 3,$ imties dydis $s = 1024$). Šios duomenų aibės aprašymas pateikiamas 4.1 poskyryje. Matyti, kad pagal trečiojo klasterio koeficientą $r_3 = 4$ iš jo į imtį turėtų būti įtraukti $N_3' = 695$ taškai, tačiau trečiajame klasteryje iš esmės yra tik $N_3 = 15$ taškų. Taigi visus 15 taškų iš trečiojo klasterio įtraukiame į imtį ir didiname įtraukiamų taškų į imtį skaičių iš likusiųjų dviejų klasterių 680 taškais pagal koeficientus r_1 ir r_2. Taškų, įtraukiamų į imtį, skaičiaus perskirstymo schema pateikiama 3.5 paveiksle.

Duomenų aibės imties X_S projekcija *6 žingsnyje* gali būti randama bet kuriuo dimensijos mažinimo metodu.

3.1 lentelė. Įtraukiamų į imtį taškų skaičiaus perskirstymas tarp klasterių

Klasteris *(i)*	Taškų skaičius klasteryje (N_i)	D_i	Santykis (r_i)	Taškų skaičius iš klasterio (N'_i)	Taškų skaičius iš klasterio (N'_i) (perskirstytas)
1	500	461	0,922	156	492
2	2000	1941	0,971	165	517
3	15	60	4	695	15

Taškų, kurie bus parinkti iš 1-ojo ir 2-ojo klasterių, skaičius turėtų būti didinamas $\delta = 680$ taškais



3.5 pav. Įtraukiamų į imtį taškų skaičiaus perskirstymo schema

3.2.2 Taškų vizualizavimas be persidengimo (2 etapas)

Jei vizualizuojamas didelis taškų kiekis, sklaidos diagramoje dažniausiai jie persidengia. Norint identifikuoti kiekvieną tašką ar jo vietą sklaidos diagramoje reikia pašalinti persidengiančius ir vienas kitą dengiančius taškus. Apibrėžkime, kad duomenų aibės imties projekcija yra Y_S, o $Y_L, l \leq s$ yra duomenų aibės imties projekcija, kurią vizualizuosime. Siūlomas taškų vizualizavimo būdas susideda iš šių žingsnių:

1 žingsnis: pradinės imties projekcijos Y_S požymių reikšmės sunormuojamos intervale $(0; 1)$, t. y. minimalios reikšmės yra lygios 0, maksimalios reikšmės lygios 1. Normavimas atliekamas nustatyti vienodoms slenksčio t (žr. *2 žingsnį*) reikšmėms visoms duomenų aibėms;

2 žingsnis: sunormuoti imties taškai Y_S, perrenkami su pasirinktu slenksčiu t. Slenkstis t leidžia kontroliuoti taškų tankumą. Taškų perrinkimas atliekamas taip: imties taškams Y_S, apskaičiuojama atstumų matrica Δ; jei atstumas nuo vieno taško iki kito yra mažesnis už pasirinktą slenkstį t, taškas pašalinamas iš imties projekcijos Y_S,. Po perrinkimo duomenų imtis Y_L, kuri bus vizualizuojama, yra dydžio $l \times m$ $(l \leq s, d < m)$;

3 žingsnis: duomenų imtis Y_L vizualizuojama sklaidos diagramoje.

Pasiūlytas didelės apimties duomenų vizualizavimo būdas eksperimentiškai ištirtas ir gauti rezultatai pateikti 4.5 poskyryje.

3.3 Trečiojo skyriaus apibendrinimas

Šiame skyriuje pasiūlyti dimensijų mažinimo metodais gautos projekcijos paklaidos apskaičiavimo būdai didelės apimties duomenų aibėms, leidžiantys optimizuoti kompiuterio operatyviosios atminties resursus ir skaičiavimo laiką. Taip pat pasiūlyta didelės apimties duomenų aibių vizualizavimo be taškų persidengimo ir išlaikant retų klasterių ir bendrą duomenų struktūrą strategija. 3 skyriuje pateikti siūlymai gali būti pritaikyti ir kitoms programavimo kalboms su analogiškomis funkcijomis.

4 Eksperimentinių tyrimų rezultatai

Šiame skyriuje pateikiami atliktų eksperimentinių tyrimų rezultatai. Iš pradžių aprašytos duomenų aibės, naudotos eksperimentiniuose tyrimuose. Pateikiama 2.2.1 ir 2.2.2 skyreliuose nagrinėtų dimensijos mažinimo metodų lyginamoji analizė. Eksperimentiškai parodyta, kad duomenų aibės projekcija turi būti vertinama skirtingas ypatybes atspindinčiais projekcijos kokybės įvertinimo matais. Ištirta, kad pasiūlyti nauji projekcijos paklaidos skaičiavimo būdai (žr. 3.1 poskyrį) projekcijos paklaidą apskaičiuoja didelės apimties duomenų aibėms. Taip pat eksperimentiškai ištirta nauja vizualizavimo strategija, pasiūlyta 3.2 poskyryje. Tyrimų rezultatai publikuoti autorės darbuose [A1], [A2], [B1–B3], [C1], [C2].

4.1 Tyrimuose naudojami duomenys

Disertacijos eksperimentinėje dalyje naudojamos įvairios duomenų aibės, turinčios skirtingų yaptybių. Šios duomenų aibės paimtos iš duomenų bazės UCI Machine Learning Repository [86]: *Iris*, *Waveform*, *Image segmentation*, *Wine quality*, *Musk*, *Mammals*, *Magic gamma segmentation*, *Letter recognition*, *Skin segmentation*, *Yeast*, *Shuttle*, *Dspatialetwork*, *Page blocks*. *Twinpeaks*, *Helix*, *Swiss roll* duomenų aibės, sugeneruotos įrankiu MATLAB *Toolbox for Dimensionality Reduction*. Funkcijos, skirtos generuoti *Crescent and full moon*, *Corners* duomenų aibėms, paimtos iš MATLAB *Central File Exchange*. *Random1* duomenų aibė yra dirbtinai sugeneruota, o požymių galimos reikšmės tolygiai pasiskirsčiusios intervale $(0; 1{,}0)$. Duomenų aibių parametrai pateikiami 4.1 lentelėje. *Random3* duomenų aibė sugeneruota taip, kad galimos reikšmės tolygiai pasiskirsčiusios intervaluose $(0; 1{,}0)$, $(1{,}5; 2{,}5)$, $(6{,}0; 11{,}0)$, tad gaunamos trys duomenų grupės. *Random4* duomenų aibė dirbtinai sudaryta taikant funkciją, generuojančią *Crescent and full moon* taškus ir taškus, tolygiai pasiskirsčiusius intervale $(6{,}0; 11{,}0)$.

4.1 lentelė. Duomenų aibių parametrai

Duomenų aibė	**Duomenų aibės tipas**	**Objektų skaičius (n)**	**Požymių skaičius (m)**
Iris	Reali	150	4
Image segmentation	Reali	2086	19
Wine quality	Reali	4898; 3961	11
Waveform	Dirbtinė	5000	21
Musk	Dirbtinė	6581	166
Mammals	Dirbtinė	15000; 16384	72
MAGIC gamma telescope	Dirbtinė	18905	10
Letter recognition	Reali	18668	16
Skin segmentation	Reali	51444	3
Helix	Dirbtinė	5000; 15000; 30000; 250000; 500000; 700000; 750000; 1000000	3
Swiss roll	Dirbtinė	15000; 30000; 250000; 500000; 700000; 1000000	3
Twinpeaks	Dirbtinė	30000	3
Crescent and full moon	Dirbtinė	300000	4
Shuttle	Reali	58000	9
Dspatialetwork	Reali	434874	3
Corners	Dirbtinė	450000	4
Yeast	Reali	1453	8
Page blocks	Reali	5406	10
Random1	Dirbtinė	10000; 20000; 40000; 60000; 50000; 80000; 100000; 150000; 500000	20; 50; 100
Random3	Dirbtinė	2515; 15020	10
Random4	Dirbtinė	3020	4

Eksperimentai atlikti kompiuteriu, kurio pagrindinės charakteristikos yra šios: operatyvioji atmintis (RAM) – 12 GB, procesorius – Intel i5-3317U, kurio taktinis dažnis – 1,7 GHz (Max Turbo dažnis 2,6 GHz), branduolių skaičius – 2. Kompiuteryje veikia operacinė sistema – MS Windows 8.

4.2 Dimensijos mažinimo metodų tyrimas

Šiame poskyryje nagrinėjami įvairūs dimensijos mažinimo metodai, aprašyti disertacijos 2.2 poskyryje. Taip pat pateikiamas dimensijos mažinimo metodų apibendrinimas.

4.2.1 Dimensijos mažinimo metodų lyginimas

Šioje disertacijos dalyje analizuojama vienuolika duomenų aibių: *Iris, Image segmentation, Wine quality, Waveform, Musk, Mammals, MAGIC gamma telescope, Letter recognition, Skin segmentation, Helix, Swiss roll.* Šio tyrimo tikslas – nustatyti, kokių apimčių duomenis geba apdoroti dimensijos mažinimo algoritmai, sprendžiantys projekcijos paieškos uždavinį, ir kiek laiko trunka skaičiavimai. Eksperimentiniame tyrime nagrinėjami šie dimensijų mažinimo metodai:

- pagrindinių komponenčių analizė (PCA),
- daugiamačių skalių metodas (MDS),
- nepriklausomų komponenčių analizė (ICA),
- atsitiktinės projekcijos metodas (RP),
- dalinai tiesinė daugiamatė projekcija (PLMP),
- lokali afinioji daugiamatė projekcija (LAMP).

Tyrime nagrinėjama dimensijos mažinimo metodų greitaveika ir projekcijos paklaida skirtingos apimties duomenų aibėms. Projekcijos paklaida E_{Stress}, taikoma visų tyrime nagrinėjamų metodų rezultatams vertinti, apskaičiuojama pagal (5) formulę (žr. 2.3 poskyrį). Tyrimui pasirinkta projekcinės erdvės dimensija yra $d = 2$. Įprastai dimensijai mažinti taikomas Euklido atstumas, bet gali būti taikomi ir kiti duomenų panašumo matai, tik šioje disertacijoje jie nėra tirti. Šios disertacijos eksperimentinėje dalyje nepriklausomos komponentės rūšiuojamos pagal negentropijos koeficiento aproksimacijos reikšmes. Taškų projekcijai rasti reikalingos pirmosios dvi nepriklausomos komponentės, jų negentropijos koeficiento aproksimacijos reikšmės yra didžiausios. Per eksperimentinį tyrimą valdymo taškų

$X' = \{X'_1, \dots, X'_k\}$ skaičius k yra lygus $\sqrt{n}$, jų projekcija sumažintos dimensijos erdvėje randama MDS metodu.

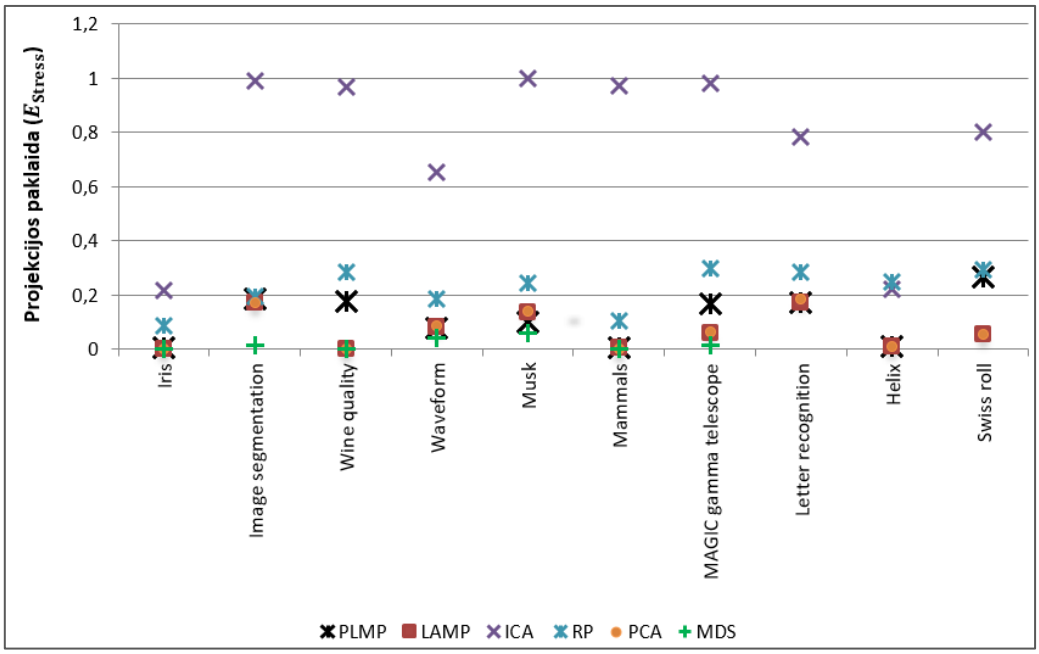
4.2 lentelėje pateikiami nagrinėtų dimensijų mažinimo metodų rezultatai iš įvairių duomenų aibių analizės. Pateikiami keturi skaičiai po kablelio, kad matytųsi ir nežymių skirtumų, o geriausios rezultatų reikšmės – paryškintos. Lentelėje prie duomenų aibės pavadinimo nurodytas objektų ir požymių skaičius. Iš pateiktų rezultatų matyti, kad MDS metodu gauta projekcijos paklaida yra pati mažiausia (nuo 0,0004 iki 0,0601) visoms duomenų aibėms, tačiau skaičiavimo laikas – pats ilgiausias. *MAGIC gamma telescope* duomenų aibė, sudaryta iš 19020 objektų, MDS metodu analizuota ilgiau kaip 33 val., tad šis metodas dar didesnės apimties duomenų aibėms nebuvo taikytas. Pačiu trumpiausiu skaičiavimo laiku dėl savo paprastumo pasižymi RP metodas, skaičiavimai neužtrunka nė sekundės su bet kuria duomenų aibe, tačiau šiuo metodu gauta projekcijos paklaida yra didesnė (nuo 0,0886 iki 0,2967) už gautą kitais metodais, išskyrus ICA metodą. Pačios didžiausios projekcijos paklaidos reikšmės gaunamos ICA metodu: kinta nuo 0,2179 iki 0,9988. Nagrinėjant dimensijų mažinimo metodus, paremtus valdymo taškais, nustatyta, kad daugeliu atveju LAMP metodu gauta projekcijos paklaida yra mažesnė už gautą PLMP metodu, išskyrus *Waveform*, *Musk* ir *Letter recognition* duomenų aibes, tačiau skaičiavimo laikas – ilgesnis. LAMP metodu gaunama projekcijos paklaida nedaug skiriasi nuo PCA metodu gaunamos paklaidos ir su didžiąją dalimi iš duomenų aibių yra mažesnė.

Projekcijos paklaidos priklausomybė nuo duomenų aibės pateikiama 4.1 paveiksle. Pastebima, kad pats išskirtiniausias yra ICA metodas, jo projekcijos paklaida su didžiąją dalimi iš duomenų aibių yra pati didžiausia. RP metodu gauta projekcijos paklaida mažesnė už gautą ICA metodu. Atskirą grupę sudaro LAMP, PLMP ir PCA metodai, jų projekcijos skiriasi nežymiai su įvairiomis duomenų aibėmis. Kaip buvo minėta anksčiau, mažiausia projekcijos paklaida gaunama, kai dimensija mažinama MDS metodu.

4.2 lentelė. Dimensijų mažinimo metodų skaičiavimo laikas (s) ir projekcijos paklaidos reikšmės naudojant skirtingas duomenų aibes

Duomenų aibė	**Laikas / paklaida**	**PLMP**	**LAMP**	**ICA**	**RP**	**PCA**	**MDS**
Iris [150×4]	Laikas	0,2344	0,5	0,0156	**<0,00001**	0,7031	5,4219
	Paklaida	0,0052	0,0021	0,2179	0,0886	0,00175	**0,0011**
Image segmentation [2310×19]	Laikas	0,2188	3,5625	0,4219	**<0,00001**	0,0469	3383,6
	Paklaida	0,1861	0,1727	0,9901	0,1969	0,17375	**0,0166**
Wine quality [4898×11]	Laikas	0,6094	8,6719	0,4844	**<0,00001**	0,0313	7092,7
	Paklaida	0,1773	0,0005	0,9668	0,2847	0,0005	**0,0004**
Waveform [5000×21]	Laikas	0,4375	12,1875	43,188	**<0,00001**	0,0781	5953,1
	Paklaida	0,0767	0,0846	0,654	0,1876	0,0852	**0,0421**
Musk [6598×166]	Laikas	0,6875	90,0781	444,5625	**<0,00001**	1,0156	12132
	Paklaida	0,0989	0,1379	0,9988	0,2464	0,1387	**0,0601**
Mammals [16384×72]	Laikas	8,1563	243,625	7,9375	**<0,00001**	1,5781	89865
	Paklaida	0,0056	0,004	0,9735	0,105	0,0016	**0,0006**
MAGIC gamma telescope [19020×10]	Laikas	1,2188	80,8281	4,7656	**<0,00001**	0,1406	119930
	Paklaida	0,1682	0,0594	0,9808	0,2967	0,0665	**0,0159**
Letter recognition [20000×16]	Laikas	1,1719	81,5781	4,7188	**0,0313**	0,375	X
	Paklaida	**0,1733**	0,1735	0,784	0,284	0,1858	
Helix [30000×3]	Laikas	0,8594	184,031	0,4531	**<0,00001**	0,0313	X
	Paklaida	0,0099	**0,0097**	0,2235	0,2482	0,0115	
Swiss roll [30000×3]	Laikas	1,7656	154,094	0,8281	**<0,00001**	0,0469	X
	Paklaida	0,2679	**0,055**	0,8009	0,2918	0,0564	

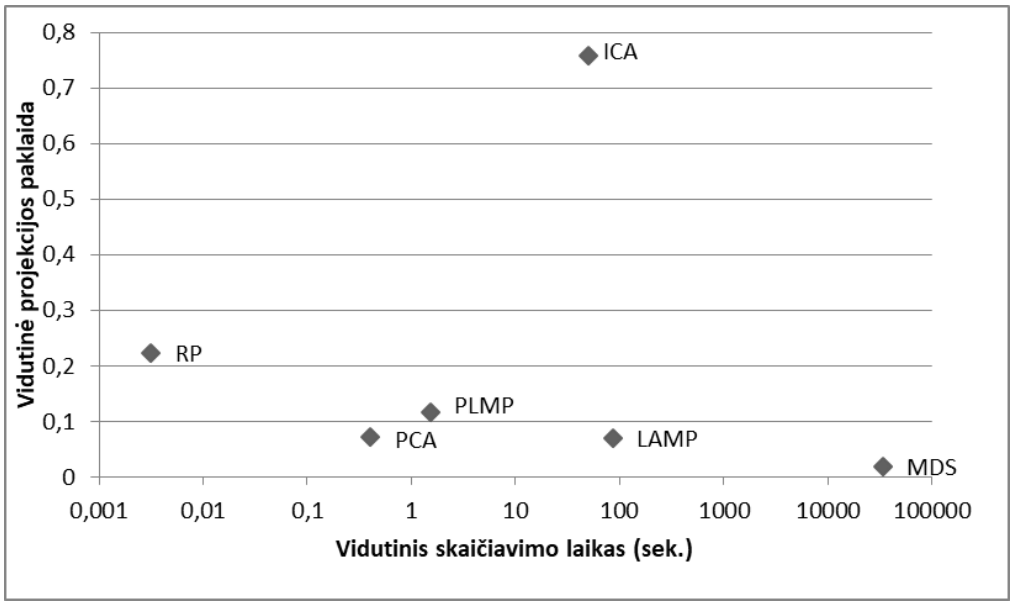
X – neskaičiuota dėl ilgos skaičiavimo trukmės



4.1 pav. Projekcijos paklaidos priklausomybė nuo duomenų aibės

4.2 paveiksle pateikti visų algoritmų vidutiniai skaičiavimo laikai ir vidutinės skaičiavimo paklaidos, gautos atitinkamai išvedus vidurkį iš visų duomenų aibių skaičiavimo laiko ir projekcijos paklaidų. Vidutiniam laikui atvaizduoti naudota logaritminė skalė. Lyginant vidutinį nagrinėjamų algoritmų skaičiavimo laiką ir vidutinę projekcijos paklaidą, nustatyta, kad MDS metodu gauta vidutinė projekcijos paklaida yra mažiausia (0,0195), tačiau vidutinis skaičiavimo laikas yra pats ilgiausias (34051,69 s). LAMP metodu gauta vidutinė projekcijos paklaida (0,0699) panaši į PCA metodu gautą paklaidą (0,0722), tačiau vidutinis skaičiavimo laikas yra beveik kelis šimtus kartų ilgesnis.

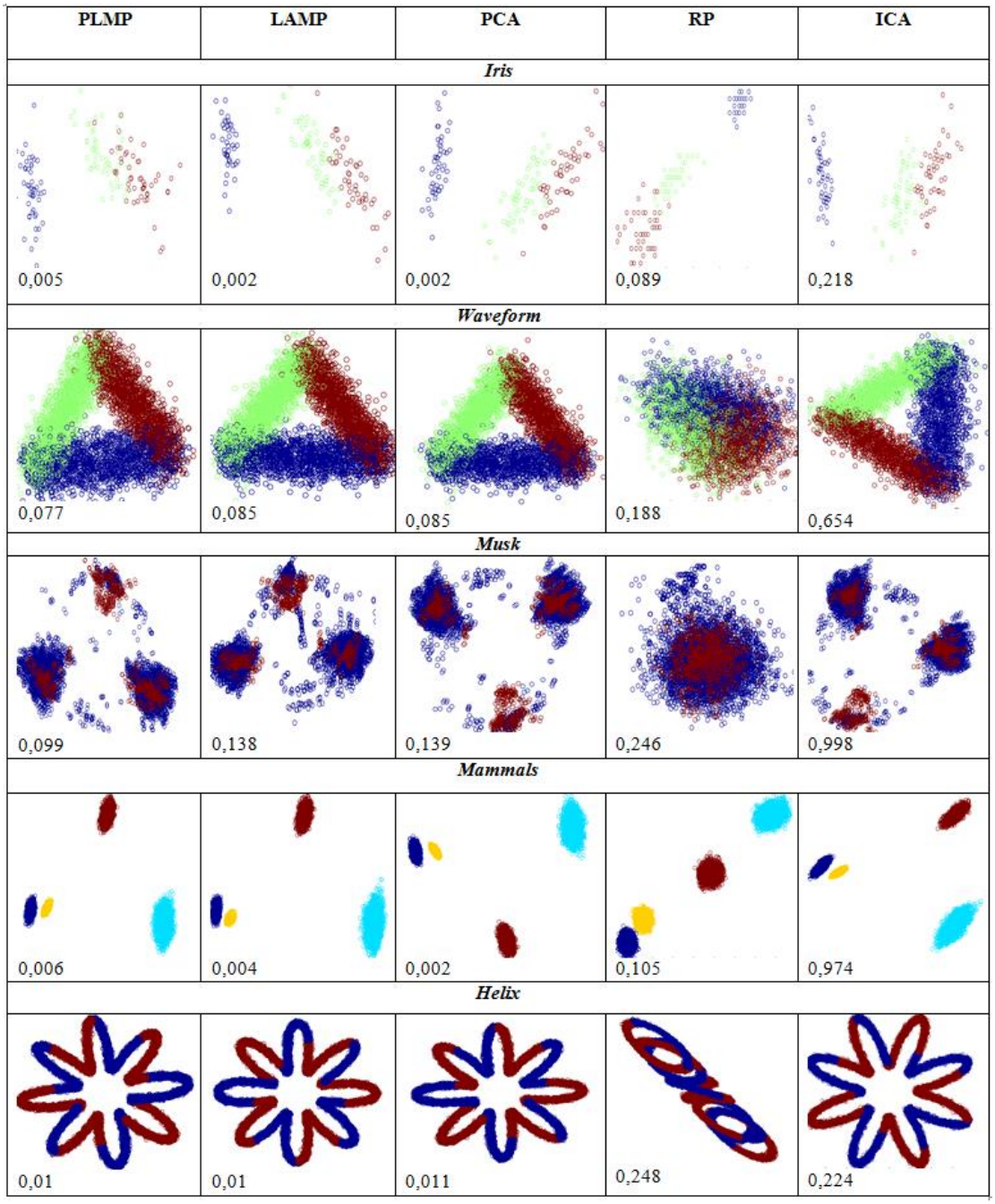
Šiek tiek didesnė vidutinė projekcijos paklaida (0,1169) gaunama PLMP metodu lyginant su PCA, LAMP, MDS metodais, tačiau vidutinis skaičiavimo laikas yra pusantros sekundės – daug trumpesnis nei LAMP metodo. RP metodo vidutinis skaičiavimo laikas yra pats trumpiausias (0,0031 s), tačiau vidutinė paklaida (0,223) yra didesnė už anksčiau minėtų metodų. ICA metodu gauta vidutinė projekcijos paklaida (0,759) yra pati didžiausia, lyginant su kitais metodais gautomis vidutinėmis projekcijos paklaidomis.



4.2 pav. Vidutinės projekcijos paklaidos priklausomybė nuo vidutinio skaičiavimo laiko naudojant skirtingus projekcijos metodus

LAMP, PLMP, PCA, RP ir ICA metodais gauti daugiamačių taškų išsidėstymai dvimatėje erdvėje, vizualizuojant skirtingas duomenų aibes, pateikiami 4.3 paveiksle. LAMP, PLMP, PCA metodais atvaizduotas daugiamačių taškų išsidėstymas dvimatėje plokštumoje yra panašus, labiausiai išsiskiria RP metodu atvaizduotos duomenų aibės, šių projekcijos paklaidos yra vienos iš didžiausių. Pastebėtina, kad ICA metodu gautuose duomenų aibių vaizduose geriau išlaikoma duomenų struktūra nei RP metodu, tačiau paklaida, rodanti, kaip išlaikomi atstumai tarp objektų pradinėje ir sumažintos dimensijos erdvėje, yra didesnė visoms duomenų aibėms, išskyrus *Helix* duomenų aibę.

Taip pat papildomai atlikti skaičiavimai didesnėms kaip 30000 objektų apimties duomenų aibėms, visų duomenų aibių požymių skaičius buvo lygus trims (žr. 4.3 lentelę). Siekta nustatyti, kokios apimties duomenims tinkami nagrinėjami dimensijos mažinimo metodai, todėl šioje tyrimo dalyje buvo tiriamas skaičiavimo laikas ir kompiuterio atminties galimybės. MDS metodas nebuvo įtrauktas į skaičiavimus, kadangi, kaip parodyta anksčiau, duomenų aibė, sudaryta iš 19020 objektų, šiuo metodu analizuota ilgiau kaip 33 val.



4.3 pav. Skirtingais metodais vizualizuotos penkios duomenų aibės (kiekvieno paveiksliuko kairiajame kampe nurodyta projekcijos paklaidos E_{Stress} reikšmė)

Visi nagrinėti metodai, išskyrus LAMP metodą, greitai randa projekciją 51444–1000000 objektų duomenų aibėms, ir tai neužtrunka nė minutės. Dimensijai sumažinti trumpiausiai trunka RP (vidutinis skaičiavimo laikas – 0,0179 s) ir PCA (vidutinis skaičiavimo laikas – 0,4218 s) metodais. Ilgiausiai skaičiavimai trunka LAMP metodu (vidutinis skaičiavimo laikas

3 val. 30 min.), be to, 1000000 objektų duomenų aibėms taikant šį metodą nepakanka kompiuterio operatyviosios atminties.

PLMP metodu projekcija randama vidutiniškai per 17,3 s 51444–1000000 objektų duomenų aibėms.

4.3 lentelė. Projekcijos metodų skaičiavimo laikas (s) naudojant didesnės apimties duomenų aibes

Duomenų aibė	PLMP	LAMP	ICA	RP	PCA
Skin segementation [51444×3]	0,9688	335,40	0,5469	>**0,00001**	0,0781
Swiss roll [500000×3]	4,7188	9198,8	1,7500	**0,0156**	0,2027
Helix [500000×3]	12,6250	12800,0	1,8750	**0,0156**	0,2188
Swiss roll [700000×3]	7,0938	18245,0	2,7969	**0,0156**	0,2656
Helix [700000x3]	14,7969	22539,0	2,4063	**0,0156**	0,2656
Swiss roll [1000000×3]	27,3750	X	8,0625	**0,0313**	0,9531
Helix [1000000×3]	53,1875	X	8,9219	**0,0313**	0,9688
Vidutinis skaičiavimo laikas	17,2523	12623,64	3,7656	**0,0179**	0,4218

X – nepakanka kompiuterio operatyviosios atminties

Apibendrinimas. Atliktas tyrimas parodė, kad mažiausia projekcijos paklaida gaunama MDS metodu, tačiau skaičiavimai trunka gana ilgai (nuo 16 val. 30 min. iki 33 val.) su didesnėmis kaip 5000 objektų apimties duomenų aibėmis. RP metodu skaičiavimams atlikti neprireikia nė sekundės su bet kuria duomenų aibe, tačiau projekcijos paklaida gaunama didelė (0,0886–0,2967). Pačios didžiausios projekcijos paklaidos reikšmės (0,2179–0,9988) gaunamos ICA metodu, tačiau duomenų aibių vaizdai dvimatėje erdvėje panašūs į kitais metodais gautus vaizdus, kurių projekcijos paklaida reikšmingai mažesnė. LAMP ir PLMP metodais gaunamos projekcijos paklaidos nėra didelės ir yra panašios į PCA metodu gaunamas paklaidas. LAMP metodu gaunama projekcijos paklaida didžiajai daliai iš duomenų aibių net yra šiek tiek mažesnė, tačiau šis metodas imlus laikui, kai objektų skaičius

didesnis už 500000. Galima teigti, kad taikyti dimensijų mažinimo metodai greitai susidoroja su įvairios apimties (150–1000000 objektų) duomenų aibėmis, išskyrus MDS ir LAMP metodus, tačiau šie metodai pasižymi mažiausia projekcijos paklaida iš nagrinėtų projekcijos metodų. Lyginant MDS ir LAMP metodų skaičiavimų laikus, LAMP metodas yra greitesnis.

4.2.2 Dimensijos mažinimas radialinėmis bazinėmis funkcijomis paremtu metodu

Šioje disertacijos dalyje detaliai ištirtas dimensijos mažinimo metodas, paremtas radialinių bazinių funkcijų teorija (RBF) ir valdymo taškais.

Tyrimo metu nustatyta, kad taikant ROLS algoritmą visų taškų projekcijai ($d = 2$) gaunamos dvi skirtingos valdymo taškų aibės, todėl disertacijoje siūloma šias dvi valdymo taškų aibes sujungti. Jungimas atliekamas dviem būdais: 1) atrenkami unikalūs valdymo taškų aibių taškai; 2) atrenkami sutampantys valdymo taškų aibių taškai. Pirmuoju būdu visi skirtingi taškai be pasikartojimų yra atrenkami iš dviejų valdymo taškų aibių; o antruoju – atrenkami tie taškai, kurie yra abiejose valdymo taškų aibėse. Taip gaunama valdymo taškų aibė mažinti duomenų aibės dimensijai RBF metodu.

Taip pat siūloma valdymo taškų aibes parinkti atsitiktiniu ir stratifikuotu imties sudarymo būdais. Paprastoji atsitiktinė imtis – tokia imtis, kurios visų populiacijos elementų galimybės patekti į imtį yra vienodos. Stratifikuotu imties sudarymo būdu duomenų aibė suskirstoma į sluoksnius (stratus). Kiekvienam sluoksniui taikomas paprastosios atsitiktinės imties sudarymo būdas [87]. Šių imties sudarymo būdų privalumas yra tas, kad jų algoritmai greiti, nereikalauja atlikti daug skaičiavimų.

Šiame tyrime nagrinėjamas RBF metodo junginys su įvairiais valdymo taškų parinkimo būdais:

1 *junginys* (*RBF, ROLS, unikalūs*): RBF metodas, kai valdymo taškai parenkami ROLS algoritmu. Valdymo taškų aibės jungiamos iš atrinktų unikalių valdymo taškų. ROLS algoritmas stabdomas, kai parenkamas

norimas valdymo taškų skaičius. Valdymo taškų skaičius parenkamas euristuškai.

2 *junginys* (*RBF, ROLS, sutampantys*)*:* RBF metodas, kai valdymo taškai parenkami ROLS algoritmu. Valdymo taškų aibės jungiamos iš atrinktų sutampančių valdymo taškų. ROLS algoritmas stabdomas, kai atrenkamas norimas valdymo taškų skaičius.

3 *junginys* (*RBF, ROLS, min Stress*)*:* RBF metodas, kai valdymo taškai parenkami ROLS algoritmu. Nagrinėjama ta valdymo taškų aibė, kurios *Stress* reikšmė yra mažiausia. ROLS algoritmas stabdomas, kai pasiekiamas arba norimas valdymo taškų skaičius, arba randama iteracija su mažiausia projekcijos paklaida (E_{Stress}). Ši iteracija rodo, kiek valdymo taškų bus įtraukta į tolesnius skaičiavimus.

4 *junginys (RBF, atsitiktinė imtis):* RBF metodas, kai valdymo taškai parenkami atsitiktinės imties sudarymo būdu.

5 *junginys* (*RBF, stratifikuota imtis*)*:* RBF metodas, kai valdymo taškai parenkami stratifikuotos imties sudarymo būdu.

RBF metodo junginys su pasiūlytais įvairiais valdymo taškų parinkimo būdais eksperimentiškai ištirtas su įvairiomis duomenų aibėmis.

Eksperimentiniam tyrimui pasirinktos aštuonios duomenų aibės: *Yeast*, *Image segmentation*, *Waveform*, *Page blocks*, *MAGIC gamma telescope*, *Letter recognition*, *Helix*, *Swiss roll*. Branduolys parinktas toks pat kaip ir siūlo metodo autoriai, t. y. multikvadratinis RBF branduolys, $\phi(r) = \sqrt{c^2+(\varepsilon r)^2}$, $c = \varepsilon = 1$. Atliktas tyrimas su keliais kandidatų ir maksimalaus valdymo taškų skaičių rinkiniais. Pirmuoju atveju analizė atlikta su šiais parametrais: kandidatų skaičius $N = 200$, maksimalus valdymo taškų skaičius yra $k = 30$. Antruoju atveju: kandidatų skaičius $N = 300$, maksimalus valdymo taškų skaičius yra $k = 100$. Daugiamačių valdymo taškų projekcija mažesnės dimensijos erdvėje ($d = 2$) randama PCA metodu. Siekiant palyginti RBF metodą naudojant įvairias valdymo taškų parinkimo strategijas, pasiūlytas 2.2.2 skyrelyje, rezultatai vertinami pagal projekcijos paklaidą E_{Stress},

pateikiamą (5) formulėje (žr. 2.3 poskyrį) ir pagal skaičiavimų įvykdymo laiką sekundėmis. Kiekvienai duomenų aibei atlikta po 100 eksperimentinių skaičiavimų, iš jų rezultatų išvestas vidurkis.

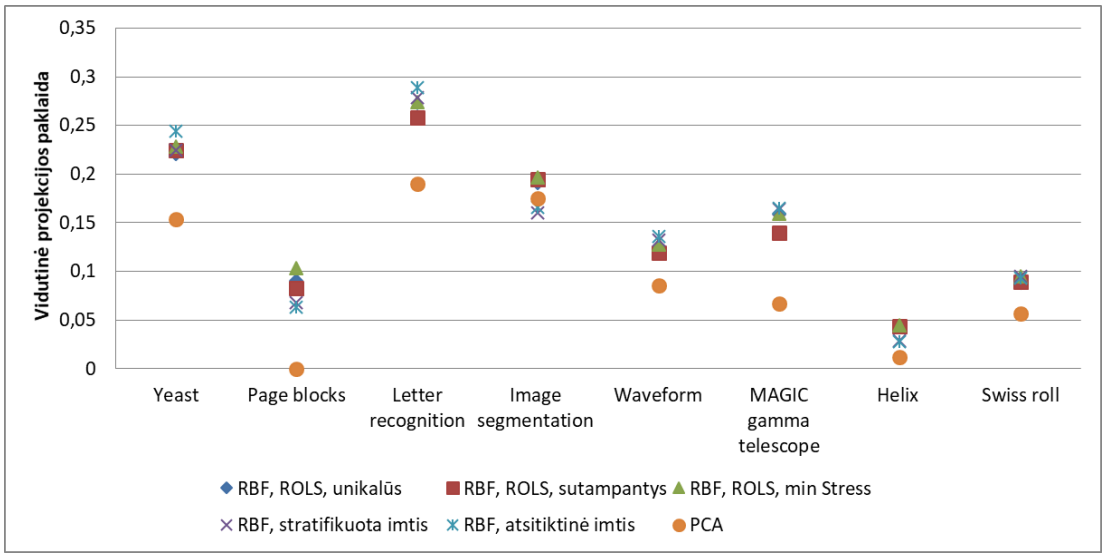
4.4 lentelėje pateikiamos skirtingų būdų projekcijos paklaidos reikšmės, mažiausios reikšmės pateikiamos paryškintu šriftu. Dar pateikiamos minimali (*Min*), maksimali (*Max*) ir dispersijos (*Var*) reikšmės. Mažiausios projekcijos paklaidos reikšmės gaunamos RBF metodu, kai valdymo taškai parenkami ROLS algoritmu, analizuojant penkias iš aštuonių duomenų aibių (*Yeast*, *Letter recognition*, *Waveform*, *MAGIC gamma telescope*, *Swiss roll*). Tyrimas parodė, kad valdymo taškų aibių jungimas (sutampantys ar unikalūs taškai) neturi įtakos gaunamiems rezultatams. Nors RBF metodu, kai valdymo taškai parenkami ROLS algoritmu, gaunamos mažesnės projekcijos paklaidos reikšmės su penkiomis iš aštuonių duomenų aibių, tačiau palyginus su kitais metodais, t. y. RBF metodu, kai valdymo taškai parenkami stratifikuotos imties sudarymo būdu (RBF metodu, kai valdymo taškai parenkami atsitiktinės imties sudarymo būdu), gauta projekcijos paklaida reikšmingai nesiskiria. Dispersijos reikšmės rodo, kad reikšmių išsibarstymas aplink vidurkį yra mažas su visais valdymo taškų junginiais.

4.4 lentelė. Projekcijos paklaidos reikšmės analizuojant aštuonias duomenų aibes

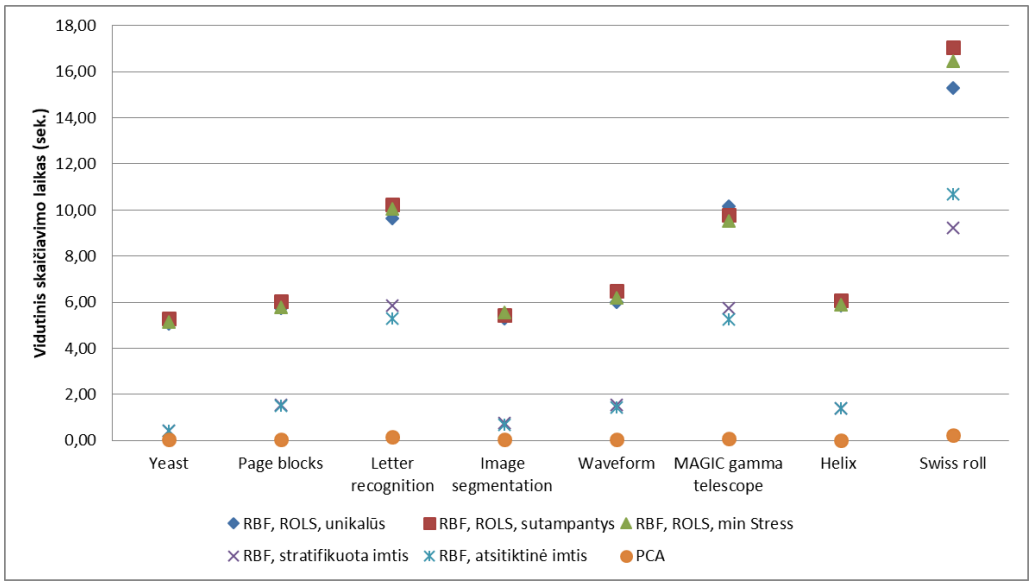
Duomenų aibė		**RBF, ROLS, sutampantys taškai (k~30)**	**RBF, ROLS, unikalūs taškai (k~30)**	**RBF, ROLS, min *Stress* (k~30)**	**RBF, stratifikuota imtis (k~30)**	**RBF, atsitiktinė imtis (k~30)**
Yeast [1453x8]	Paklaida	**0,221**	0,224	0,228	0,225	0,244
	Min	0,156	0,156	0,164	0,183	0,180
	Max	0,383	0,395	0,401	0,281	0,360
	Var	0,002	0,002	0,003	$4{,}85\times10^{-4}$	0,001
Page blocks [5406x10]	Paklaida	0,090	0,083	0,104	0,068	**0,063**
	Min	0,014	0,015	0,016	0,004	0,003
	Max	0,593	0,436	0,498	0,284	0,251
	Var	0,010	0,008	0,009	0,003	0,003
Letter recognition [18668x16]	Paklaida	0,259	**0,258**	0,274	0,279	0,289
	Min	0,220	0,217	0,224	0,236	0,242
	Max	0,327	0,343	0,438	0,327	0,339
	Var	$5{,}33\times10^{-4}$	$5{,}92\times10^{-4}$	0,0016	$3{,}71\times10^{-4}$	$5{,}04\times10^{-4}$
Image segmentation [2086x19]	Paklaida	0,191	0,195	0,197	**0,160**	0,166
	Min	0,121	0,136	0,142	0,129	0,137
	Max	0,606	0,557	0,571	0,209	0,293
	Var	0,005	0,006	0,005	$1{,}91\times10^{-4}$	$4{,}12\times10^{-4}$
Waveform [5000x21]	Paklaida	0,119	**0,119**	0,128	0,133	0,136
	Min	0,107	0,107	0,108	0,117	0,116
	Max	0,136	0,145	0,159	0,170	0,160
	Var	$3{,}45\times10^{-5}$	$4{,}17\times10^{-4}$	$1{,}51\times10^{-4}$	$8{,}83\times10^{-4}$	$1{,}06\times10^{-4}$
MAGIC gamma telescope [18905x10]	Paklaida	0,159	**0,140**	0,159	0,164	0,165
	Min	0,083	0,075	0,084	0,105	0,097
	Max	0,287	0,296	0,343	0,317	0,267
	Var	0,003	0,002	0,003	0,001	0,002
Helix [5000x3]	Paklaida	0,043	0,044	0,044	0,029	**0,028**
	Min	0,015	0,016	0,015	0,015	0,015
	Max	0,173	0,145	0,193	0,062	0,054
	Var	$7{,}98\times10^{-4}$	$6{,}54\times10^{-4}$	$8{,}92\times10^{-4}$	$7{,}56\times10^{-5}$	$7{,}59\times10^{-4}$
Swiss roll [30000x3]	Paklaida	**0,088**	0,089	0,095	0,095	0,093
	Min	0,061	0,064	0,058	0,063	0,064
	Max	0,180	0,151	0,194	0,135	0,147
	Var	$6{,}45\times10^{-4}$	$4{,}36\times10^{-4}$	$7{,}37\times10^{-4}$	$2{,}26\times10^{-4}$	$2{,}74\times10^{-4}$

4.4 paveiksle pateikiamos vidutinės projekcijos reikšmės, kai duomenų aibių dimensija mažinama RBF metodu su skirtingais valdymo taškų parinkimo junginiais. Matyti, kad vidutinės projekcijos paklaidos reikšmės reikšmingai nesiskiria, kai valdymo taškai (k~30) parenkami su skirtingais junginiais visoms duomenų aibėms. Nors projekcijos paklaidos reikšmės panašios, tačiau pastebėtinas reikšmingas skaičiavimo laiko skirtumas. 4.5 paveiksle pateikiamas aštuonių skirtingų duomenų aibių vidutinis skaičiavimo laikas. Matyti, kad vidutinis skaičiavimo laikas RBF metodu, kai valdymo taškai parenkami ROLS algoritmu, yra 2,4 kartus ilgesnis už RBF metodo, kai valdymo taškai parenkami stratifikuotos imties sudarymo būdu (ar RBF metodo, kai valdymo taškai parenkami atsitikinės imties sudarymo būdu).

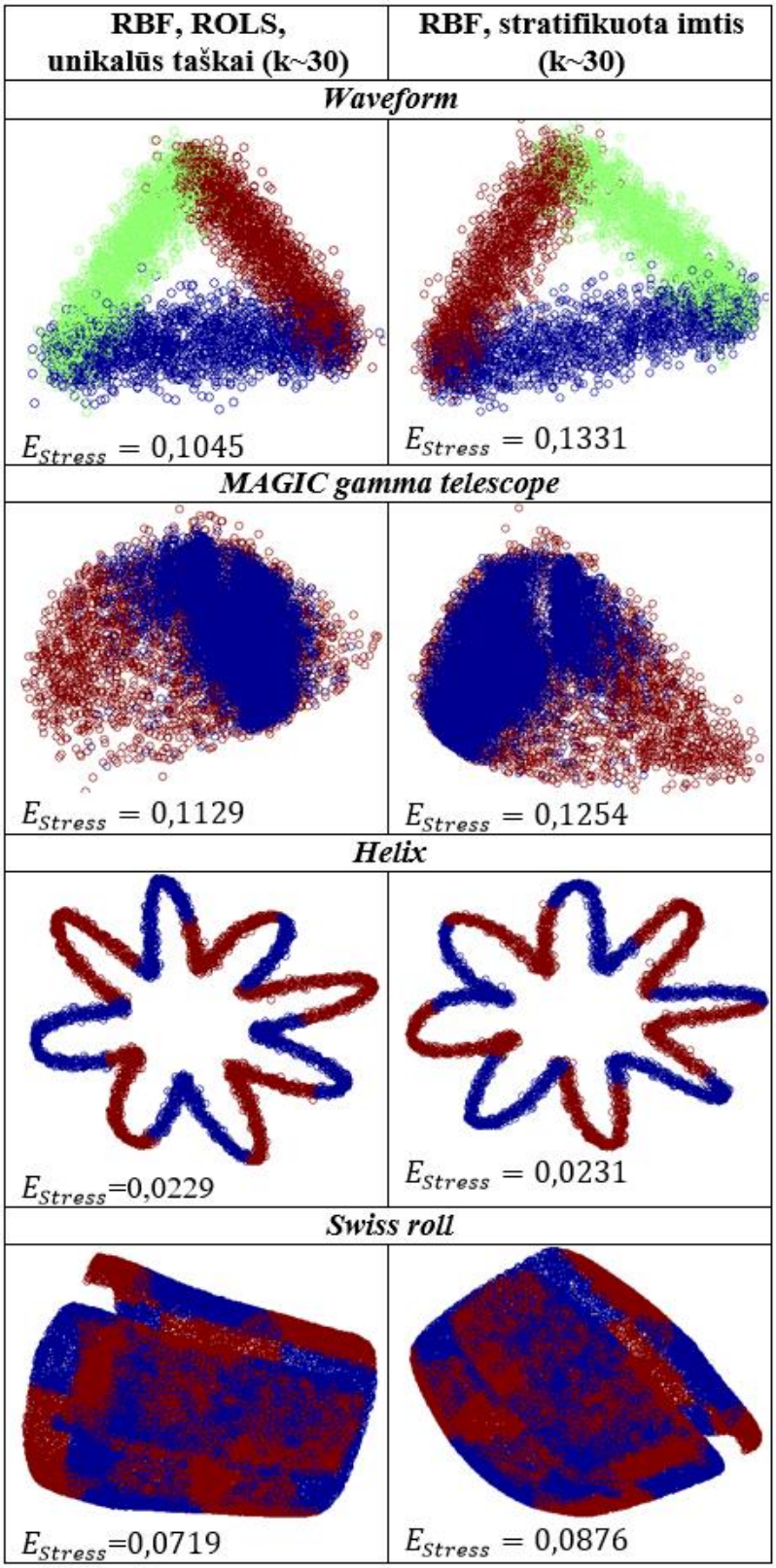
Waveform duomenų aibės projekcija randama per 6 s RBF metodu, kai valdymo taškai parenkami ROLS algoritmu, ir 1,65 s, RBF metodu, kai valdymo taškai parenkami stratifikuotos imties sudarymo būdu (ar RBF metodu, kai valdymo taškai parenkami atsitiktinės imties sudarymo būdu). RBF metodo, kai valdymo taškai parenkami stratifikuotos imties sudarymo būdu (ar RBF metodo, kai valdymo taškai parenkami atsitiktinės imties sudarymo būdu), skaičiavimo laikas yra greitas dėl paprasto valdymo taškų parinkimo algoritmo. Duomenų aibių *Waveform, Helix, MAGIC gamma telescope, Swiss roll* projekcijos, gautos RBF metodu, kai valdymo taškai parenkami ROLS algoritmu ir RBF metodu, kai valdymo taškai parenkami stratifikuotu imties sudarymo būdu, vizualizuotos 4.6 paveiksle. Matyti, kad abiem atvejais gautas taškų išsidėstymas yra panašus ir projekcijos paklaida reikšmingai nesiskiria.



4.4 pav. Vidutinė projekcijos paklaidos reikšmė naudojant skirtingus dimensijos mažinimo metodus



4.5 pav. Vidutinis skaičiavimo laikas (s) naudojant skirtingus dimensijos mažinimo metodus



4.6 pav. Keturių duomenų aibių projekcija naudojant RBF metodą, kai valdymo taškai parenkami ROLS algoritmu ir kai valdymo taškai parenkami stratifikuotos imties sudarymo būdu

Be to, atlikti tyrimai ir su didesniu taškų kandidatų ($N = 300$) ir valdymo taškų ($k = 100$) skaičiumi. Gauti skaičiavimų rezultatai pateikiami 4.5 lentelėje. Šioje tyrimo dalyje nagrinėjami du būdai: RBF metodas, kai valdymo taškai parenkami ROLS algoritmu; RBF metodas, kai valdymo taškai parenkami stratifikuotos imties sudarymo būdu. Mažesnės projekcijos reikšmės pateikiamos paryškintu šriftu. Akivaizdu, kad padidinus valdymo taškų skaičių, projekcijos paklaida sumažėja su abiem būdais. Matyti, kad RBF metodu, kai valdymo taškai parenkami ROLS algoritmu, gaunamos mažesnės projekcijos paklaidos reikšmės su šešiomis iš aštuonių duomenų aibių (*Yeast*, *Letter recognition*, *Waveform*, *Magic Gama telescope*, *Helix*, *Swiss roll*), palyginus su RBF metodu, kai valdymo taškai parenkami stratifikuotos imties sudarymo būdu. Šiais būdais gautos vidutinės projekcijos paklaidos yra atitinkamai $0{,}107$ ir $0{,}111$.

Kitaip nei projekcijos paklaida, šių dviejų būdų vidutinis skaičiavimo laikas skiriasi $8{,}6$ karto. Pavyzdžiui, *Letter recognition* duomenų aibės projekcija randama per $35{,}2$ s RBF metodu, kai valdymo taškai parenkami ROLS algoritmu, ir per 5 s RBF metodu, kai valdymo taškai parenkami stratifikuotos imties sudarymo būdu, o projekcijos paklaidos reikšmės atitinkamai yra $0{,}21$ ir $0{,}22$. Nors RBF metodu, kai valdymo taškai parenkami ROLS algoritmu, su didesniu taškų kandidatų ir valdymo taškų skaičiumi gaunamos mažesnės projekcijos paklaidos reikšmės, tačiau esti ryšis tarp projekcijos kokybės ir ilgo skaičiavimo laiko. Pažymėtina, kad valdymo taškų skaičiaus didinimas neturi stiprios įtakos RBF metodo, kai valdymo taškai parenkami stratifikuotos imties sudarymo būdu, skaičiavimo laikui. Matyti, kad RBF metodu, kai valdymo taškai parenkami stratifikuotos imties sudarymo būdu, taupomas skaičiavimo laikas, o prarandamas projekcijos paklaidos tikslumas nėra reikšmingas, palyginus su RBF metodu, kai valdymo taškai parenkami ROLS algoritmu.

4.5 lentelė. Projekcijos paklaidos (E_{Stress}) ir skaičiavimo laiko (s) reikšmė naudojant aštuonias duomenų aibes

Duomenų aibė	**Paklaida / Laikas**	**RBF, ROLS, unikalūs taškai (k~100)**	**RBF, stratifikuota imtis (k~100)**
Yeast [1453×8]	Paklaida	**0,178**	0,189
	Laikas	29,729	0,571
Page blocks [5406×10]	Paklaida	0,032	**0,019**
	Laikas	29,729	1,615
Letter recognition [18668×16]	Paklaida	**0,207**	0,221
	Laikas	35,219	5,574
Image segmentation [2086×19]	Paklaida	0,161	**0,144**
	Laikas	29,280	0,737
Waveform [5000×21]	Paklaida	**0,0950**	0,103
	Laikas	30,225	1,565
MAGIC gamma telescope [18905×10]	Paklaida	**0,0972**	0,121
	Laikas	31,923	6,428
Helix [5000×3]	Paklaida	**0,0176**	0,0179
	Laikas	29,762	1,4928
Swiss roll [30000×3]	Paklaida	**0,065**	0,071
	Laikas	39,163	11,7306
Vidurkis	Paklaida	0,107	0,111
	Laikas	31,879	3,714

Apibendrinimas. Išnagrinėjus dimensijos mažinimą RBF metodu su įvairiomis valdymo taškų parinkimo strategijomis nustatyta, kad RBF metodu, kai valdymo taškai parenkami stratifikuotos imties sudarymo būdu (RBF metodu, kai valdymo taškai parenkami atsitiktinės imties sudarymo būdu), skaičiavimai atliekami greičiau nei RBF metodu, kai valdymo taškai parenkami ROLS algoritmu. Vidutinis šių būdų skaičiavimo laikas skiriasi 2,4 karto su 30 valdymo taškų ir 8,6 karto su 100 valdymo taškų. Nors projekcijos paklaida RBF metodu, kai valdymo taškai parenkami ROLS algoritmu yra mažesnė nei RBF metodu, kai valdymo taškai parenkami stratifikuotos imties sudarymo būdu (RBF metodu, kai valdymo taškai parenkami atsitiktinės imties sudarymo būdu), skirtumas yra nereikšmingas. Rezultatai parodė, kad RBF metodu, kai valdymo taškai parenkami stratifikuotos imties sudarymo būdu (RBF metodu, kai valdymo taškai parenkami atsitiktinės imties sudarymo būdu), skaičiavimai atliekami greitai ir nėra apriboti valdymo taškų skaičiumi.

Galima pabrėžti, kad parandamas projekcijos paklaidos tikslumas nėra reikšmingas lyginant su skaičiavimo laiku, kurį taupome RBF metodu, kai valdymo taškai parenkami stratifikuotos imties sudarymo būdu (RBF metodu, kai valdymo taškai parenkami atsitiktinės imties sudarymo būdu). Atsitiktinės ir stratifikuotos imties sudarymo būdai gali būti naudojami vietoje ROLS algoritmo valdymo taškams parinkti.

4.2.3 Dimensijos mažinimas valdymo taškais paremtais metodais

Šioje tyrimo dalyje nagrinėjami tik metodai, paremti valdymo taškais. Iš pradžių parenkama dalis iš duomenų aibės taškų, vadinamų valdymo taškais, tada randama jų vieta mažesnės dimensijos erdvėje ($d = 2$). Informacija, gauta iš valdymo taškų, naudojama likusiųjų taškų projekcijai rasti. Lyginimui parinkti trys pastaruoju metu pasiūlyti metodai, skirti daugiamačių duomenų dimensijai mažinti: PLMP, LAMP ir metodas paremtas RBF teorija ir valdymo taškais. Šie metodai aprašyti disertacijos 2.2.2 skyrelyje. Vykdant šį eksperimentinį tyrimą PLMP ir LAMP metodais valdymo taškai parenkami atsitiktinai, o jų skaičius parinktas $k = \sqrt{n}$ (tokius parametrus rekomenduoja metodų autoriai). RBF metodu taškai kandidatai parenkami atsitiktinai, o valdymo taškai ROLS metodu taip, kaip nurodo šį metodą pasiūlę autoriai [39]. Šiame tyrime nagrinėjant RBF metodą lyginimui parinkti keli kandidatų ir valdymo taškų skaičių rinkiniai: 1) $N = 200$ kandidatų taškų ir $k = 30$ valdymo taškų; 2) $N = 300$ kandidatų taškų ir $k = 100$ valdymo taškų. Projekcijos metodų valdymo parametrai ir nustatymai pateikiami 4.6 lentelėje.

Tyrimu siekiama parodyti, kokios apimties duomenis gali apdoroti dimensijos mažinimo metodai, paremti valdymo taškais. Naudojamos sugeneruotos įvairių apimčių ir dimensijų duomenų aibės (*Random1*). Eksperimentiniame tyrime nagrinėjama dimensijų mažinimo metodų greitaveika ir projekcijos paklaida su skirtingos apimties duomenų aibėmis. Projekcijos paklaida E_{Stress}, naudojama visų tyrime nagrinėjamų metodų rezultatams vertinti, apskaičiuojama pagal (5) formulę (žr. 2.3 poskyrį).

4.6 lentelė. Metodų valdymo parametrai ir kiti nustatymai

Metodas	Pagrindinės ypatybės	Valdymo parametrai ir kiti nustatymai
Dalinai tiesinė daugiamatė projekcija, PLMP	Vertinami atstumai tarp dalies taškų (sudaroma imtis), pagal kurių projekciją randamos likusiųjų duomenų aibės taškų projekcijos.	Imčiai sudaryti parinktas valdymo taškų skaičius $\sqrt{n}$, jų projekcija randama daugiamačių skalių metodu. Tiesinių lygčių sistema sprendžiama jungtinių gradientų metodu: tolerancija yra $(0{,}2)$, maksimalus iteracijų skaičius lygus 15.
Lokali afinioji daugiamatė projekcija, LAMP	Randama valdymo taškų projekcija, pagal kurią randamos likusiųjų duomenų aibės taškų projekcijos.	Valdymo taškų skaičius $\sqrt{n}$, jų projekcija randama MDS metodu.
Metodas, paremtas radialinių bazinių funkcijų teorija, kai valdymo taškai parenkami ROLS algoritmu, RBF+ROLS	Iš pradžių randamos tik dalies duomenų aibės taškų (valdymo taškų), koordinatės sumažintos dimensijos erdvėje, pagal kurias naudojant RBF randamos likusiųjų duomenų aibės taškų projekcijos.	Taškų kandidatų skaičius $\sqrt{n}$, valdymo taškai parenkami ROLS metodu. Jų projekcija randama MDS metodu. Taškų kandidatų ir valdymo taškų aibės: 1) 200 kandidatų taškų ir 30 valdymo taškų; 2) 300 kandidatų taškų ir 100 valdymo taškų. Branduolys: $\phi''(r) = \sqrt{c^2+(\varepsilon r)^2}$, $c = \varepsilon = 1$.

4.7 lentelėje pateikiami nagrinėtų dimensijų mažinimo metodų rezultatai įvairios apimties duomenų aibėms. Greičiausiai duomenų aibės dimensija sumažinama PLMP metodu, taip pat projekcijos paklaida yra mažiausia lyginant su kitais dviem nagrinėtais metodais. Vidutiniškai ilgiausiai projekcija skaičiuojama LAMP metodu, o gauta projekcijos paklaida yra didesnė arba tokia pati kaip gauta RBF metodu, kai valdymo taškai randami ROLS algoritmu. Iš RBF metodo ir dviejų skirtingų taškų kandidatų ir valdymo taškų aibių analizės nustatyta, kad mažesnė projekcijos paklaida gaunama, kai

valdymo taškų yra $k = 100$, tačiau skaičiavimo laikas šiuo atveju yra ilgesnis. Suprantama, kad dar padidinus kandidatų ir valdymo taškų skaičių projekcijos paklaida sumažėtų, tačiau reikia ieškoti pusiausvyros tarp skaičiavimo laiko ir projekcijos paklaidos.

4.7 lentelė. Dimensijų mažinimo metodų skaičiavimų rezultatai naudojant įvairios apimties duomenų aibes

Random1 **duomenų aibė**	**PLMP** $k = \sqrt{n}$		**LAMP** $k = \sqrt{n}$		**RBF + ROLS** $k = 30$ $N = 200$		**RBF + ROLS** $k = 100$ $N = 300$	
	Paklaida	Laikas	Paklaida	Laikas	Paklaida	Laikas	Paklaida	Laikas
10000×20	**0,38**	**0,32**	0,51	10,39	0,50	7,40	0,40	31,15
10000×100	**0,58**	**0,38**	0,76	19,03	0,76	8,09	0,64	31,40
20000×20	**0,37**	**0,42**	0,51	29,8	0,49	11,91	0,40	33,37
20000×100	**0,55**	**0,78**	0,77	71,65	0,77	14,07	0,65	36,98
50000×20	**0,37**	**1,56**	0,52	136,7	0,48	19,87	0,41	43,84
50000×100	**0,53**	**2,16**	0,77	358,4	0,76	22,04	0,66	49,01
100000×20	**0,36**	**3,36**	0,52	597,82	0,51	33,43	0,42	56,08
100000×100	**0,48**	**6,04**	0,77	1237,32	0,77	39,05	0,63	63,03
500000×20	**0,34**	**45,57**	0,52	8365,34	0,51	147,11	0,40	173,86
500000×100	**0,47**	**48,34**	0.77	25156,56	0,74	158,84	0,64	209,28

PLMP, LAMP ir RBF, kai valdymo taškai parenkami ROLS algoritmu, metodai realizuoti MATLAB programoje taikant ciklą *for*. Šį ciklą galima keisti MATLAB ciklu *parfor*, šis ciklo iteracijas vykdo lygiagrečiai, tai leidžia pagreitinti skaičiavimą. Duomenų dimensijos mažinimo metodų skaičiavimo laiko rezultatai, kai naudojami *for* ir *parfor* ciklai, pateikiami 4.8 lentelėje. PLMP metodo skaičiavimo laikas nestipriai sutrumpėja arba net pailgėja dėl šio metodo algoritmo specifikos. LAMP metodo skaičiavimo laikas vidutiniškai pagreitėja 51 %, kai naudojamas ciklas *parfor* vietoje ciklo *for*. RBF metodu, kai valdymo taškai parenkami ROLS algoritmu, skaičiavimo laikas vidutiniškai pagreitėja 30 % naudojant ciklą *parfor* vietoje ciklo *for*.

Apibendrinimas. Projekcijos metodų, paremtų valdymo taškais, analizė parodė, kad greičiausiai projekcija randama ir mažiausia projekcijos paklaida gaunama PLMP metodu su įvairios apimties duomenų aibėmis. Ilgiausiai skaičiavimai trunka ir gaunama didžiausia projekcijos paklaida LAMP metodu.

4.8 lentelė. Dimensijų mažinimo metodų skaičiavimo laikas naudojant įvairios apimties duomenų aibes

Random1 **duomenų aibė**	**PLMP**			**LAMP**			**RBF+ROLS** $(N = 200;\ k = 30)$		
	for	***parfor***	**Pokytis (proc.)**	***for***	***parfor***	**Pokytis (proc.)**	***for***	***parfor***	**Pokytis (proc.)**
10000×20	**0,32**	0,35	-9,38	10,39	**7,73**	25,60	7,4	**6,76**	8,65
10000×100	**0,38**	0,43	-13,16	19,03	**8,89**	53,28	8,09	**7,17**	11,37
20000×20	**0,42**	0,63	-50,00	29,8	**20,79**	30,23	11,91	**8,6**	27,79
20000×100	**0,78**	0,95	-21,79	71,65	**26,03**	63,67	14,07	**9,05**	35,68
50000×20	**1,56**	1,63	-4,49	136,7	**86,39**	36,80	19,87	**12,79**	35,63
50000×100	**2,16**	2,63	-21,76	358,4	**109,07**	69,57	22,04	**14,36**	34,85
100000×20	**3,36**	3,32	1,19	597,82	**276,2**	53,80	33,43	**20,7**	38,08
100000×100	6,04	**5,01**	17,05	1237,32	**338,43**	72,65	39,05	**23,15**	40,72
500000×20	45,57	**41,14**	9,72	8365,34	X	-	147,11	**91,01**	38,13
500000×100	**48,34**	51,15	-5,81	25156,56	X	-	158,84	**113,06**	28,82

X – nepakanka kompiuterio operatyviosios atminties

4.2.4 Apibendrinti dimensijos mažinimo metodų rezultatai

Išnagrinėjus dimensijos mažinimo metodus, skaičiavimams naudojant disertacijos 4.1 poskyryje aprašytą techninę ir programinę įrangą, nustatyta:

- MDS pasižymi projekcijos tikslumu, t. y. taikant šį metodą gaunama mažiausia projekcijos paklaida, tačiau analizuojant didelės apimties duomenų aibes skaičiavimai trunka ilgai, tad šis metodas gali būti taikomas, kai analizuojamos duomenų aibės iki 5000 taškų. Tokiu atveju skaičiavimai įvykdomi per priimtiną laiką ir trunka iki 1 val. 30 min. Be to, šio metodo skaičiavimų trukmė nedaug priklauso nuo požymių skaičiaus.
- PCA metodas yra optimalus derinant skaičiavimo laiką ir projekcijos paklaidos tikslumą. Taikant šį metodą skaičiavimai atliekami greitai, t. y. jie trunka iki kelių sekundžių, kai duomenų aibės apimtis yra nuo 5000 iki 1000000 taškų. Tačiau šio metodo skaičiavimų trukmė labiau priklauso nuo požymių skaičiaus nei MDS.
- RP metodas pasižymi sparčiu skaičiavimu, kai nagrinėjamos didelės apimties duomenų aibės. RP metodui skaičiavimams atlikti

neprireikia nė sekundės, analizuojant iki 1000000 taškų duomenų aibes. Analizės rezultatai parodė, kad šis metodas nėra tinkamas daugiamačiams duomenims vizualizuoti, nes nėra išlaikoma duomenų struktūra.

- Norint sutaupyti skaičiavimo laiko galima taikyti metodus, kurie taškų projekcijai rasti naudoja panašumų matricą ne tarp visų duomenų aibės taškų, bet tik tarp dalies taškų, t. y. valdymo taškų. Visais nagrinėtais projekcijos paieškos metodais, paremtais valdymo taškais, gaunama projekcijos paklaida tarpusavyje reikšmingai nesiskiria. Renkantis valdymo taškais paremtą projekcijos metodą, tikslinga rinktis PLMP metodą, juo projekcija randama greičiausiai. Šiuo metodu duomenų aibės projekcija, kurios apimtis yra 3000000 taškų, o požymių skaičius yra 100, randama per 6 min.

4.3 Projekcijos kokybės įvertinimo matų tyrimas

Šioje disertacijos dalyje eksperimentiškai ištirti struktūros išlaikymo tarp objektų pradinėje ir sumažintos dimensijos erdvėse kokybės įvertinimo matai, pateikti 2.3 poskyryje: projekcijos paklaida (5), Spirmeno rho koeficientas (6), Konigo topologijos išlaikymo matas (7), silueto koeficientas (8), Renyi entropija (9). Eksperimento tikslas – parodyti, kad norint palyginti skirtingais metodais gautas projekcijas jos turėtų būti vertinamos skirtingas ypatybes atspindinčiais projekcijos kokybės įvertinimo matais. Šiam tikslui pasiekti lyginamos dviem dimensijų mažinimo metodais gautos projekcijos. Lyginimui parinktas populiarus tiesinės projekcijos metodas – PCA ir valdymo taškais pagrįstas PLMP metodas. Šioje tyrimo dalyje analizuojamos šešios duomenų aibės: *Iris*, *Waveform*, *Musk*, *Mammals*, *Helix*, *Swiss roll*.

4.9 lentelėje pateikiami dimensijų mažinimo metodų projekcijos paklaidos, Spirmeno rho koeficiento ir Konigo topologijos išlaikymo reikšmės iš įvairių duomenų aibių analizės. Geriausios matų reikšmės pateikiamos paryškintu

šriftu. PCA metodu gaunama mažesnė projekcijos paklaida nei PLMP metodu su trimis (*Iris*, *Mammals*, *Swiss roll*) iš šešių duomenų aibių.

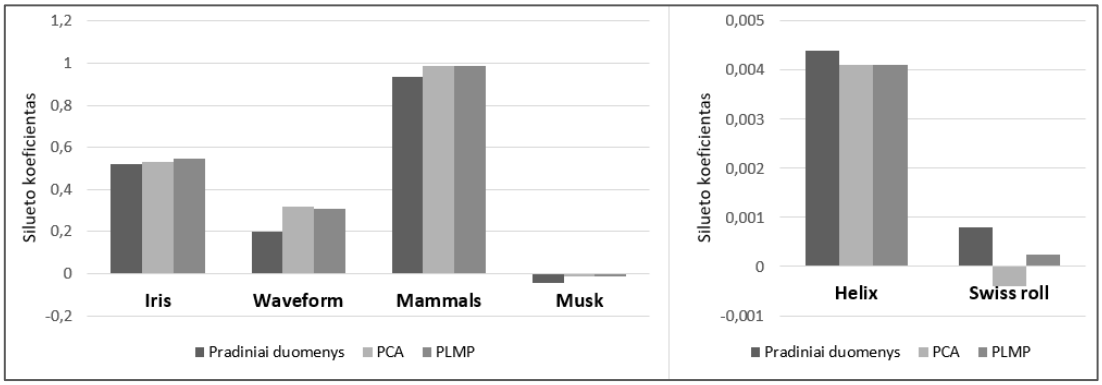
4.9 lentelė. PCA ir PLMP metodais gautų kokybės matų reikšmės naudojant skirtingas duomenų aibes

Duomenų aibė	**Projekcijos paklaida (E_{Stress})**		**Spirmeno rho (ρ)**		**Konigo topologijos išlaikymas (E_{KT})**	
	PCA	**PLMP**	**PCA**	**PLMP**	**PCA**	**PLMP**
Iris [150×4]	**0,0018**	0,005	**0,9935**	0,9927	**0,5533**	0,5467
Waveform [5000×21]	0,0852	**0,0078**	**0,9571**	0,9516	**0,0128**	0,0127
Musk [6598×166]	0,1387	**0,1089**	**0,9262**	0,9146	0,1811	**0,1841**
Mammals [15000×72]	**0,0016**	0,0052	**0,9932**	0,9928	0,3032	**0,3038**
Helix [15000×3]	0,0115	**0,0097**	**0,9788**	0,9785	0,2942	**0,2946**
Swiss roll [15000×3]	**0,0568**	0,2692	**0,8667**	0,6021	**0,3826**	0,3669

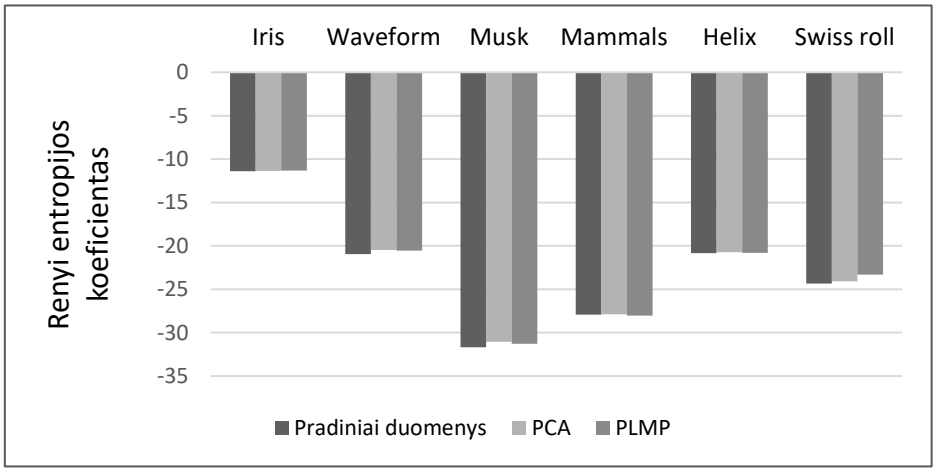
PLMP metodu gauta projekcijos paklaida mažesnė trims (*Waveform*, *Musk*, *Helix*) iš šešių duomenų aibių. O PCA metodu gaunamos projekcijos Spirmeno rho koeficientas, nors ir nestipriai, tačiau yra didesnis už PLMP su visomis duomenų aibėmis. PCA metodu gaunamos projekcijos Konigo topologijos koeficiento reikšmės nedaug didesnės trimis iš šešių duomenų aibių, o PLMP metodu taip pat trimis (*Musk*, *Mammals*, *Helix*) duomenų aibėmis.

Kadangi šioje tyrimo dalyje nebuvo sprendžiamas klasterizavimo uždavinys, vertinant silueto koeficientą klasteriais buvo laikomos duomenų aibių klasės. Silueto koeficiento reikšmės su skirtingomis duomenų aibėmis pateikiamos 4.7 paveiksle. Pastebėtina, kad pradinių duomenų aibių silueto koeficiento reikšmės beveik nepakinta sumažinus pradinių duomenų aibės dimensiją PCA ir PLMP metodais. Labiausiai skiriasi *Waveform* duomenų aibės silueto koeficientas ($S = 0{,}1972$) nuo projekcijos silueto koeficientų (PCA: $S = 0{,}3188$, PLMP: $S = 0{,}307$).

Renyi entropija pakinta nestipriai lyginant pradinių duomenų entropijos koeficientą su dimensijų mažinimo metodais gautos projekcijos entropijos koeficientais su visomis duomenų aibėmis (4.8 paveikslas). Galima teigti, kad sumažinus duomenų aibės dimensiją tiek PCA metodu, tiek PLMP metodu, informacijos kiekis išlieka beveik nepakitęs.



4.7 pav. Silueto koeficiento reikšmės, gautos analizuojant šešias skirtingas duomenų aibes PCA ir PLMP metodais



4.8 pav. Renyi entropijos koeficiento reikšmės, gautos analizuojant šešias skirtingas duomenų aibes PCA ir PLMP metodais

Apibendrinimas. Atliktas tyrimas parodė, kad analizuojant tris iš šešių duomenų aibių mažesnės projekcijos paklaidos ir Konigo topologijos išlaikymo reikšmės gaunamos, kai projekcija randama PCA metodu. O Spirmeno rho koeficiento reikšmės, apskaičiuotos PCA projekcijai, didesnės su visomis duomenų aibėmis. Pagal Renyi entropiją ir silueto koeficientą abu

dimensijų mažinimo metodai yra tinkami informacijai išlaikyti ir priskirti klasteriams sumažinus duomenų aibės dimensiją.

4.4 Pasiūlytų projekcijos paklaidos apskaičiavimo būdų tyrimas

Šiuo tyrimu siekiama eksperimentiškai ištirti disertacijos 3.1 poskyryje pasiūlytus projekcijos paklaidos apskaičiavimo būdus. 1 būdas – projekcijos paklaidą vertinti ne visai duomenų aibei, o tik jos imčiai, 2 būdas – projekcijos paklaidą skaičiuoti visai duomenų aibei, tačiau duomenų aibę skaičiavimų metu padalyti į dalis.

Šioje dalyje aprašytas tyrimas buvo atliktas naudojant dvylika duomenų aibių: *Image segementation*, *Waveform*, *Mammals*, *MAGIC gamma telescope*, *Skin segmentation*, *Shuttle*, *Dspatialnetwork*, *Twinpeaks*, *Helix*, *Swiss roll*, *Crescent and full moon*, *Corners*.

Atliekant eksperimentinį tyrimą ir taikant 1-ąjį būdą kiekvienai duomenų aibei, sudarytai iš iki 50000 taškų, nagrinėtos trijų dydžių imtys, t. y. $n'' = 10000, 5000, 1000$, o duomenų aibėms, sudarytoms nuo 50000 taškų iki 500000 taškų, nagrinėtos šių dydžių imtys: $n'' = 20000, 10000, 5000$. Lyginami du imties sudarymo būdai: atsitiktinis (AI) ir stratifikuotas (SI). Sluoksniai (stratai) nustatomi *k*-vidurkių metodu (angl. *k-means*) [82], [88] arba stratams priskiriamos žinomos klasės. Pradinių duomenų dimensija mažinama PCA metodu iki dviejų $d = 2$.

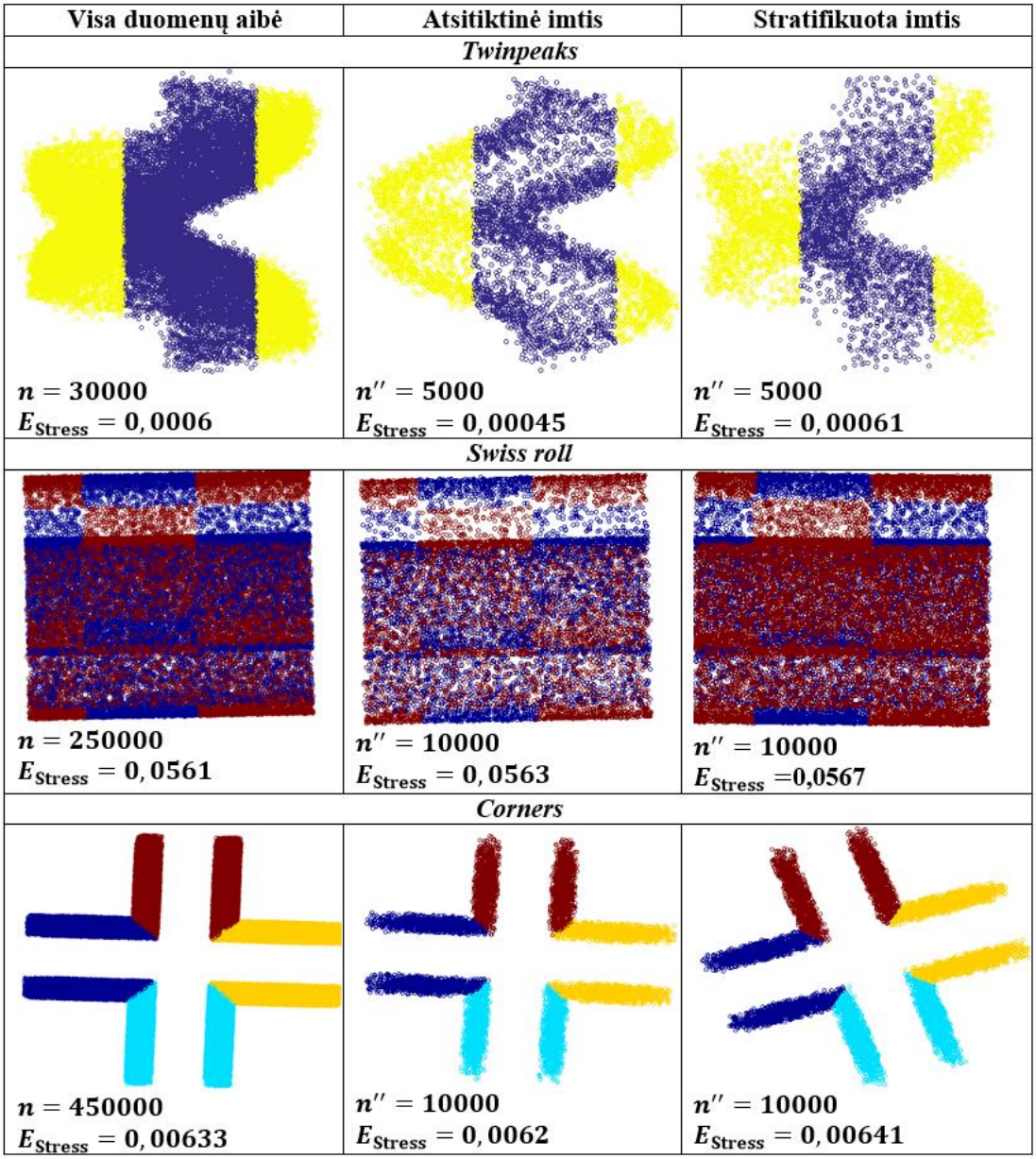
1-asis būdas. 4.10 lentelėje pateiktos įvairių duomenų aibių imčių PCA metodu gautų projekcijų paklaidų reikšmės ir skaičiavimo laikas. Būtina paminėti, kad į projekcijos skaičiavimo laiką klasterizavimo laikas nėra įtraukiamas. Lyginimui projekcijos paklaida suskaičiuota ir visoms duomenų aibėms, ne tik jų imtims. Visų duomenų aibių projekcijų paklaidų reikšmės 4.10 lentelėje nurodytos paryškintu šriftu. Pateikiami ne mažiau kaip keturi ženklai po kablelio, kad matytųsi ir nežymių skirtumų. Projekcijos paklaida apskaičiuojama pagal (5) formulę (žr. 2.3 poskyrį). Dalis iš duomenų aibių sudarytos iš daugiau kaip 30000 taškų, tad joms projekcijos paklaida randama 2-uoju pasiūlytu projekcijos paklaidos būdu (žr. 3.1.2 poskyrį), kadangi kiti

būdai nėra efektyvūs (nepakanka operatyviosios atminties arba skaičiavimai užtrunka ilgai) didelės apimties duomenų aibėms. Iš visų duomenų aibių analizės nustatyta, kad projekcijų paklaidų skirtumas tarp pradinės duomenų aibės ir jos imčių yra nedidelis nepriklausomai nuo to, kokiu būdu sudaryta imtis (ar į imtį įtraukiami objektai atsitiktinai, ar iš pradžių klasterizuojami duomenys ir į imtį įtraukiamas reikiamas objektų kiekis). Pavyzdžiui, nagrinėjant *Helix* duomenų aibę matyti, kad visos duomenų aibės projekcijos paklaida yra 0,0115, atsitiktinės imties, sudarytos iš 5000 taškų, projekcijos paklaida yra 0,0113, o didesnės atsitiktinės duomenų aibės imties, sudarytos iš 20 000 taškų, projekcijos paklaida yra tokia pati, kaip ir visos duomenų aibės – 0,0115. *Helix* duomenų aibės startifikuotos imties sudarytos iš 5000 taškų projekcijos paklaida yra 0,0116. Projekcijos paklaida reikšmingai nesiskiria su skirtingo dydžio duomenų imtimis. O skaičiavimo laikas skiriasi, pavyzdžiui, *Corners* duomenų aibės projekcijos paklaida apskaičiuojama per 1 val. 14 min., jos atsitiktinė imtis, sudaryta iš 20000 taškų, apskaičiuojama per 4,52 s, o projekcijos paklaida atitinkamai yra 0,0629 ir 0,0633.

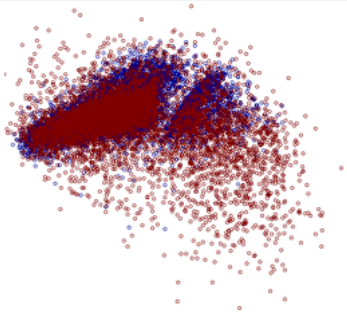
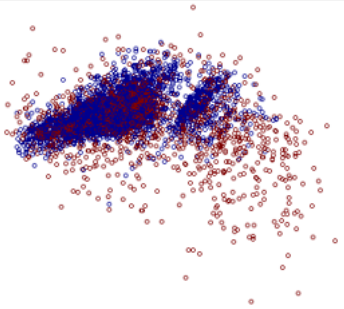
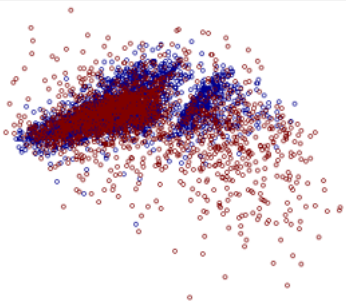
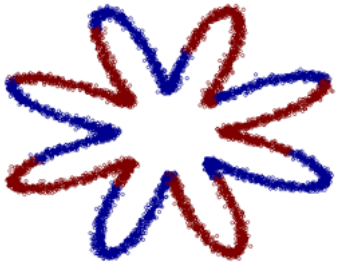
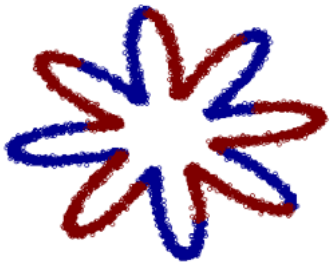
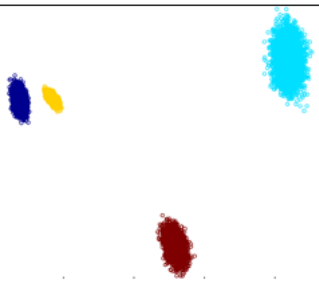
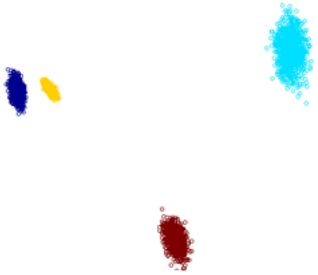
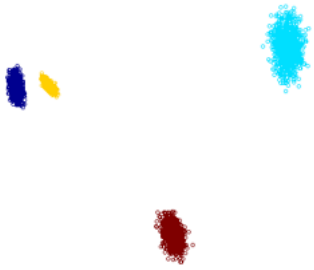
PCA metodu gauti daugiamačių taškų išsidėstymai dvimatėje erdvėje, vizualizuojant *Twinpeaks*, *Swiss roll*, *Corners*, *MAGIC gamma telescope*, *Helix*, *Mammals* duomenų aibes ir jų imtis pateikiama 4.9 ir 4.10 paveiksluose. Kairiajame kampe nurodytas duomenų aibės arba imties dydis ir projekcijos paklaida. Matyti, kad imčių projekcijų taškų išsidėstymo struktūra yra panaši į visos duomenų aibės taškų projekcijos struktūrą analizuojant *Twinpeaks*, *Swiss roll*, *Corners*, *MAGIC gamma telescope*, *Helix*, *Mammals* duomenų aibes. Prarandamas projekcijos paklaidos netikslumas yra nereikšmingas palyginus su sutaupomu laiku. Atliktų eksperimentų rezultatai rodo, kad didelės apimties duomenų aibėms projekcijos paklaidą galima vertinti pagal duomenų aibės imtį.

4.10 lentelė. Projekcijos paklaidos (E_{Stress}) reikšmės ir skaičiavimo laikas analizuojant skirtingas duomenų aibių imtis (n'' – imties dydis)

Duomenų aibė	**(n'')**	**Paklaida (AI)**	**Paklaida (SI)**	**Laikas (AI), s**	**Laikas (SI), s**
Mammals [16384×72]	**16384**	**0,00159**	**0,00159**	**16,20**	**16,20**
	10000	0,00157	0,00159	3,30	3,26
	5000	0,00159	0,00158	0,91	0,87
	1000	0,00155	0,00156	0,08	0,07
MAGIC gamma telescope [19020×10]	**19020**	**0,0665**	**0,0665**	**8,17**	**8,17**
	10000	0,0643	0,0666	1,38	1,44
	5000	0,0643	0,0672	0,48	0,39
	1000	0,0691	0,0672	0,02	0,03
Twinpeaks [30000×3]	**30000**	**0,00059**	**0,00059**	**19,75**	**19,75**
	10000	0,00055	0,00060	1,37	1,28
	5000	0,00045	0,00061	0,35	0,35
	1000	0,00058	0,00059	0,02	0,02
Skin segmentation [51444×3]	**51444**	**0,0093**	**0,0093**	**54,80**	**54,80**
	20000	0,0098	0,0094	4,66	4,87
	10000	0,0097	0,0097	1,51	1,27
	5 000	0,0094	0,0094	0,35	0,32
Helix [250000×3]	**250000**	**0,0115**	**0,0115**	**1 396,80**	**1 396,80**
	20000	0,0115	0,0113	4,43	4,44
	10000	0,0113	0,0114	1,28	1,28
	5000	0,0113	0,0116	0,36	0,36
Swiss roll [250000×3]	**250000**	**0,0561**	**0,0561**	**1 363,80**	**1 363,80**
	20000	0,0560	0,0571	4,50	4,54
	10000	0,0563	0,0567	1,61	1,64
	5000	0,0562	0,0556	0,37	0,35
Crescent and full moon [300000×4]	**300000**	**0,0676**	**0,0676**	**2 007,80**	**2 007,80**
	20000	0,0680	0,0679	4,38	4,45
	10000	0,0682	0,0690	1,31	1,32
	5000	0,0679	0,0664	0,38	0,38
Dspatialnetwork [434874×3]	**434874**	**$9{,}1534\times10^{-7}$**	**$9{,}1534\times10^{-7}$**	**4 610,50**	**4 610,50**
	20000	$9{,}3305\times10^{-7}$	$9{,}2155\times10^{-7}$	4,18	4,41
	10000	$9{,}1543\times10^{-7}$	$9{,}3122\times10^{-7}$	1,33	1,40
	5000	$9{,}1433\times10^{-7}$	$9{,}2613\times10^{-7}$	0,38	0,39
Corners [450000×4]	**450000**	**0,00633**	**0,00633**	**4 455,40**	**4 455,40**
	20000	0,00629	0,00632	4,52	4,42
	10000	0,00621	0,00641	1,58	1,27
	5000	0,00618	0,00629	0,44	0,42



4.9 pav. Visų duomenų aibių ir jų imčių dvimačiai vaizdai (I)

Visa duomenų aibė	Atsitiktinė imtis	Stratifikuota imtis
MAGIC gamma telescope		
 $n = 19020$ $E_{\text{Stress}} = 0,0665$	 $n'' = 5000$ $E_{\text{Stress}} = 0,0643$	 $n'' = 5000$ $E_{\text{Stress}} = 0,0672$
Helix		
 $n = 250000$ $E_{\text{Stress}} = 0,0115$	 $n'' = 10000$ $E_{\text{Stress}} = 0,0113$	 $n'' = 10000$ $E_{\text{Stress}} = 0,0114$
Mammals		
 $n = 16384$ $E_{\text{Stress}} = 0,00159$	 $n'' = 5000$ $E_{\text{Stress}} =0,0016$	 $n'' = 5000$ $E_{\text{Stress}} =0,00158$

4.10 pav. Visų duomenų aibių ir jų imčių dvimačiai vaizdai (II)

4.11 lentelė. Projekcijos paklaidos skaičiavimo laikas (s), analizuojant dvyliką skirtingų duomenų aibių: (a) duomenų aibė dalijama į dalis (2-asis būdas); (b) naudojamas ciklas *for*, kuriame paklaida skaičiuojama panariui kiekvienam duomenų aibės taškui; (c) atstumai randami naudojant MATLAB funkciją *pdist*

Duomenų aibė	**(a)**	**(b)**	**(c)**
Image segmentation [2310×19]	0,35	0,41	0,12
Waveform [5000×21]	2,13	2,93	0,67
Mammals [16384×72]	48,61	233,47	16,20
MAGIC gamma telescope [19020×10]	8,92	45,04	8,17
Twinpeaks [30000×3]	50,70	40,53	19,75
Skin segmentation [51444×3]	54,80	153,72	X
Shuttle [58000×9]	74,60	510,59	X
Helix [250000×3]	1396,80	7184,0	X
Swiss roll [250000×3]	1363,8	7010,7	X
Crescent and full moon [300000×4]	2007,8	10855,0	X
Dspatialnetwork [434874×3]	4610,5	19957,0	X
Corners [450000×4]	4455,4	26280,0	X

X – nepakanka kompiuterio operatyviosios atminties

2-asis būdas. 4.11 lentelėje pateikiami projekcijos paklaidos skaičiavimo greitaveikos rezultatai, kai projekcijos paklaida skaičiuojama duomenų aibę dalijant į dalis. Kaip palyginimas 4.11 lentelėje pateikiamas projekcijos paklaidos skaičiavimo laikas, kai projekcijos paklaidai apskaičiuoti naudojamas ciklas *for*, jame paklaida skaičiuojama panariui kiekvienam duomenų aibės taškui, o atstumai tarp taškų randami taikant MATLAB funkciją *pdist* ir nedalijant duomenų aibių. Atliekant eksperimentinį tyrimą nagrinėjamos duomenų aibės buvo dalijamos į mažesnes – 10000 objektų aibes (duomenų aibės sudarytos iš 5000 objektų ir mažesnės dalijamos į 500 objektų aibes).

Greičiausiai projekcijos paklaida apskaičiuojama, kai taikoma funkcija *pdist* ir nedalijama duomenų aibė arba kai taikomos *pdist* ir *pdist2* funkcijos ir dalijama duomenų aibė į dalis (2-asis būdas). Skirtumas paaiškinamas tuo, kad MATLAB programa turi bibliotekų, skirtų greitai atlikti veiksmams su matricomis ir vektoriais. Pastebėta, kad kai skaičiuojama projekcijos paklaida nedalijant duomenų aibės į dalis ir taikant funkciją *pdist*, jei objektų skaičius

didesnis kaip 30000, nepakanka esamos 12 GB kompiuterio operatyvios atminties. Ilgiausiai projekcijos paklaida skaičiuojama taikant ciklą *for*, nors jis ir nereikalauja daug kompiuterio operatyviosios atminties, nes jame paklaida skaičiuojama panariui kiekvienam duomenų aibės taškui. Šiuo metodu analizuojant *Corners* duomenų aibę, sudarytą iš 450000 objektų, projekcijos paklaida skaičiuojama ilgiau kaip 7 val. 18 min., tad šis būdas nėra tinkamas didelės apimties duomenų aibėms. Dėl duomenų aibės dalijimo skaičiavimų metu projekcijos paklaida apskaičiuojama 450000 objektų duomenų aibei per 1 val. 14 min., t. y. beveik 6 kartus greičiau nei taikant ciklą *for*, jame paklaida skaičiuojama panariui kiekvienam duomenų aibės taškui.

Svarbu paminėti, kad vietoje ciklo *for* gali būti naudojamas ciklas *parfor*, šis skaičiavimus išlygiagretina. Projekcijos paklaidos skaičiavimo laikas analizuojant *Random1* duomenų aibes, kai taikomi MATLAB ciklai *for* ir *parfor*, pateikiamas 4.12 lentelėje.

4.12 lentelė. Projekcijos paklaidos skaičiavimų laikas (s) naudojant įvairios apimties duomenų aibes: (a) naudojamas ciklas, kuriame paklaida skaičiuojama panariui kiekvienam duomenų aibės taškui, (b) duomenų aibė dalijama į dalis (2-asis būdas)

Duomenų aibė	**(a)**			**(b)**			**Projekcijos paklaida (E_{Stress})**
	for	***parfor***	**Pokytis (%)**	***for***	***parfor***	**Pokytis (%)**	
20000×50	222,73	**161,64**	27,43	17,66	**14,52**	17,78	0,4639
20000×100	423,53	**323,61**	23,59	29,75	**24,4**	17,98	0,5510
40000×50	900,69	**666,37**	26,02	87,94	**71,69**	18,48	0,4499
40000×100	1722,26	**1260,32**	26,82	139,81	**123,09**	11,96	0,5357
60000×50	2026,97	**1518,8**	25,07	191,68	**161,34**	15,83	0,4661
60000×100	3899,77	**2866,62**	26,49	286,06	**236,22**	17,42	0,5203
80000×50	3635,94	**2668,12**	26,62	349,48	**260,39**	25,49	0,4684
80000×100	6951,34	**5154,29**	25,85	522,6	**424,99**	18,68	0,5039
100000×50	5721,35	**4279,2**	25,21	556,0,1	**440,25**	20,82	0,4277
100000×100	11063,07	**8116,12**	26,64	842,23	**682,9**	18,92	0,4817
150000×50	13006,21	**9852,35**	24,25	1262,09	**963,57**	23,65	0,4514
150000×100	25297,25	**18716,47**	26,01	1985,95	**1560,36**	21,43	0,4930

Projekcijos paklaidos, kai taikomas ciklas *parfor*, kuriame paklaida skaičiuojama panariui kiekvienam duomenų aibės taškui, skaičiavimo laikas vidutiniškai pagreitėja 26 %. Kai dalijama duomenų aibė į mažesnes duomenų aibes taikant *parfor* funkciją, projekcijos paklaidos skaičiavimo laikas vidutiniškai pagreitėja 19 %.

Apibendrinimas. Šiame tyrime ištirti du pasiūlyti projekcijos paklaidos skaičiavimo būdai, t. y. 1-uoju būdu siūloma projekcijos paklaidą vertinti iš sudarytos duomenų aibės imties; 2-uoju būdu duomenų aibė skaičiavimų metu dalijama į dalis. Eksperimentinis tyrimas parodė, kad didelės apimties duomenų aibėms projekcijos paklaida gali būti vertinama duomenų aibės imčiai, t. y. projekcijos paklaidos skirtumas tarp visos duomenų aibės ir jos imčių yra nereikšmingas su visomis šiame tyrime tirtomis duomenų aibėmis. Šis būdas leidžia greičiau gauti rezultatą, reikia mažiau skaičiavimo laiko. Nustatyta, kad dalijant duomenų aibę į mažesnes dalis, projekcijos paklaida apskaičiuojama 450000 objektų duomenų aibei. Derinant du pasiūlytus būdus, projekcijos paklaida galėtų būti apskaičiuota ir labai didelės apimties duomenų aibėms, kurių imtis būtų sudaryta iš 450000 taškų. Projekcijos paklaida būtų tiksli, būtų rasta per priimtiną laiką (1 val. 14 min.), be to, neiškiltų nepakankamos kompiuterio operatyviosios atminties problema. Didelės apimties duomenų aibėms šių pasiūlytų būdų derinys galėtų būti įgyvendintas taip: pirma – randama duomenų aibės imtis; antra – skaičiavimų metu dalijant duomenų aibės imtį į dalis būtų apskaičiuota projekcijos paklaida. Abu pasiūlyti būdai galėtų būti pritaikyti ir skaičiuojant kitus projekcijos kokybės įvertinimo matus, kuriuose yra taikomos didelės apimties atstumų matricos.

4.5 Pasiūlytos duomenų vizualizavimo strategijos tyrimas

Šio tyrimo tikslas – ištirti 3.2 poskyryje pasiūlytą duomenų vizualizavimo strategiją bei ją palyginti su kitais vizualizavimo sprendimais.

4.5.1 Pasiūlytos duomenų vizualizavimo strategijos tyrimas

Šioje tyrimo dalyje analizuojamos šios duomenų aibės: *MAGIC gamma telescope*, *Waveform*, *Wine quality*, *Letter recognition*, *Musk*, *Mammals*, *Skin segmentation*, *Image segmentation*, *Helix*, *Yeast*, *Random3*, *Random4*. Iš pradžių tiriamas 1-ajame etape pasiūlytas duomenų aibės imties sudarymo būdas. Ši imtis 2-ajame etape vizualizuojama. Daugiamačiai duomenys klasterizuojami *k-medoids* metodu [82]. Klasterių skaičius τ nustatomas pagal *Calinski-Harabasz* klasterių nustatymo kriterijų [89]. Daugiamačių duomenų aibės imties dimensijai sumažinti taikomas MDS metodas. Dimensija mažinama iki dviejų ($d = 2$). 1-ajame etape pasiūlytas imties sudarymo būdas lyginamas su stratifikuotu imties sudarymo būdu, jo metu duomenų aibė suskirstoma į sluoksnius (stratus). Kiekvienam sluoksniui taikomas paprastosios atsitiktinės imties sudarymo būdas [87]. Šiame tyrime stratai (sluoksniai) yra nustatomi *k-medoids* metodu. Dėl paprastumo eksperimentiniame tyrime imties dydis $s = 1000$.

4.13 lentelėje pateikiami *stratifikuotos* imties sudarymo būdo ir *pasiūlyto būdo duomenų* imčiai sudaryti lyginimo rezultatai su keturiomis duomenų aibėmis (*Random3*, *Magic gamma*, *Musk*, *Random4*). Stulpeliuose „Imtis (pasiūlytas būdas)“ ir „Imtis (stratifikuota)“ pateikiama informacija, kiek taškų (N_i') turėtų būti atrinkta iš kiekvieno klasterio į imtį. Atliktų skaičiavimų lyginimas parodė, kad taškų skaičius, atrinktas iš kiekvieno klasterio į imtį, skiriasi priklausomai nuo to, kokiu būdu sudaryta imtis, ypač kai nagrinėjamos *Random3* ir *Magic gamma* duomenų aibės (4.16 lentelė). *Random3* visos duomenų aibės ir jos imčių, sudarytų skirtingais būdais, vaizdai pateikiami 4.11 paveiksle. Vizualizavimo rezultatai parodė, kad *pasiūlytu būdu sudarytos imties* taškų pasiskirstymas panašus į visos *Random3* duomenų aibės, o *stratifikuota imtis* neišlaiko trečio klasterio struktūros (raudoni taškai), t. y. tik vienas taškas parenkamas iš trečiojo klasterio. Kai analizuojama *Magic gamma* duomenų aibė, 526 taškai (žalios spalvos) yra atrenkami į imtį, taikant *pasiūlytą imties sudarymo būdą*, ir 194 – taikant *stratifikuotą imties* būdą. Iš

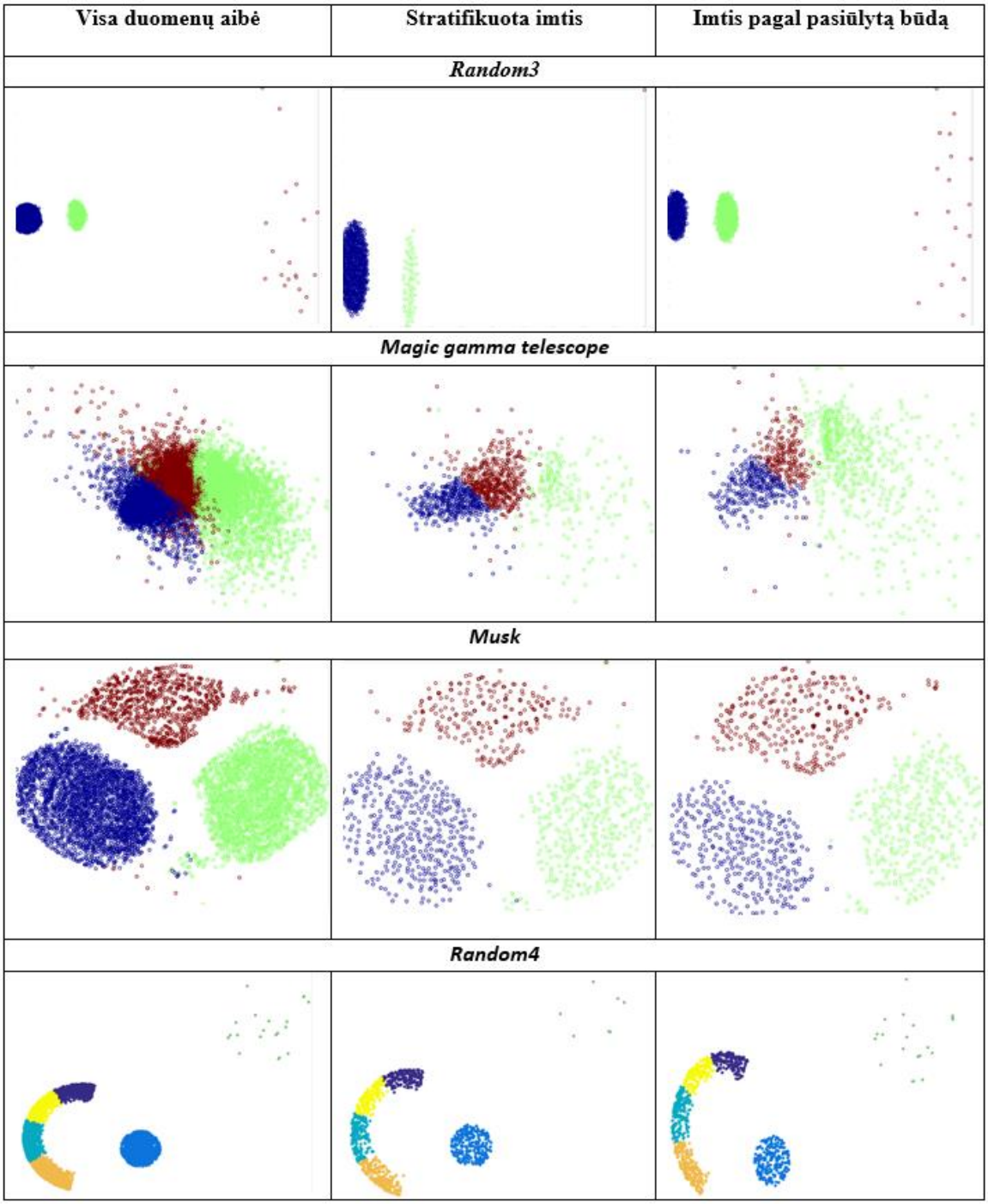
trečiojo klasterio (raudoni taškai) atitinkamai atrenkami 195 ir 424 taškai. Taikant *pasiūlytą imties sudarymo būdą* atsižvelgiama į taškų tankumą: kuo didesnis santykis r_i, tuo daugiau taškų yra parenkama į imtį iš klasterio ir priešingai. Analizuojant *Musk* duomenų aibės imtis, sudarytas dviem skirtingais būdais, struktūra skiriasi nereikšmingai: taikant pasiūlytą būdą iš atitinkamų klasterių atrenkami $323, 335, 342$ taškai; taikant stratifikuotos imties sudarymo būdą atitinkamai atrenkami $383, 354, 263$ taškai (4.13 lentelė). Abiem būdais sudarytų duomenų aibių imčių struktūra nedaug skiriasi, santykis r_i ir taškų skaičius klasteryje panašus į visų klasterių.

4.13 lentelė. Imties dydžio apskaičiavimas naudojant skirtingus imties sudarymo būdus

Duomenų aibė	**Klasteris (i)**	**Taškų skaičius klasteryje (N_i)**	**D_i**	**Santykis (r_i)**	**Imtis *(pasiūlytas būdas)***	**Imtis *(strati-fikuota)***
Random3 [15020×10]	1	14000	13410	0,96	125 (468**)	932
	2	1000	1048	1,05	136 (512**)	67
	3	20	114	6	739 (20*)	1
Magic gamma telescope [18905×10]	1	7218	533352	13969	279	382
	2	3672	511559	26342	526	194
	3	8015	413549	9754	195	424
Musk [6581×166]	1	2518	2259955	898	323	383
	2	2332	2172782	932	335	354
	3	1731	1644631	950	342	263
Random4 [3020×4]	1	630	$5{,}3\times10^{14}$	$8{,}4\times10^{11}$	51 (219**)	209
	2	459	$3{,}5\times10^{14}$	$7{,}7\times10^{11}$	47 (200**)	152
	3	750	$3{,}7\times10^{14}$	$5{,}0\times10^{11}$	30 (130**)	248
	4	528	$4{,}2\times10^{14}$	$8{,}0\times10^{11}$	49 (209**)	175
	5	633	$5{,}4\times10^{14}$	$8{,}5\times10^{11}$	52 (222**)	210
	6	20	$2{,}5\times10^{14}$	$1{,}3\times10^{13}$	751 (20*)	7

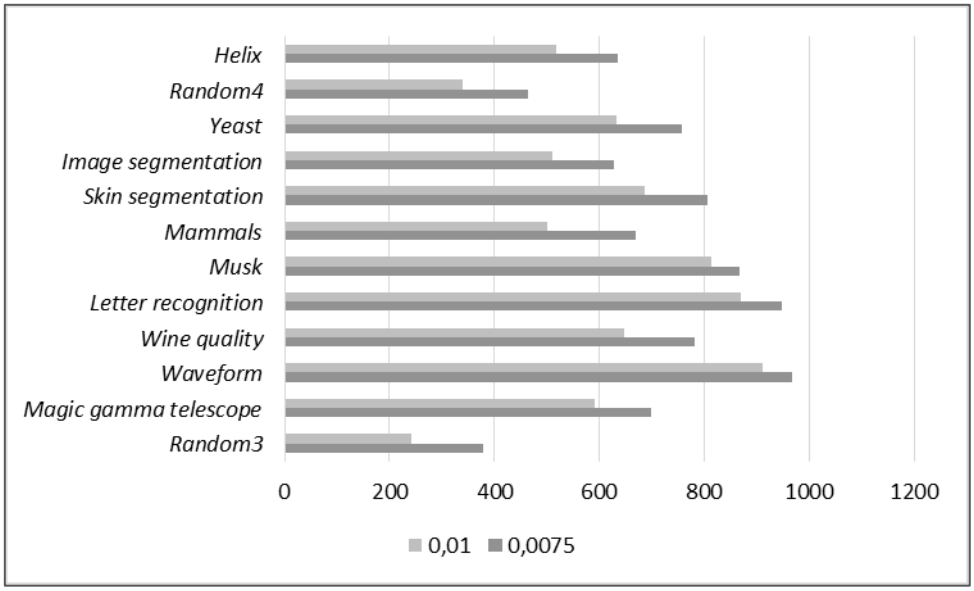
*– kai apskaičiuotas išrenkamų taškų skaičius N_i' yra didesnis už faktinį taškų skaičių N_i klasteryje, tada į imtį įtraukiami visi to klasterio taškai

**– taškų skaičius po perskirstymo



4.11 pav. Vizualizavimo rezultatų palyginimas naudojant skirtingus duomenų aibės imties sudarymo būdus

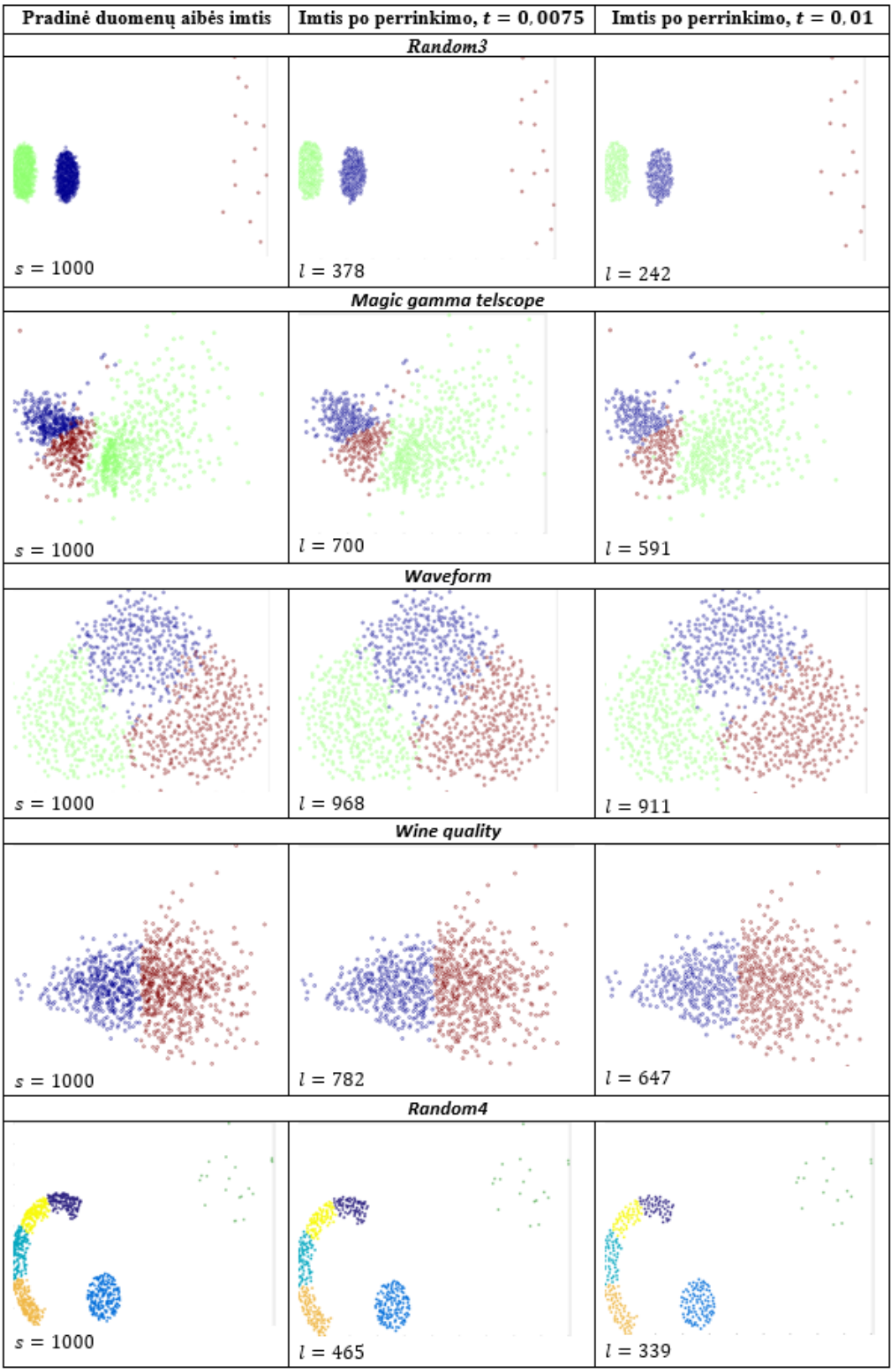
2-uoju vizualizavimo strategijos etapu siūloma pašalinti taškų persidengimą. Šiame tyrime eksperimentiškai ištirtos dvi slenksčio t reikšmės, su kuriomis pašalinamas taškų persidengimas. 4.12 paveiksle matyti dviejų skirtingų slenksčio reikšmių $t = 0{,}01$, $t = 0{,}0075$ lyginimas analizuojant dvylika skirtingų duomenų aibių. Visų duomenų aibių imčių dydis yra $s = 1000$. Duomenų aibių imčių projekcijos sunormuotos intervale (0; 1). 4.12 paveiksle pateikiama, kiek taškų turėtų būti vizualizuota iš kiekvienos duomenų aibės, kad taškai nepersidengtų. Akivaizdu, kad su didesne slenksčio reikšme taškų lieka mažiau, tačiau tai priklauso nuo konkrečios duomenų aibės.



4.12 pav. Dviejų slenksčio reikšmių lyginimas ($t = 0{,}01$, $t = 0{,}0075$)

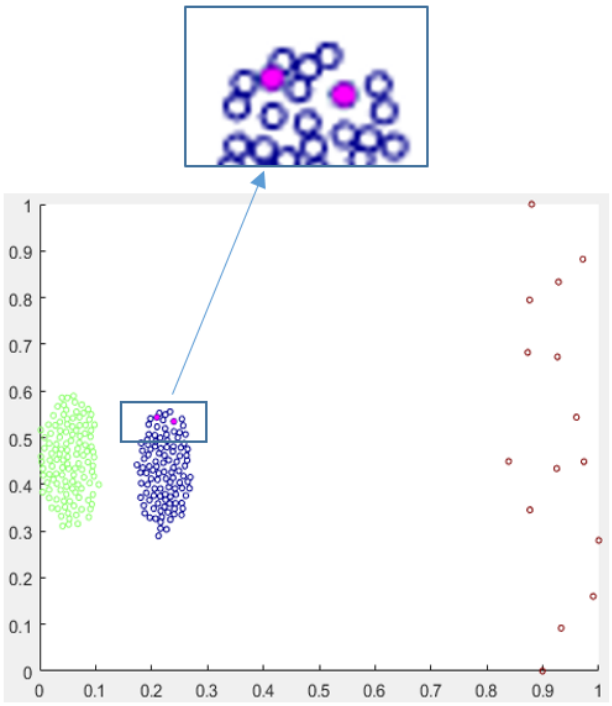
Kai kurių duomenų aibių imčių vaizdai po perrinkimo su tam tikra slenksčio reikšme pateikiami 4.13 paveiksle. Tyrimo rezultatai parodė, kad persidengimas šalinamas su $t = 0{,}01$ visoms tyrimo duomenų aibėms.

Pažymėtina, kad taikant siūlomą vizualizavimo strategiją vizualizuojami ne visi duomenų aibės taškai. Jei domina taškas, neatrinktas į imtį ir todėl nevizualizuotas, galima rasti jo artimiausią vizualizuotą kaimyną daugiamatėje erdvėje ir projekciją dvimatėje erdvėje. Artimiausio kaimyno vieta vaizde parodo, kur turėtų būti dominantis taškas.



4.13 pav. Vizualizavimo rezultatai, kai imtis perrenkama su tam tikra slenksčio reikšme. Imties dydis nurodytas apatiniame kairiajame kampe

4.14 paveiksle pateikiamas pavyzdys, kaip nustatyti dviejų dominančių taškų, neįtrauktų į imtį ir nevizualizuotų, vietą vaizde. Čia artimiausi dominančių taškų kaimynai atvaizduoti purpurine spalva. Tai reiškia kad dominantys taškai yra netoli pažymėtų taškų (kaimynų).



4.14 pav. Dviejų dominančių taškų artimiausi kaimynai

Apibendrinimas. Eksperimentiškai parodyta, kad taikant pasiūlytą imties sudarymo būdą išlaikoma duomenų struktūra. Parodyta, kad vizualizavimui užtenka imties, sudarytos iš 1000 ir mažiau taškų, gerai reprezentuojančios didelės apimties duomenų aibę. Eksperimentiškai ištirtos dvi slenksčio reikšmės, kurios naikina taškų persidengimą. Pateiktas pavyzdys, kaip rasti tam tikro taško, neatrinkto į imtį ir nevizualizuoto, vietą dvimatėje erdvėje. Pasiūlyta vizualizavimo yra strategija taikytina didelės apimties duomenų aibėms vizualizuoti, kai vizualizuojant duomenų aibę reikia pamatyti tik visą duomenų struktūrą ir retus klasterius.

4.5.2 Vizualizavimo strategijų lyginimas

Šiame poskyryje pateikiamas pasiūlytos vizualizavimo strategijos lyginimas su kitų tyrėjų pasiūlytais sprendimais, kai didelės apimties duomenys vizualizuojami sklaidos diagramoje ir taškai vienas kitą perdengia. Vertinant skirtingus aspektus nagrinėjami šie sprendimai: apibendrinta sklaidos diagrama (angl. *generalized scatter plot*) [69], sudėtinė sklaidos diagrama (angl. *stacking graphic*) [70], kontūruota sklaidos diagrama (angl. *splaterrplots*) [71], padalytų kintamųjų sklaidos diagrama (angl. *variable binned scatter plot*) [72].

4.14 lentelė. Vizualizavimo metodų lyginimas su pasiūlyta duomenų vizualizavimo strategija

Aspektai	Egzistuojantys sprendimai	Pasiūlyta strategija
Dimensijos mažinimas	[69], [70], [71], [72] nėra atsižvelgiama į dimensijos mažinimą, nagrinėjami dvimačiai duomenys arba vizualizuojamos daugiamačių požymių poros. Nors [70] dimensijos mažinimas nėra taikomas, tačiau sprendimai pritaikyti daugiamačiams duomenims juos vizualizuojant tiesioginio vizualizavimo metodu – lygiagrečiosiomis koordinatėmis.	Įtrauktas dimensijos mažinimas, todėl strategija yra tinkama daugiamačiams duomenims vizualizuoti.
Persidengiančių taškų problema	[69] persidengiančių taškų problema sprendžiama deformuojant sklaidos diagramos koordinates. Tokiu būdu tankiau išsidėsčiusiems taškams skiriama daugiau erdvės, o rečiau – mažiau. [70] persidengiančių taškų problema sprendžiama įvedant papildomą dimensiją. [71] didelio tankio taškų sritys vaizduojamos kaip kontūru apriboti plotai, pažymėti tam tikra spalva, o	Persidengiančių taškų problema sprendžiama vizualizuojant dalį taškų, t. y. vizualizuojama duomenų aibės imtis.

Aspektai	Egzistuojantys sprendimai	Pasiūlyta strategija
	šios intensyvumas parodo taškų tankį srityje. Mažo tankio taškų srityse vaizduojamų taškų skaičius priklauso nuo mažiausio nustatyto atstumo tarp taškų ir mažiausio atstumo tarp taškų ir kontūro linijos. Šie atstumai nustatomi atsižvelgiant į ekrano dydį. [72] persidengiančių taškų problema sprendžiama padalijant x ir y dimensijas į intervalus, o tada kiekvienas taškas atvaizduojamas į atitinkamus stačiakampius. Kiekvieną tašką atitinka vienas pikselis. Taip pat įvedama trečia dimensija norint gauti vizualinį klasterizavimą.	
Vaizdo priartinimas	[69] vaizdo priartinimas yra taikomas. [70] nėra taikomas. [71] vaizdo priartinimas gali būti taikomas. [72] vaizdo priartinimas gali būti taikomas.	Nėra taikomas.
Išskirtys	[69] išskirčių problema nenagrinėjama. [70] išskirtys gali būti pažymėtos kita spalva ar forma. [71] išskirtys yra išsaugomos. [72] išskirtys yra išsaugomos.	Strategijoje siekiama išlaikyti retų klasterių taškus ir struktūrą bei taškus, nutolusius nuo daugumos taškų.
Duomenų apimtis	[69] atlikti skaičiavimai didelės apimties duomenų aibėms iki 69056 taškų. [70] didelės apimties duomenų aibės nenagrinėjamos. [71] atlikti skaičiavimai didelės apimties duomenų aibėms, sudarytoms iš daugiau kaip 500000 taškų.	Tinka didelės apimties duomenų aibėms iki 750000 taškų.

Aspektai	Egzistuojantys sprendimai	Pasiūlyta strategija
	[72] atlikti skaičiavimai didelės apimties duomenų aibėms iki 70465 taškų.	
Kiekvieno taško identifikavimas	[69], [70] nėra galimybės identifikuoti kiekvieno taško pozicijos. [71] galima identifikuoti kiekvieno taško poziciją. [72] reliatyvi taško pozicija gali būti identifikuota.	Galima identifikuoti kiekvieną tašką, be to, jei domina taškas, neatrinktas į imtį ir nevizualizuotas, galima rasti jo artimiausią vizualizuotą kaimyną daugiamatėje erdvėje ir jo projekciją dvimatėje erdvėje. Artimiausio kaimyno vieta vaizde rodo, kur turėtų būti dominantis taškas.

Iš 4.14 lentelės matyti, kad pasiūlyta vizualizavimo strategija yra vienintelė, į kurią yra įtraukiamas dimensijos mažinimas, be to, matyti bendra duomenų struktūra ir galima identifikuoti kiekvieną tašką. Darbuose [69], [71], [72] siūlomos vaizdo pateikimo formos nėra intuityviai suprantamos lyginant su įprasta sklaidos diagrama. Apibendrintoje sklaidos diagramoje deformacijas sudėtinga suprasti, o duomenis vizualizuojant sudėtinėje sklaidos diagramoje ir nagrinėjant didelės apimties duomenų aibėmis susiduriama su mastelio išlaikymu. Kai kontūruojamoje sklaidos diagramoje yra didelis kontūruojamų sričių persidengimas, vaizdas tampa neinformatyvus. Lyginamoji analizė parodė, kad pasiūlyta vizualizavimo strategija turi ypatybių, dėl kurių ji yra pranašesnė už kitų autorių sprendimus.

4.6 Ketvirtojo skyriaus apibendrinimas

Šiame skyriuje atlikta MDS, PCA, ICA, RP, LAMP, PLMP ir RBF metodų lyginamoji analizė parodė, kad taikyti dimensijų mažinimo metodai greitai apdoroja (neužtrunka nė minutės) įvairios apimties (150–1000000 objektų)

duomenų aibes, išskyrus MDS ir LAMP metodus, tačiau pastarieji metodai pasižymi mažiausia projekcijos paklaida iš nagrinėtų projekcijos metodų. Dimensijos mažinimo metodai, paremti valdymo taškais, kai skaičiuojama projekcija, nenaudoja atstumų tarp visų duomenų aibės taškų, o tik tarp dalies taškų, taip sutrumpinamas projekcijos paieškos skaičiavimo laikas. RP metodas patrauklus dėl savo paprastumo ir skaičiavimo greičio, tačiau daugiamačiams duomenims vizualizuoti nėra tinkamas.

Eksperimentiškai parodyta, kad lyginti skirtingais metodais gautas projekcijas reikia lyginti keliais projekcijos kokybės įvertinimo matais, siekiant nustatyti, ar dimensijos mažinimas išlaiko duomenų ypatybes.

Eksperimentinis tyrimas su įvairiomis duomenų aibėmis parodė, kad didelės apimties duomenų aibių projekcijos paklaida gali būti vertinama duomenų aibės imčiai, t. y. projekcijos paklaidos skirtumas tarp visos duomenų aibės ir jos imčių yra nedidelis su visomis duomenų aibėmis. Be to, šis būdas leidžia sutrumpinti skaičiavimo laiką.

Nustatyta, kad dalijant duomenų aibę į mažesnes dalis projekcijos paklaida apskaičiuojama 450000 objektų duomenų aibei per tinkamą laiką (1 val. 14 min.). Skaičiavimai sutrumpėja beveik 6 kartus, kai 450000 objektų duomenų aibei projekcijos paklaida skaičiuojama dalijant duomenų aibę į dalis (4455,4 s) palyginus su projekcijos skaičiavimo būdu, taikančių ciklą, kuriame paklaida skaičiuojama panariui kiekvienam duomenų aibės taškui (26280,0 s). Pasiūlytu projekcijos paklaidos skaičiavimo būdu projekcijos paklaida gali būti randama 15 kartų $\left(\frac{450000\ \text{objektų}}{30000\ \text{objektų}}\right)$ didesnei duomenų aibei, palyginus su projekcijos paklaidos apskaičiavimu taikant būdą, kuriame naudojama nedalyta duomenų aibė ir speciali funkcija, pritaikyta greitam atstumų skaičiavimui. Pasiūlyti projekcijos paklaidos vertinimo būdai galėtų būti pritaikyti ir kitiems projekcijos kokybės matams skaičiuoti.

Eksperimentiškai ištirta nauja vizualizavimo strategija, sudaryta iš dviejų etapų: duomenų aibės imties sudarymas; imties taškų vizualizavimas be persidengimo. Parodyta, kad vizualizavimui užtenka imties, sudarytos iki 1000

taškų, gerai atskleidžiančios bendrą didelės apimties duomenų struktūrą. Eksperimentiškai parodyta, kad taikant pasiūlytą imties sudarymo būdą išlaikoma retų klasterių ir duomenų aibės struktūra. Pasiūlytos dvi slenksčio reikšmės, su kuriomis gali būti pašalintas taškų persidengimas.

5 Pasiūlytų sprendimų taikymas meteorologinių duomenų aibės analizei

Disertacijoje pasiūlyti sprendimai (projekcijos paklaidos apskaičiavimas didelės apimties duomenų aibėms (3.1 poskyris) ir vizualizavimo strategija (3.2 poskyris) pritaikyti analizuojant meteorologinių duomenų aibę. Šiame skyriuje aprašyti gauti rezultatai.

5.1 Duomenų aprašymas

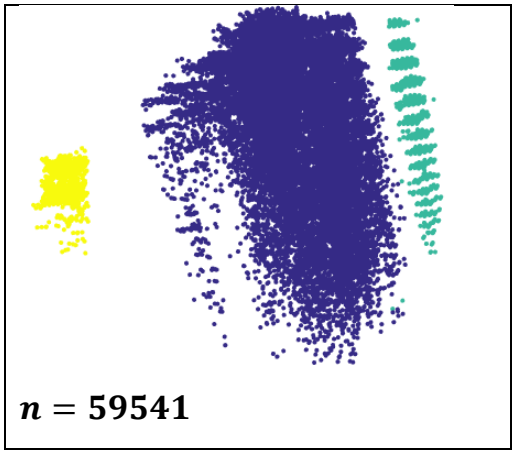
Šioje disertacijos dalyje pateikiamo tyrimo tikslas – parodyti visus disertacijoje pasiūlytus sprendimus sprendžiant realų duomenų analizės uždavinį, t. y. apskaičiuoti projekcijos paklaidą didelės apimties duomenims ir juos vizualizuoti.

Šiame skyriuje nagrinėjama meteorologinių duomenų aibė, kuri buvo sudaryta remiantis *Weather Underground* (https://www.wunderground.com/) duomenimis. *Weather Underground* teikia trumpalaikes ir ilgalaikes pasaulio orų prognozes, orų ataskaitas ir žemėlapius, taip pat meteorologinių stočių istorinius duomenis. Pasiūlytiems sprendimams pritaikyti pasirinkti trijų meteorologinių stočių (Vilnius (EYVI), Singapūras (WSAP), Jakutskas (UEEE) matavimai, atlikti 2016–2017 m. Meteorologinių duomenų aibę nusako šie septyni požymiai:

- oro temperatūra (C),
- rasos taškas (C),
- drėgmė (%),
- jutiminė temperatūra (C),
- vėjo greitis (km/h),
- slėgis (pHa),
- matomumas (km).

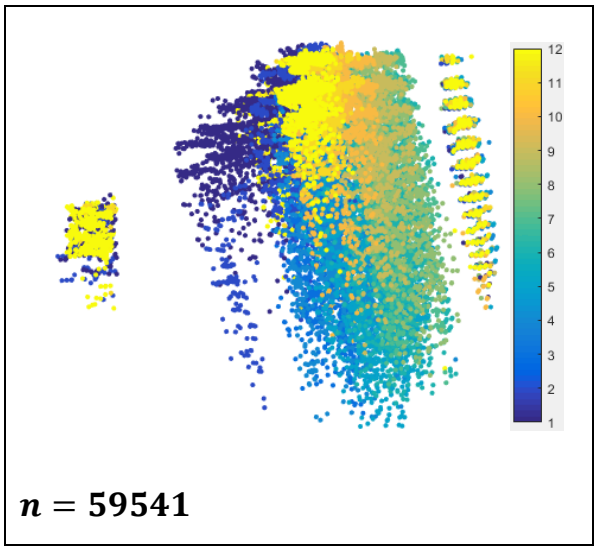
40939 matavimai buvo iš Vilniaus (EYVI) meteorologinės stoties, 15680 matavimai iš Singapūro (WSAP) meteorologinės stoties ir 2922 iš Jakutsko (UEEE) meteorologinės stoties. Iš viso nagrinėjamą duomenų aibę sudarė

59541 taškas. Ši duomenų aibė tinkama parodyti disertacijoje pasiūlytus sprendimus, kadangi ją sudaro daugiamačiai taškai, o jų skaičius gana didelis, ir tai leis atskleisti pasiūlytų sprendimų privalumus. Siekiant realios interpretacijos meteorologinių duomenų aibė gali būti suskirstyta į 3 klases pagal meteorologinę stotį arba į 12 klasių, nusakančių mėnesius. Taikant pasiūlytus sprendimus pradinių duomenų aibės dimensija mažinama PCA metodu ($d = 2$). Nors tyrimais ir nustatyta, kad MDS metodu gaunama tiksliausia projekcija ir mažiausia projekcijos paklaida, ieškant meteorologinių duomenų aibės projekcijos, sudarytos iš 59541 taško, nepakanka kompiuterio operatyviosios atminties. Dėl to šis metodas nepasirinktas tolimesnei analizei. Metodai, paremti valdymo taškais (PLMP, LAMP, RBF), nepasirodė akivaizdžiai pranašesni už kitus pagal projekcijos paklaidą ir skaičiavimo laiką (žr. 4.2.1 skyrelį), tad dimensijai mažinti pasirinktas PCA metodas. Šis metodas pasirinktas dėl greitų skaičiavimų su didelės apimties duomenų aibėmis. 5.1 paveiksle pateikiamas daugiamačių taškų išsidėstymas dvimatėje erdvėje pagal meteorologines stotis (geltoni taškai atitinka Jakutsko duomenis, mėlyni taškai – Vilniaus, turkio spalvos – Singapūro), 5.2 paveiksle – pagal mėnesius. 5.1 paveiksle matyti aiškiai atsiskiriantys trys skirtingose klimato juostose esantys miestai.



5.1 pav. Vizualizuota meteorologinių duomenų aibė pagal meteorologines stotis

5.2 paveiksle pastebėtina, kad taškai, atitinkantys pavasario, vasaros, rudens mėnesius Vilniuje, išsidėstę arčiau taškų, atitinkančių Singapūrą, o taškai, atitinkantys šaltuosius žiemos mėnesius Vilniuje, esti arčiau taškų, atitinkančių Jakutską.



5.2 pav. Vizualizuota meteorologinių duomenų aibė pagal mėnesius (skalėje pateikiami metų mėnesiai pagal spalvas)

5.2 Projekcijos paklaidos apskaičiavimas

Šiame poskyryje pateikiamas disertacijos 3.1 poskyryje dviejų pasiūlytų projekcijos paklaidos apskaičiavimo būdų pritaikymas meteorologinių duomenų aibei.

1-asis pasiūlytas būdas – projekcijos paklaidą vertinti ne visai duomenų aibei, o jos imčiai. 2-asis būdas – projekcijos paklaidą skaičiuoti visai duomenų aibei, tačiau duomenų aibę skaičiavimų metu dalyti į dalis.

1-ajam būdui pasirinktos trys imtys, kurių dydžiai yra n''=$10000,\ 5000, 1000$. Imtys parinktos atsitiktinės imties sudarymo būdu.

5.1 lentelėje pateiktos PCA metodu gautų imčių projekcijų paklaidų reikšmės ir skaičiavimo laikas. Visos meteorologinių duomenų aibės projekcijos paklaida ir skaičiavimo laikas pateikiamas paryškintu šriftu. Kadangi meteorologinių duomenų aibę sudaro daugiau kaip 30000 taškų, visos duomenų aibės projekcijos paklaida apskaičiuota 2-uoju pasiūlytu būdu.

Skaičiuojant meteorologinių duomenų aibės projekcijos paklaidą 2-uoju pasiūlytu būdu, duomenų aibė buvo dalijama į mažesnes 10000 objektų duomenų aibes. Imčių projekcijos paklaidos apskaičiuotos taikant MATLAB funkciją *pdist.* 5.1 lentelėje pateikti skaičiavimo rezultatai rodo, kad projekcijos paklaidos reikšmės reikšmingai nesiskiria, o skaičiavimo laikas skiriasi daug kartų.

5.1 lentelė. Projekcijos paklaidos reikšmės ir skaičiavimo laikas analizuojant skirtingos apimties meteorologinių duomenų aibės imtis

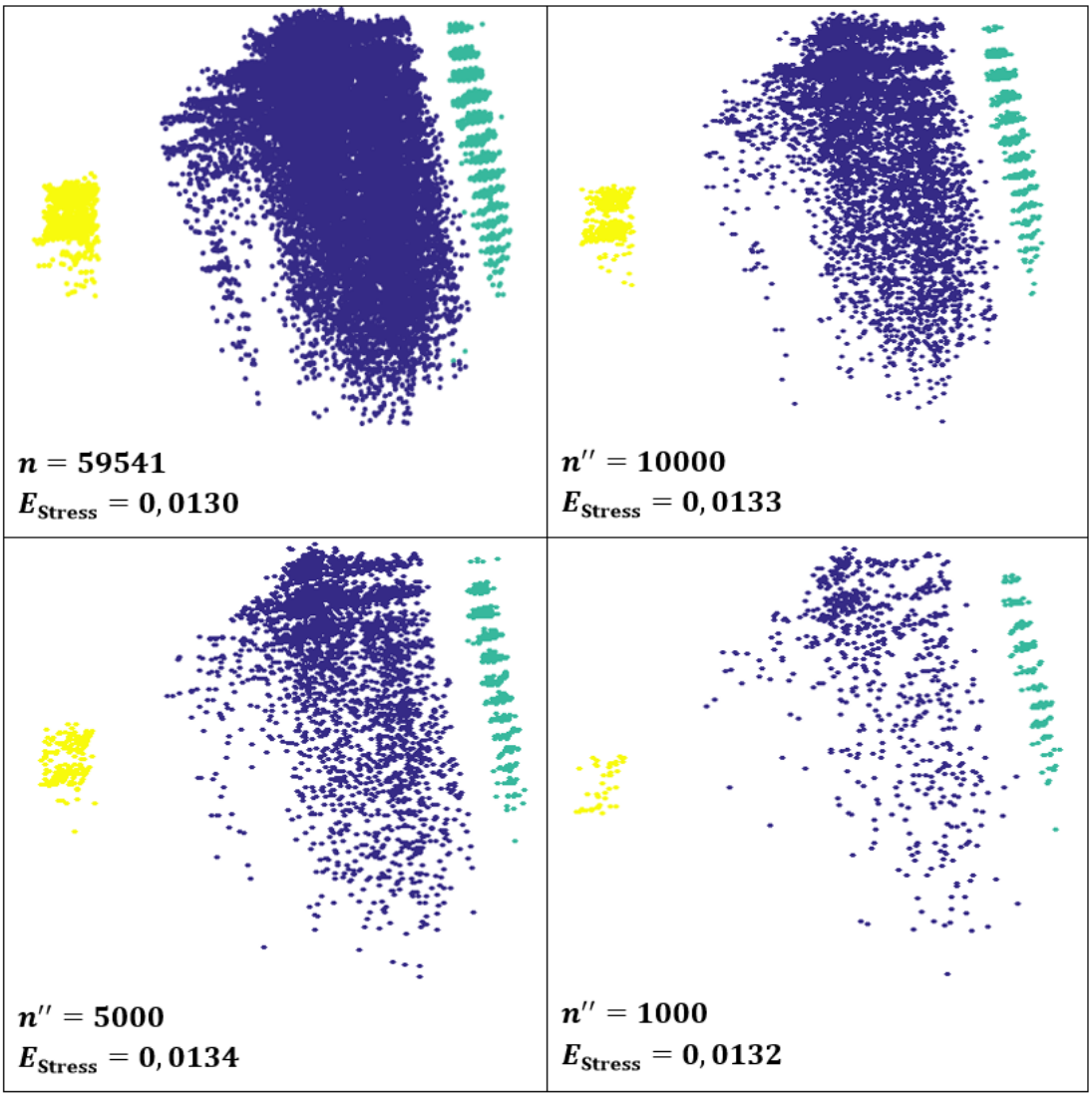
(n'' – imties dydis)

($\boldsymbol{n}$'')	**Paklaida**	**Laikas (s)**
59541	**0,0130**	**61,18**
10000	0,0133	1,32
5000	0,0134	0,47
1000	0,0132	0,02

PCA metodu gautų daugiamačių taškų išsidėstymai dvimatėje erdvėje, kai vizualizuojama meteorologinių duomenų aibė ir jos imtys, pateikiami 5.3 paveiksle. Matyti, kad imčių projekcijų taškų išsidėstymo struktūra panaši į visos meteorologinių duomenų aibės taškų projekcijos struktūrą.

2-uoju būdu projekcijos paklaida visai meteorologinių duomenų aibei apskaičiuojama per 61,18 s. O projekcijos paklaidos skaičiavimas, kai taikomas ciklas *for*, kuriame paklaida skaičiuojama panariui kiekvienam duomenų aibės taškui, užtrunka 289,95 s. Taikyti MATLAB funkciją *pdist* nedalijant orų duomenų aibės į dalis nepakanka kompiuterio operatyviosios atminties.

Skaičiavimo rezultatai patvirtina, kad abu pasiūlyti būdai gali būti efektyviai taikomi projekcijos paklaidai skaičiuoti, kai dirbama su realiais didelės apimties daugiamačiais duomenimis.



5.3 pav. Meteorologinių duomenų aibės ir jos imčių dvimačiai vaizdai. Kiekvieno paveikslėlio kairiajame kampe pateikiamas duomenų aibės arba imties dydis (n“) ir projekcijos paklaida (E_{Stress})

5.3 Duomenų vizualizavimas

Šiame poskyryje pateikiamas disertacijos 3.2 poskyryje pasiūlytos duomenų vizualizavimo strategijos pritaikymas meteorologinių duomenų aibei. Duomenų vizualizavimo strategiją sudaro du etapai.

1-asis etapas – duomenų aibės imties sudarymas *pasiūlytu imties* sudarymo būdu, kuriame:

- daugiamačiai duomenys klasterizuojami *k-medoids* metodu;

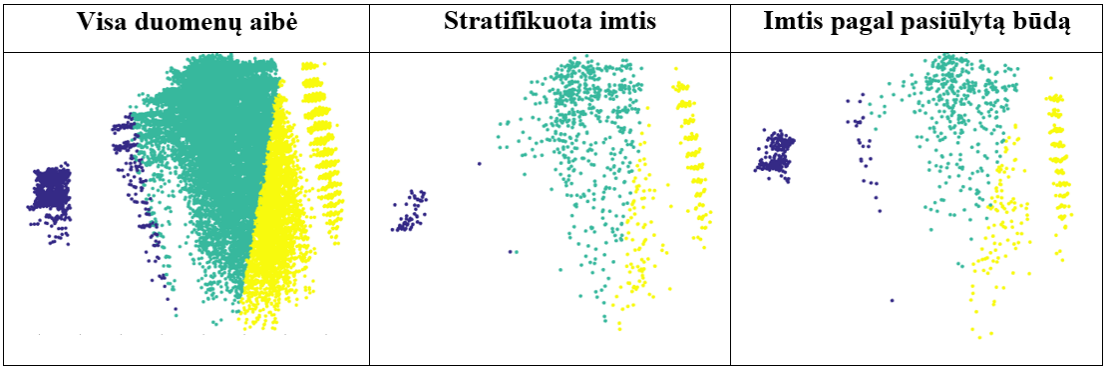
- klasterių skaičius τ nustatomas pagal *Calinski-Harabasz* klasterių nustatymo kriterijų;
- įvertinus duomenų aibės taškų tankumą sudaroma imtis.

Siekiant įvertinti *pasiūlytą imties* sudarymo būdą, pastarasis lyginamas su *stratifikuotu* imties sudarymo būdu. Imties dydis $s = 1000$.

5.2 lentelėje pateikiamas *stratifikuotos* imties ir *pasiūlyto būdo* duomenų aibės imčiai sudaryti atrenkamų taškų skaičius iš kiekvieno klasterio. Meteorologinių duomenų aibė suskirstyta į 3 klasterius. Matyti, kad skirtingais būdais sudarant imtį atrenkamų taškų iš kiekvieno klasterio skaičius skiriasi, ypač iš 2-ojo klasterio. Meteorologinių duomenų aibės ir jos imčių, sudarytų skirtingais būdais, vaizdai pateikiami 5.4 paveiksle. Dalis taškų, atitinkančių sausio ir vasario mėnesius Vilniuje, sudaro vieną klasterį su taškais, atitinkančiais Jakutską. Taškai, atitinkantys birželio–rugpjūčio mėnesius Vilniuje, yra klasterizuojami kartu su taškais, atitinkančiais Singapūrą. Taškai, atitinkantys likusiuosius mėnesius Vilniuje, sudaro atskirą klasterį. Vizualizavimo rezultatai parodė, kad *pasiūlytu būdu* sudarytos imties taškų pasiskirstymas yra panašus į visos meteorologinių duomenų aibės, o *stratifikuota* imtis neatspindi antrojo klasterio struktūros (mėlyni taškai), t. y. tik keletas taškų, atitinkančių Vilnių, parenkami į imtį iš 2-ojo klasterio.

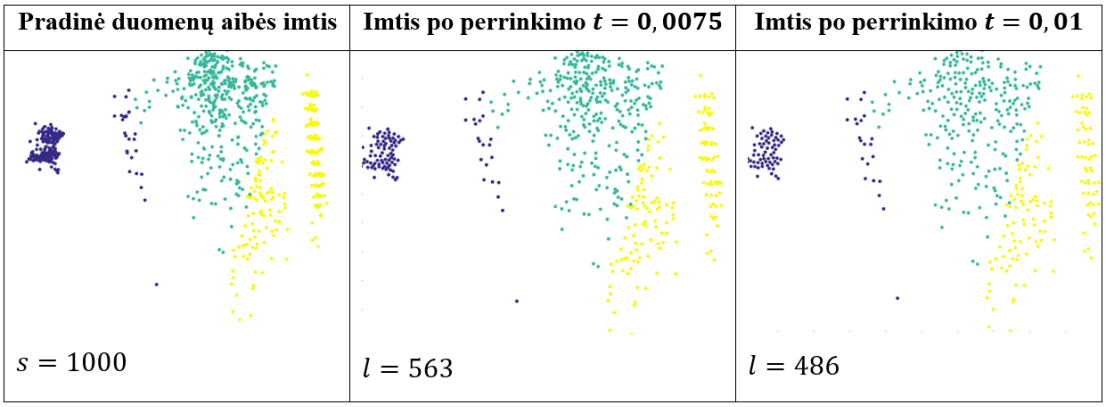
5.2 lentelė. Į imtį atrenkamų taškų skaičius iš kiekvieno klasterio

Klasteris (*i*)	**Visa duomenų aibė**	**Imtis *pasiūlytas būdas***	**Imtis *stratifikuota***
1	34640	397	582
2	3180	261	53
3	21721	342	365



5.4 pav. Vizualizavimo rezultatų lyginimas naudojant skirtingus duomenų aibės imties sudarymo būdus

2-asis etapas – imties vizualizavimas. Šiame etape šalinamas taškų persidengimas. Iš pradžių duomenų aibės imties projekcija, gauta 1-ajame etape, sunormuojama intervale $(0; 1)$. Taškai perrenkami su dviem skirtingomis slenksčio reikšmėmis $t = 0{,}01$, $t = 0{,}0075$. Meteorologinių duomenų aibės imties vaizdai po perrinkimo su tam tikra slenksčio reikšme pateikiami 5.5 paveiksle. Matyti, kad persidengimas šalinamas su slenksčio reikšme $t = 0{,}01$ ir galime identifikuoti kiekvieno taško vietą sklaidos diagramoje.



5.5 pav. Vizualizavimo rezultatai, kai imtis perrenkama su tam tikra slenksčio reikšme analizuojant meteorologinių duomenų aibę. Imties dydis nurodytas apatiniame kairiajame kampe

5.4 Penktojo skyriaus apibendrinimas

Atlikta meteorologinių duomenų analizė parodė, kad skaičiuojant projekcijos paklaidą duomenų aibės imčiai prarandamas projekcijos paklaidos tikslumas nėra reikšmingas. Skaičiuojant projekcijos paklaidą 2-uoju pasiūlytu būdu, kai duomenų aibė dalijama į dalis, pakanka kompiuterio operatyviosios atminties. Meteorologinių duomenų aibės vizualizavimas pagal pasiūlytą duomenų vizualizavimo strategiją patvirtino, kad sudarant imtį pasiūlytu būdu ir vizualizavus jos dvimačius taškus išlaikoma realių duomenų struktūra.

Meteorologinių duomenų aibės analizės rezultatai leidžia teigti, kad disertacijoje pasiūlyti sprendimai taikytini sprendžiant realius duomenų analizės uždavinius.

Bendrosios išvados

Tiriant dimensijų mažinimo metodus darbe gauti šie rezultatai: ištirti įvairūs dimensijos mažinimo metodai, iš jų klasikiniai gerai žinomi metodai ir metodai, kuriuose projekcija randama remiantis valdymo taškais; ištirti įvairūs projekcijos kokybės įvertinimo matai; pasiūlyti ir ištirti projekcijos paklaidos apskaičiavimo būdai didelės apimties duomenų aibėms; pasiūlyta ir ištirta didelės apimties duomenų aibių vizualizavimo strategija, leidžianti neprarasti retų klasterių ir bendros duomenų struktūros ir vizualizuoti duomenų aibės taškus be persidengimo; pademonstruotas disertacijoje pasiūlytų sprendimų pritaikymas sprendžiant realų uždavinį, kai analizuojama meteorologinių duomenų aibė.

Atlikti tyrimai atskleidė darbe pasiūlytų projekcijos paklaidos apskaičiavimo būdų ir duomenų aibių vizualizavimo strategijos naudą tirti didelės apimties duomenų aibėms. Eksperimentinių tyrimų rezultatai leidžia daryti šias išvadas:

1. Projekcijos paklaida gali būti vertinama pagal duomenų aibės imtį analizuojant didelės apimties duomenų aibes. Projekcijos paklaidos duomenų aibės imčiai skaičiavimo laikas yra trumpesnis nei skaičiuojant visai duomenų aibei. Vienoms nagrinėtoms duomenų aibėms laikas sutrumpėja 2–9 kartus, tačiau yra duomenų aibių, kurioms skaičiavimo laikas sutrumpėja šimtais kartų.
2. Projekcijos paklaidos skaičiavimas dalijant pradinę duomenų aibę į dalis leidžia sutrumpinti skaičiavimo laiką beveik 6 kartus lyginant su projekcijos paklaidos skaičiavimo būdu, taikančių ciklą, kuriame paklaida skaičiuojama panariui kiekvienam duomenų aibės taškui. Projekcijos paklaidos skaičiavimui, kai pradinė duomenų aibė dalijama į dalis, pakanka personalinio kompiuterio operatyviosios atminties 15 kartų didesnei duomenų aibei lyginant su projekcijos paklaidos

skaičiavimo būdu, kai naudojama nedalyta duomenų aibė ir speciali funkcija, pritaikyta greitam atstumų skaičiavimui.

3. Duomenų aibės vizualizavimo strategijoje pasiūlytas duomenų aibės imties sudarymo būdas išlaiko duomenų struktūrą įvairioms testinėms duomenų aibėms. Tiriant duomenų vizualizavimą be persidengimo, parodyta, kad daugiamačiams taškams vizualizuoti sklaidos diagramoje pakanka iki 1000 taškų, kad būtų atskleista bendra duomenų struktūra.

Literatūra

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

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

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

[4] P. Stefanovič, Saviorganizuojančių neuroninių tinklų vizualizavimas ir jo kokybės nustatymas. Daktaro disertacija, Vilniaus universitetas, 2015.

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

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

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

[8] N. Galiauskas, Optimizavimo algoritmai daugiamatėms skalėms su miesto kvartalo atstumais ir jų lygiagretinimas. Daktaro disertacija, Vilniaus universitetas, 2015.

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

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

[11] A. Žilinskas and J. Žilinskas, "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.

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

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

[14] International Data Corporation, The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things, 2014. [Online]. Available:https://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm. [Accessed 30 7 2018].

[15] M. Cox and D. Ellsworth, "Application-Controlled Demand Paging for Out-of-Core Visualization," in *VIS '97 Proceedings of the 8th conference on Visualization '97* , Phoenix, 1997.

[16] S. Landset, T. Khoshgoftaar, A. N. Richter and T. Hasanin, "A survey of open source tools for machine learning with big data in the Hadoop ecosystem," *Journal of big data,* vol. 2, no. 24, pp. 1–36, 2015.

[17] D. Laney, "3D data management: controlling data volume, velocity and variety," *Meta group,* 2001.

[18] Y. Demchenko, P. Grosso, C. de Laat and P. Membrey, "Addressing big data issues in scientific data infrastructure," in *International Conference on Collaboration Technologies and Systems (CTS),* San Diego, 2012.

[19] G. Dzemyda, O. Kurasova and J. Žilinskas, Daugiamačių duomenų vizualizavimo metodai, 1 leidimas., Vilnius: Mokslo aidai, 2008.

[20] P. Joia, F. V. Paulovich, D. Coimbra, J. A. Cuminato and L. G. Nonato, "Local affine multidimensional projection," *IEEE Transactions on Visualization and Computer Graphics,* vol. 17, no. 12, pp. 2563–2571, 2011.

[21] F. V. Paulovich, C. T. Silva and L. G. Nonato, "Two-phase mapping for projecting massive data sets," *IEEE Transactions on Visualization and Computer Graphics,* vol. 16, no. 6, pp. 1281–1290, 2010.

[22] I. Borg and P. Groenen, Modern Multidimensional Scaling: Theory and Applications, 2 ed., New York: Springer, 2005.

[23] K. Pearson, "On Lines and Planes of Closest Fit to Systems of Points in Space," *Philosophical Magazine,* vol. 2, no. 11, pp. 559–572, 1901.

[24] L. P. J. van der Maaten, E. O. Postma and H. J. van den Herik, "Dimensionality reduction: a comparative review," 2009.

[25] C. Sorzano, J. Vargas and A. Pascual-Monato, "A survey of dimensionality reduction techniques," 2014.

[26] I. Joliffe, Principle component analysis, 2 ed., New York: Springer, 2002.

[27] I. K. Fodor, "A survey of dimension reduction techniques," 2002.

[28] C. J. C. Burges, "Dimension reduction: a guided tour," *Foundations and trends in machine learning*, vol. 2, no. 4, pp. 275–365, 2010.

[29] A. Hyvarinen, "Independent component analysis: recent advances," *Philosophical Transactions of the Royal Society A*, vol. 371, no. 1984, 2013.

[30] P. Comon, "Independent component analysis – a new concept," *Signal Processing*, vol. 36, no. 3, pp. 287–214, 1994.

[31] A. Hyvarinen, J. Karhunen and E. Oja, Independent Component Analysis, 1 ed., New York: John Wiley & Sons, 2001.

[32] A. Hyvarinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[33] J. Wang and C. I. Chang, "Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis," *IEEE Transactions on geoscience and remote sensing*, vol. 44, no. 6, pp. 1586–1600, 2006.

[34] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: applications to image and text data," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, 2001.

[35] W. B. Johnson and J. Lindenstrauss, "Extensions of Lipshitz mapping into Hilbert space," *Contemporary Mathematics*, vol. 26, pp. 189–206, 1984.

[36] S. Dasgupta and A. Gupta, "An elementary proof of the Johnson-Lindenstrauss lemma. Technical Report TR-99-006," California, 1999.

[37] D. Achlioptas, "Database-friendly random projections," in *Symposium on Principles of Database Systems (PODS)*, 2001.

[38] P. Li, T. J. Hastie and K. W. Church, "Very sparse random projections," in *KDD'06: Proceedings of the 12th ACMSIG KDD international conference on Knowledge discovery and data mining*, New York, 2006.

[39] E. Amorim, E. Brazil, L. Nonato and F. Samavati, "Multidimensional projection with radial basis function and control points selection," in *IEEE Pacific Visualization Symposium*, Yokohama, 2014.

[40] S. Chen, C. Cowan and P. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302–309, 1991.

[41] O. Kurasova and A. Molytė, "Quality of quantization and visualization of vectors obtained by neural gas and self-organizing map," *Informatica,* vol. 22, no. 1, pp. 115–134, 2011.

[42] P. Tan, M. Steinbach and V. Kaumar, Introduction to data mining, Boston: Addison-Wesley, 2005.

[43] A. Gupta and R. Bowden, "Evaluating dimensionality reduction techniques for visual category recognition using Renyi entropy," in *19th European Signal Processing Conference (EUSIPCO 2011)*, Barcelona, 2011.

[44] P. J. Rousseeuw, "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis," *Computational and Applied Mathematics,* vol. 20, pp. 53–65, 1987.

[45] K. Paulauskienė and O. Kurasova, "Dimensijų mažinimo metodų tyrimas įvairių apimčių duomenims analizuoti," įtraukta *Informacinės technologijos. 19-osios konferencijos „Informacinė visuomenė ir universitetinės studijos“ pranešimų medžiaga*, Kaunas, 2014.

[46] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations,* vol. 11, no. 1, pp. 10–18, 2009.

[47] M. Berthold, N. Cebron, F. DILL, T. Gabriel, T. Kotter and T. Meinl, "KNIME: The Konstanz Information Miner," *Data Analysis, Machine Learning and Applications, Studies in Classification, Data Analysis, and Knowledge Organization*, pp. 319–326, 2008.

[48] T. Curk, J. Demšar, Q. Xu, G. Leban, U. Petrovič and I. Bratko, "Microarray data mining with visual programming," *Bioinformatics,* vol. 21, no. 3, pp. 396–398, 2005.

[49] J. Nahar, T. Imam, K. S. Tickle and Y. P. P. Chen, "Computational intelligence for heart disease diagnosis: a medical knowledge driven approach," *Expert systems with applications,* vol. 40, no. 1, pp. 96–104, 2013.

[50] M. P. Mazantez, R. J. Marmon, C. B. Reisser and I. Morao, "Drug discovery applications for KNIME: an open source data mining platform," *Current topics in medicinal chemistry,* vol. 12, no. 18, pp. 1965–1979, 2012.

[51] J. A. Kory and L. Wei, "Data-mining based detection of glaciers: quantifying the extent of Alpine valley glaciation," *Geosciences,* vol. 1, no. 1, pp. 1–18, 2015.

[52] A. H. Wahbeh, Q. Al-Radaideh, M. N. Al-Kabi and E. M. Al-Shawakfa, "A Comparison Study between Data Mining Tools over some Classification Methods," *International Journal of Advanced Computer Science and Applications, Special Issue on Artificial Intelligence,* pp. 18–25, 2011.

[53] B. Zupan and J. Demsar, "Open-Source Tools for Data Mining," *Laboratory and Clinical Medicine,* vol. 28, pp. 37–54, 2008.

[54] B. Madasamy and J. J. Tamiselvi, "Assesement of Freeware Data Mining Tools over Some Wide-Range Characteristics," *Communications in Computer and Information Science,* vol. 292, no. 529–535, 2012.

[55] X. Chen, Y. Ye, G. Williams and X. Xu, "A Survey of Open Source Data Mining Systems," *Lecture Notes in Computer Science,* vol. 4819, pp. 3–14, 2007.

[56] J. Bernataviciene, G. Dzemyda, O. Kurasova and V. Marcinkevicius, "Optimal decisions in combining the SOM with nonlinear projection methods," *European Journal of Operational Research,* vol. 173, no. 3, pp. 729–745, 2006.

[57] J. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on Computers,* vol. C-18, no. 5, pp. 401-409, 1969.

[58] X. Wu, X. Zhu, G. Wu and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering,* vol. 26, no. 1, pp. 97-107, 2014.

[59] N. Choudhary and P. Singh, "Cloud Computing and Big Data Analytics," *International Journal of Engineering Research & Technology,* vol. 2, no. 12, pp. 2700–2704, 2013.

[60] D. O'Leary, "Artificial intelligence and big data," *IEEE Intelligent Systems,* vol. 28, no. 2, pp. 96–99, 2013.

[61] O. Kurasova, V. Marcinkevičius, V. Medvedev and A. Rapečka, "Duomenų tyrybos sistemos, pagrįstos saityno paslaugomis," *Informacijos mokslai,* t. 65, p. 66–74, 2013.

[62] A. G. Shoro and T. R. Soomro, "Big Data Analysis: Apache Spark Perspective," *Global Journal of Computer Science and Technology,* vol. 15, no. 1, pp. 7–14, 2015.

[63] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen, D. Xin, R. Xin, M. J. Franklin, R. Zadeh, M. Zaharia and A. Talwalkar, "MLlib: Machine Learning in Apache Spark," *Journal of Machine Learning Research,* vol. 17, no. 34, pp. 1–7, 2016.

[64] A. Richter, T. Khoshgoftaar, S. Landset and T. Hasanin, "A multi-dimensional comparison of toolkits for machine learning with big data," in *IEEE International conference on Information Reuse and Integration (IRI),* San Francisco, 2015.

[65] L. Yang, L. Kuang, J. Chen, F. Hao and C. Luo, "A holistic approach to distributed dimensionality reduction of big data," *IEEE Transactions on Cloud Computing,* no. 99, 2015.

[66] J. W. Dunn, A. Burgun, M. O. Krebs and B. Rance, "Exploring and visualizing multidimensional data in translational research platforms," *Briefings in Bioinformatics,* pp. 1–13, September 2016.

[67] L. Zhang, A. Stoffel, S. Mittelst, M. Behrisch, T. Schreck, R. Pompl, S. Weber, H. Last and D. Keim, "Visual Analytics for the Big Data Era – A Comparative Review of State-of-the-Art Commercial Systems," in *IEEE Conference on Visual Analytics Science & Technology 2012*, Washington, 2012.

[68] B. Franke, J. F. Plante, R. Roscher, E. A. Lee, C. Smyth, A. Hatefi, F. Chen, E. Gil, A. Schwing, A. Selvitella, M. M. Hoffman, R. Grosse and D. Hendricks, "Statistical Inference, Learning and Models in Big Data," *International Statistical Review,* vol. 84, no. 3, pp. 371–389, December 2016.

[69] D. Keim, M. Hao, U. Dayal, H. Janetzko and P. Bak, "Generalized Scatter Plots," *Information Visualization,* vol. 9, no. 4, pp. 301–311, 2010.

[70] T. N. Dang, L. Wilkinson and A. Anand, "Stacking Graphic Elements to Avoid Over-Plotting," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1044–1052, 2010.

[71] A. Mayorga and M. Gleicher, "Splatterplots: Overcoming Overdraw in Scatter Plots," *IEEE Transactions on Visualization and Computer Graphics* , vol. 19, no. 9, pp. 1526–1538, 2013.

[72] M. C. Hao, U. Dayal, S. R.C., D. Keim ir H. Janetzko, „Variable binned scatter plots,“ *Information vizualization,* vol. 9, no. 3, pp. 194–203, 2010.

[73] J. Li, j. Martens and J. van Wijk, "A Model of Symbol Size Discrimination in Scatterplots," in *Proceedings 28th ACM Conference on Human Factors in Computing Systems*, Atlanta, 2010.

[74] J. Li, J. van Wijk and J. Martens, "A Model of Symbol Lightness Discrimination in Sparse Scatterplots," in *Proceedings 2010 IEEE Pacific Visualization Symposium (PacificVis)*, Taipei, 2010.

[75] J. Li, J. van Wijk and J. Martens, "Evaluation of Symbol Contrast in Scatterplots," in *Proceedings of the 2009 IEEE Pacific Visualization Symposium (PacificVis)*, Kyoto, 2009.

[76] M. Gounder, V. Iyer and A. Mazyad, "A survey on business intelligence tools for university dashboard development," in *2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC)*, Sultanate of Oman, 2016.

[77] A. Shukla and S. Dhir, "Tools for Data Visualization in Business Intelligence: Case Study Using the Tool Qlikview," *Information Systems Design and Intelligent Applications,* vol. 434, pp. 319–326, 2016.

[78] J. Miller, Big Data Visualization, Birmingham: Packt Publishing, 2017.

[79] S. G. Archambault, J. Helouvry, B. Strohl and G. Williams, "Data visualization as a communication tool," *Library Hi Tech News,* vol. 32, no. 2, pp. 1–9, 2015.

[80] R. Agrawal, A. Kadadi, X. Dai and F. Andres, "Challenges and Opportunities with Big Data Visualization," in *7th International Conference on Management of computational and collective intElligence in Digital EcoSystems*, Caraguatatuba, 2015.

[81] P. Pawliczek and W. Dzwinel, "Interactive Data Mining by Using Multidimensional Scaling," *Procedia Computer science,* vol. 18, pp. 40–49, 2013.

[82] J. Han and M. Kamber, Data Mining Concepts and Techniques, 2 ed., San Francisco: Morgan Kaufman Publishers, 2006, p. 743.

[83] S. L. Lohr, Sampling: Design and Analysis, 2nd edition, Boston: Brooks/Cole, 2010.

[84] G. S. Maddala, Introduction to Econometrics 2nd ed., New York: MacMilan, 1992, p. 89.

[85] O. Maimon and L. Rockah, Data Mininig and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers, Kulwer Academic Publishers, 2005.

[86] M. Lichman, "UCI Machine Learning Repository," University of California, Irvine, School of Information and Computer Sciences, 2013. [Online]. Available: http://archive.ics.uci.edu/ml. [Accessed 1 1 2017].

[87] Y. Ye, Q. Wu, J. Z. Huang and L. X. Li, "Stratified sampling for feature subspace selection in random forests for high dimensional data," *Pattern Recognition,* vol. 46, no. 3, pp. 769–787, 2013.

[88] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters,* vol. 31, no. 8, pp. 651–666, 2010.

[89] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, no. 12, pp. 1650–1654, 2002.

Autorės publikacijų sąrašas disertacijos tema

Straipsniai recenzuojamuose periodiniuose mokslo leidiniuose:

A 1. Paulauskienė, Kotryna; Kurasova, Olga. Control Point Selection For Dimensionality Reduction By Radial Basis Function. *Computational Science and Techniques*. ISSN 2029-9966. 2016, t. 4(1), p. 487–499.

A 2. Paulauskienė, Kotryna; Kurasova, Olga. Projection error evaluation for large data sets. *Nonlinear Analysis: Modelling and Control*. ISSN 1392-5113. 2016, t. 21(1), p. 92–102 (Clarivate Analytics Web of Science, Impact Factor 2017: 0,896).

A 3. Paulauskienė, Kotryna; Kurasova, Olga. Duomenų tyrybos sistemų galimybių tyrimas įvairių apimčių duomenims analizuoti. *Informacijos mokslai*. Vilnius: Vilniaus universiteto leidykla. ISSN 1392-0561. 2013, t. 65, p. 85–95.

Straipsniai konferencijų medžiagoje:

B 1. Paulauskienė, Kotryna; Kurasova, Olga. Dimensijų mažinimo metodų tyrimas įvairių apimčių duomenims analizuoti. *Informacinės technologijos: 19-oji tarpuniversitetinė magistrantų ir doktorantų konferencija „Informacinė visuomenė ir universitetinės studijos“ (IVUS 2014): pranešimų medžiaga*. Kaunas: Technologija.ISSN 2029-4832. 2014, p. 114–121.

B 2. Paulauskienė, Kotryna; Kurasova, Olga. Dimensijų mažinimo metodais gautų projekcijų įvertinimas. *Lietuvos matematikos rinkinys: Lietuvos matematikų draugijos darbai*. ISSN 0132-2818. 2014, t. 55, ser. B, p. 137–142.

B 3. Paulauskienė, Kotryna; Kurasova, Olga. A new dimensionality reduction-based visualization approach for massive data. *WSCG 2017 Posters Proceedings*: *Computer Science Research Notes*. ISSN 2464-4617. 2017, p. 19–24.

Santraukos tarptautinių konferencijų santraukų rinkiniuose:

C 1. Paulauskienė, Kotryna; Kurasova, Olga. Massive data visualization via selecting a data subset. 8-th International workshop on Data analysis methods for software systems: Abstracts book, Druskininkai, 1–3 December, 2016, p. 48.

C 2. Paulauskienė, Kotryna; Kurasova, Olga. Improvement of projection error evaluation for massive data sets. 6-th International Conference on Advanced Technology and Sciences: Abstract book, Riga, 12–15 September, 2017, p. 70.

Kotryna Paulauskienė

DIMENSIJŲ MAŽINIMU PAGRĮSTAS DIDELĖS APIMTIES DUOMENŲ VIZUALIZAVIMAS IR PROJEKCIJOS PAKLAIDOS VERTINIMAS

Daktaro disertacija
Fiziniai mokslai
Informatika (09 P)
Redaktorė Jorūnė Rimeisytė