



Kauno technologijos universitetas
Matematikos ir gamtos mokslų fakultetas

Daugiamačių duomenų vizualizavimas mechaninių sistemų diagnostikai

Baigiamasis magistro studijų projektas

Rūta Leonaitė
Projekto autorė

Doc. dr. Mantas Landauskas
Vadovas

Kaunas, 2024



Kauno technologijos universitetas
Matematikos ir gamtos mokslų fakultetas

Daugiamačių duomenų vizualizavimas mechaninių sistemų diagnostikai

Baigiamasis magistro studijų projektas
Taikomoji matematika (6211AX006)

Rūta Leonaitė
Projekto autorė

Doc. dr. Mantas Landauskas
Vadovas

Doc. dr. Tomas Iešmantas
Recenzentas

Kaunas, 2024



Kauno technologijos universitetas

Matematikos ir gamtos mokslų fakultetas

Rūta Leonaitė

Daugiamačių duomenų vizualizavimas mechaninių sistemų diagnostikai

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;
2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąrašė;
3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);
4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Rūta Leonaitė

Patvirtinta elektroniniu būdu

Leonaitė, Rūta. Daugiamačių duomenų vizualizavimas mechaninių sistemų diagnostikai. Magistro studijų baigiamasis projektas / vadovas doc. dr. Mantas Landauskas; Kauno technologijos universitetas, Matematikos ir gamtos mokslų fakultetas.

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

Reikšminiai žodžiai: daugiamačių duomenų vizualizavimas, likusi naudinga darbo trukmė, atraktorius, rekonstrukcija fazinėje plokštumoje.

Kaunas, 2024. 63 p.

Santrauka

Šiame darbe tiriamos mechaninių sistemos diagnostikos galimybės taikant daugiamačių duomenų vizualizavimo metodus. Mechaninės sistemos dėvėsi, kol galiausiai atsiranda gedimų, galinčių sukelti pavojingų padarinių. Dėl šios priežasties svarbu sistemos pokyčius pastebėti kuo anksčiau. Analizuojami daugiamačiai lėktuvų variklių duomenys, kuriuose užfiksuotas variklio gedimas; variklio būseną nusako likusi naudinga darbo trukmė (RUL). Analizei pasitelkiami klasikiniai daugiamačių duomenų vizualizavimo metodai, projektuojantys daugiamačius duomenis į dvimatę arba trimatę erdvę: *RadViz*, *t* paskirstyta kaimyno rekonstrukcija (t-SNE), principinių komponentų analizė. Apžvelgiama vizualizavimo metodų įvairovė ir klasifikacija, taip pat fazinės erdvės rekonstrukcijos metodai. Kuriamas atraktoriaus rekonstrukcija fazinėje plokštumoje paremtas vizualizavimo metodas. Atliekamas laiko vėlinimų vektoriaus optimizavimas (naudojamas genetinis algoritmas) įvairioms dimensijoms bei stebima, kokią galimą įtaką toks rekonstravimas turi vizualizacijoms. Siekiama vizualiai išskirti gerai veikiančio bei gendančio variklio duomenis, skirstant pagal likusią naudingą veikimo laiką. Mechaninės sistemos būsenos kitimas laike analizuojamas dvimates projekcijas, gautas pradinėje ir rekonstruotoje erdvėje, praplečiant trimatėje erdvėje su laiko dimensija. Slenkančio lango principu kuriamos tarpinės vizualizacijos. Tokiu būdu vertinama vizualizacijų evoliucija pradinėje ir rekonstruotoje erdvėje *RadViz*, t-SNE, principinių komponentų analizės metodams, skaičiuojant vidutinę kvadratinę paklaidą laike kintantiems vaizdams. Nustatyta, kad rekonstruotoje fazinėje erdvėje kurtos vizualizacijos jautriau reaguoja į atsirandančius sistemos gedimus, iš jų ryškiausiai gedimo atsiradimą fiksuoja *RadViz* metodas.

Leonaitė, Rūta. Visualization of Multidimensional Data for Diagnostics of Mechanical Systems. Master's Final Degree Project / supervisor assoc. prof. Mantas Landauskas; Faculty of Mathematics and Natural Sciences, Kaunas University of Technology.

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

Keywords: multidimensional data visualization, remaining useful life, attractor, phase space embedding.

Kaunas, 2024. 63 p.

Summary

This paper presents research on mechanical systems diagnostics using multidimensional data visualization methods. Mechanical systems tend to wear out and, eventually, fail. Failures can have catastrophic consequences. Therefore, diagnostics and early detection of upcoming failure is crucial for safe operation. This paper aims to analyse multidimensional data of aircraft engines, which describes run-to-failure experiments; remaining useful life (RUL) is the indicator to evaluate system health. Multidimensional visualization methods (*RadViz*, t-Stochastic neighbor embedding, principal component analysis), projecting high dimensional data to low dimensional space, are explored. Attractor in phase space is reconstructed, embeddings created in different phase space dimensions and time delays are optimized by means of genetic algorithm. Attractor's possible impact on visualizations is explored, aiming to distinguish the time series observations by remaining useful life. Changes in a mechanical system are visualized. Classical two-dimensional projections of both time series data and reconstructed attractor are expanded to three-dimensional by adding a time dimension. Intermediate images are created using a sliding window, evolution of images is analysed as faults accumulate in the system. Images are assessed by computing mean squared differences for the consecutive time-varying images for *RadViz*, t-Stochastic neighbor embedding, principal component analysis projections in both original and reconstructed phase space. Results show that phase space multidimensional visualizations are better at capturing moments of appearing faults, *RadViz* being the most sensitive method to visualize changes in a mechanical system.

Turinys

Lentelių sąrašas	8
Paveikslų sąrašas	9
Santrumpų ir terminų sąrašas	11
Įvadas.....	12
1. Literatūros apžvalga	13
1.1. Daugiamačių duomenų vizualizavimo metodai ir jų klasifikacija	13
1.1.1. Neprojektuojantys duomenų į mažesnių dimensijų erdvę metodai.....	15
1.1.2. Daugiamačių projekcijų metodai.....	16
1.1.3. Spindulinė vizualizacija.....	16
1.1.4. Principinių komponentų analizė	18
1.1.5. t paskirstyta stochastinė kaimyno rekonstrukcija	18
1.1.6. Klasterizavimo metodai daugiamačiams duomenims	19
1.2. Atraktorius fazinėje plokštumoje	20
1.2.1. Dimensijų ir vėlinimo vektoriaus parinkimo metodai.....	20
1.2.2. Atraktoriaus kūrimas daugiamačių laiko eilučių atveju	22
1.3. Vizualizacijų kokybės vertinimas.....	23
1.3.1. Vizualizacijų iškreipimo priežastys.....	23
1.3.2. Vizualizacijų tikslumo vertinimo metrikos	24
1.4. Mechaninių sistemų diagnostika	24
1.4.1. Vizualizavimo metodų taikymas mechaninių sistemų gedimų diagnostikoje.....	26
1.5. Baigiamojo projekto temos ir uždavinių pagrindimas.....	27
2. Duomenys ir tyrimo metodai.....	28
2.1. Duomenys.....	28
2.2. Duomenų apdorojimas: filtravimas ir normavimas	29
2.3. Atraktoriaus rekonstrukcija fazinėje plokštumoje.....	30
2.3.1. Rekonstrukcija fazinėje erdvėje daugiamačiu atveju	31
2.3.2. Rekonstruotos erdvės dimensijos parinkimas	31
2.3.3. Vėlinimų vektoriaus optimizavimas genetiniu algoritmu	32
2.4. Daugiamačių duomenų vizualizacijos metodai	33
2.4.1. Principinių komponentų analizė	33
2.4.2. Spindulinė vizualizacija <i>RadViz</i>	34
2.4.3. t-SNE vizualizavimo metodas	35
2.5. Skaitinis vizualizacijų tikslumo vertinimas.....	35
2.5.1. Vidutinė kvadratinė paklaida.....	35
2.5.2. Vaizdų evoliucijos laike stebėjimas	36
2.6. Metodinė schema.....	36
2.7. Programinė įranga	37
3. Tyrimų rezultatai ir jų aptarimas.....	38
3.1. Pradinių duomenų apdorojimas ir vizualizacijos	38
3.1.1. Tiriamoji duomenų analizė.....	38
3.1.2. Triukšmo filtravimas	40
3.1.3. Pradinių duomenų vizualizacijos dimensijų mažinimo metodais	41
3.2. Atraktoriaus rekonstrukcija fazinėje erdvėje ir jos vizualizavimas.....	42
3.2.1. Vėlinimų vektoriaus optimizavimo genetiniu algoritmu rezultatai.....	43

3.2.2. Normavimo įtakos atraktoriaus vizualizacijoms palyginimas.....	45
3.2.3. Kintamųjų koreliacijos su RUL įtaka atraktoriaus vizualizacijoms	47
3.2.4. Dimensijų tyrimas ir parinkimas	49
3.3. Principinių komponentų analizė su laiko dimensija pradiniam duomenims.....	52
3.4. Vizualizacijų pokyčio laike skaitinis įvertinimas.....	52
Išvados	58
Literatūros sąrašas	59

Lentelių sąrašas

1 lentelė. Dimensijų mažinimo metodai duomenų vizualizavimui.....	14
2 lentelė. Duomenų rinkinio kintamieji ir jų reikšmės.....	28
3 lentelė. Genetinio algoritmo optimizavimo parametrai ir jų reikšmės; čia n – vėlinimų vektorių ilgis, įgyja reikšmes: 2, 15, 32.....	43
4 lentelė. Genetinio algoritmo optimizavimo rezultatai, naudojant standartiniu būdu normuotus duomenis.....	44
5 lentelė. Genetinio algoritmo optimizavimo rezultatai, naudojant nestandartiniu <i>MinMaxRUL</i> būdu normuotus duomenis	44
6 lentelė. Vaizdų evoliucija duomenims pradinėje erdvėje <i>RadViz</i> , t-SNE ir PCA metodais, kintant laikui; čia i – laiko indeksas, kurio metu fiksuotas vaizdas (0 – eksperimento pradžia, 96 – eksperimento pabaiga).....	54
7 lentelė. Vaizdų evoliucija fazinėje erdvėje rekonstruotam atraktoriui, vizualizuotam <i>RadViz</i> , t-SNE ir PCA metodais, kintant laikui; čia i – laiko indeksas, kurio metu fiksuotas vaizdas (0 – eksperimento pradžia, 1412 – eksperimento pabaiga)	54

Paveikslų sąrašas

1 pav. Daugiareikšmių lygiagrečiųjų koordinatinių grafikas [12].....	15
2 pav. <i>RadViz</i> (a) ir Žvaigždžių koordinatinių (b) metodų palyginimas retų duomenų vizualizavimui [20]	17
3 pav. (a) t-SNE, (b) <i>Sammon</i> metodų vizualizacijos MNIST duomenų rinkiniui [23]	19
4 pav. (a) <i>Isomap</i> , (b) LLE metodų vizualizacijos MNIST duomenų rinkiniui [23].....	19
5 pav. Optimalių rekonstrukcijos parametrų paieška su Monte Karlo sprendimų medžio algoritmu, čia D_i – rekonstravimo ciklas, s_i – laiko eilutės [36]	23
6 pav. Gedimo vizualizavimas lygiagrečiosiomis koordinatėmis: mėlyna spalva vaizduojama normaliai veikianti sistema, raudona – gedimas [43].....	26
7 pav. Variklių veikimo laiko (RUL) kvartilų grafikas	29
8 pav. Variklių veikimo laiko (RUL) pasiskirstymo diagrama.....	29
9 pav. Rekonstrukcijos fazinėje plokštumoje schema 3 dimensijų atveju.....	31
10 pav. Tyrimo metodinė schema	37
11 pav. Sklaidos grafikai priklausomiems kintamiesiems ir RUL.....	38
12 pav. Koreliacijų matricos grafikas pradiniam duomenims.....	39
13 pav. Pradinių duomenų vizualizacijos skirtingiems varikliams, <i>NRC</i> ir <i>Ps30</i> kintamiesiems	39
14 pav. Triukšmo filtravimas skirtingiems slenkančio lango dydžiams su medianos ir slenkančio vidurkio filtrais, <i>osI</i> kintamajam.....	40
15 pav. PCA metodas su dviem komponentėmis filtruotiems duomenims su 3, 5 ir 7 slenkančio lango dydžiais su slenkančio vidurkio (MA) ir medianos (Med.) filtrais	41
16 pav. Pradinių duomenų vizualizacijos visiems 100 variklių, visų kategorijų („Aukšta“, „Vidutinė“, „Žema“) RUL, metodai: (a) – <i>RadViz</i> , (b) – t-SNE, (c) – PCA	42
17 pav. Pradinių duomenų vizualizacijos 15 variklio duomenims, „Aukštos“ ir „Žemos“ RUL kategorijoms, metodai: (a) – <i>RadViz</i> , (b) – t-SNE, (c) – PCA	42
18 pav. Tikslų funkcijos kitimo grafikas optimizavimo proceso metu, naudojant standartiškai normuotus duomenis: (a) – $n = 2$, (b) – $n = 15$, (c) – $n = 32$, čia: x ašis – „ <i>Generation</i> “ – žymi iteracijų skaičių (skirtingas populiacijos kartas), y ašis – „ <i>Fitness</i> “ – žymi tikslo funkcijos reikšmę	44
19 pav. Tikslų funkcijos kitimo grafikas optimizavimo proceso metu, naudojant nestandartinio <i>MinMax</i> normavimo duomenis: (a) – $n = 2$, (b) – $n = 15$, (c) – $n = 32$, čia: x ašis – „ <i>Generation</i> “ – žymi iteracijų skaičių (skirtingas populiacijos kartas), y ašis – „ <i>Fitness</i> “ – žymi tikslo funkcijos reikšmę	45
20 pav. Atraktoriaus vizualizacijos skirtingam dimensijų (paveiksle: dim.) skaičiui, 51 varikliui, naudojant standartinį <i>StandardScaler</i> ir nestandartinį <i>MinMaxRUL</i> normavimo metodus; vėlinimų vektoriai pagal dimensijas iš 4 lentelės bei 5 lentelės	46
21 pav. <i>RadViz</i> , t-SNE, PCA vizualizacijos naudojant standartinį <i>StandardScaler</i> normavimo metodą; kuriamas 8 dimensijų atraktorius 51 varikliui ($n = 15$, optimalus vėlinimų vektorius [6, 9, 6, 6, 11, 19, 17])	46
22 pav. <i>RadViz</i> , t-SNE, PCA vizualizacijos naudojant nestandartinį <i>MinMaxRUL</i> normavimo metodą; kuriamas 8 dimensijų atraktorius 51 varikliui ($n = 15$, optimalus vėlinimų vektorius [7, 4, 1, 6, 15, 19, 17])	47
23 pav. Atraktoriaus (3 dimensijų) tik su vienu kintamuoju (<i>Ps30</i>) vizualizacijos <i>RadViz</i> , t-SNE, PCA metodais; 96 variklis.....	47
24 pav. Atraktoriaus (3 dimensijų) tik su vienu kintamuoju (<i>Phi</i>) vizualizacijos <i>RadViz</i> , t-SNE, PCA metodais; 96 variklis.....	48

25 pav. Atraktoriaus (3 dimensijų) tik su dviem kintamaisiais (<i>Ps30</i> ir <i>Phi</i>) vizualizacijos <i>RadViz</i> , t-SNE, PCA metodais; 96 variklis	48
26 pav. Atraktoriaus (8 dimensijų), kurto tik su teigiamai koreliuojančiais su RUL kintamaisiais (<i>Phi</i> , <i>P30</i> , <i>W31_HPT</i> , <i>W32_LPT</i>) vizualizacijos <i>RadViz</i> , t-SNE, PCA metodais; 51 variklis	49
27 pav. Atraktoriaus vizualizacijos 34 varikliui, 3 dimensijų (paveiksle: dim.) atveju – tiesiogiai, 8 ir 17 – trijų komponentų PCA	49
28 pav. Atraktoriaus dimensijų (paveiksle: dim.) skaičiaus tyrimas (34 variklis, <i>MinMaxRUL</i>).....	50
29 pav. Atraktoriaus vizualizacijų palyginimas skirtingiems varikliams (17 dimensijų, <i>MinMaxRUL</i> , vėlinimų vektorius: [9, 2, 18, 9, 3, 2, 3, 3, 1, 11, 5, 16, 8, 2, 4, 2]).....	51
30 pav. Dvimatės RUL kategorijų vizualizacijos su PCA metodu, 1 (a), 34 (b) ir 71 (c) variklių duomenims	52
31 pav. Trimatės RUL kategorijų vizualizacijos, 1 (a), 34 (b) ir 71 (c) variklių duomenims, naudojant dviejų komponentų PCA ir laiko dimensiją bei <i>StandardScaler</i> normavimą.....	52
32 pav. Pradinių viso signalo duomenų („Aukštos“, „Vidutinės“ ir „Žemos“ RUL), <i>RadViz</i> , t-SNE, PCA grafikai; 34 variklis.....	53
33 pav. Atraktoriaus (17 dimensijų), kurto iš viso signalo duomenų („Aukštos“, „Vidutinės“ ir „Žemos“ RUL), <i>RadViz</i> , t-SNE, PCA grafikai; 34 variklis.....	53
34 pav. Pradinėje erdvėje (a) ir 17 dimensijų atraktoriaus (b) trimačiai grafikai, vizualizuojantys dviejų komponentų PCA ir laiko dimensiją; 34 variklis	53
35 pav. Pradinių duomenų ir rekonstruoto 17 dimensijų atraktoriaus <i>RadViz</i> vizualizacijų MSE kitimo palyginimas; (a) – originali ir filtruota MSE pradinių vaizdų vizualizacijoms, (b) – originali ir filtruota MSE atraktoriaus vizualizacijoms, (c) – pradinės erdvės ir atraktoriaus vizualizacijų MSE palyginimas.....	55
36 pav. Pradinių duomenų ir rekonstruoto 17 dimensijų atraktoriaus t-SNE vizualizacijų MSE kitimo palyginimas; (a) – originali ir filtruota MSE pradinių vaizdų vizualizacijoms, (b) – originali ir filtruota MSE atraktoriaus vizualizacijoms, (c) – pradinės erdvės ir atraktoriaus vizualizacijų MSE palyginimas.....	56
37 pav. Pradinių duomenų ir rekonstruoto 17 dimensijų atraktoriaus PCA vizualizacijų MSE kitimo palyginimas; (a) – originali ir filtruota MSE pradinių vaizdų vizualizacijoms, (b) – originali ir filtruota MSE atraktoriaus vizualizacijoms, (c) – pradinės erdvės ir atraktoriaus vizualizacijų MSE palyginimas.....	56
38 pav. Pradinių duomenų (a) ir rekonstruoto 17 dimensijų atraktoriaus (b) MSE pokytis visiems metodams kartu	57

Santrumpų ir terminų sąrašas

Santrumpos:

PCA – principinių komponentų analizė (angl. *principal component analysis*)

t-SNE – *t* paskirstytas stochastinis kaimyno rekonstravimas (angl. *t-distributed stochastic neighbor embedding*)

RUL – likusi naudinga darbo trukmė (angl. *remaining useful life*)

MNIST – modifikuota Nacionalinio standartų ir technologijų instituto duomenų bazė (angl. *Modified National Institute of Standards and Technology database*).

NASA – Nacionalinė aeronautikos ir kosmoso administracija (angl. *National Aeronautics and Space Administration*)

MSE – vidutinė kvadratinė paklaida (angl. *mean squared error*)

Terminai:

RadViz – spindulinės vizualizacijos metodas, daugiamačius duomenis projektuojantis į dvimatę erdvę, taškus išdėstant apskritime.

StandardScaler – standartinis duomenų normavimas, atliekamas atimant vidurkį ir dalinant iš standartinio nuokrypio.

MinMaxRUL – tyrime pasiūlytas rankiniu būdu suprogramuotas normavimo metodas, kurio metu didžiausių ir mažiausių reikšmių parinkimui naudojami tik aukštą likusią naudingą darbo trukmę turintys stebėjimai.

Įvadas

Mechaninės sistemos aptinkamos daugelyje kasdienio gyvenimo sričių – transporto priemonėse (vidaus degimo varikliai), gamybos pramonėje, buitiniuose prietaisuose. Mechaninėms sistemoms būdingas dėvėjimasis, po tam tikro veikimo laiko gali atsirasti gedimų. Šių sistemų diagnostika ir ankstyvas gedimų aptikimas yra svarbus saugiam eksploatavimui, nes kai kurie gedimai gali turėti katastrofiškų padarinių. Pavyzdžiui, sklandus orlaivių variklių veikimas yra būtinas oro transporto patikimumui užtikrinti, skrydžio metu atsiradęs gedimas keltų grėsmę žmonių saugumui. Ekonominiu atžvilgiu, anksčiau suradus atsirandantį gedimą, jį galima pašalinti su mažesnėmis išlaidomis.

Tyrinėjant mechanines sistemas, dažnai siekiama įvertinti likusią naudingą darbo trukmę (RUL). Šis požymis nusako, kiek laiko sistema veiks, kol suges. Šiame darbe bus tyrinėjamas daugiamačių duomenų vizualizacijų metodų taikymas mechaninių sistemų diagnostikai. Dirbama su orlaivių variklių duomenimis, siekiant sukurti vizualizacijas, kur būtų aiškiai išskiriami variklio veikime atsiradę pakitimai, likusios naudingos darbo trukmės sumažėjimas. Vizualizavimui išbandomi klasikiniai daugiamačių projekcijų metodai, taip pat sudaromos fazinės erdvės rekonstravimu paremtos vizualizacijos.

Tikslas:

Vizualizuoti daugiamačius orlaivių variklių duomenis taip, kad būtų galima įžvelgti mechaninės sistemos likusios naudingos darbo trukmės sumažėjimą.

Uždaviniai:

1. Apžvelgti literatūrą ir identifikuoti, kokie yra daugiamačių duomenų vizualizacijos metodai. Surasti taikymo mechaninių sistemų diagnostikai ir likusios naudingos jų veikimo trukmės vertinimo atvejus.
2. Ištirti daugiamačių duomenų klasikinius vizualizacijos metodus duomenims iš viešai prieinamos duomenų bazės.
3. Sudaryti fazinės erdvės rekonstravimu paremtą vizualizacijos metodą. Optimizuoti metodą, parenkant optimalius laiko vėlinimus ir rekonstruojamo atraktoriaus dimensiją.
4. Nustatyti empirines implikacijas apie tirtų klasikinio ir darbe sudarytų metodų tinkamumą vaizduoti daugiamačius diagnostinius duomenis. Padaryti apibendrinančias tyrimo išvadas.

1. Literatūros apžvalga

Šioje darbo dalyje apžvelgiami daugiamačių duomenų vizualizavimo metodai, jų klasifikacija. Aprašomas atraktoriaus fazinėje plokštumoje kūrimas, siekiant išvelgti laiko eilučių duomenų sąryšius, taip pat šio metodo taikymas daugiamačių laiko eilučių atvejais. Tolimesniuose skyriuose apžvelgiami mechaninių sistemų diagnostikoje aptinkami metodai, daugiamačio vizualizavimo taikymo atvejai šioje srityje.

1.1. Daugiamačių duomenų vizualizavimo metodai ir jų klasifikacija

Duomenų vizualizavimas gali padėti geriau suprasti tiriamus duomenis, pamatyti dėsniumus, savybes bei jų vidinę struktūrą. Dažnai tyrimuose naudojami duomenys būna daugiamačiai – tokiu atveju susiduriama su jų vizualizavimo problema. Daugiamačių duomenų projektavimui į dvimatę ar trimatę erdvę gali būti naudojami įvairūs metodai, kurie dažniausiai kartu atlieka ir dimensijų mažinimo funkciją, o nuo jų tinkamo parinkimo priklauso išlaikytos informacijos kiekis, vidinių duomenų struktūrų perteikimas [1, 2].

Pagrindiniai sunkumai, su kuriais susiduriama dirbant su daugiamačiais duomenimis: tinkamų projekcijų radimas, prasmingas dimensijų parinkimas, triukšmo pašalinimas. Dažnai šie sunkumai apibendrinant įvardijami kaip „dimensijų prakeikimas“ [3]. Kadangi vizualizavimo erdvė yra ribota (dvi arba trys dimensijos), nėra efektyvu stengtis atvaizduoti kiekvieną dimensiją – svarbu atrinkti tik informatyvias, pašalinant mažai reikšmingas, triukšmingas dimensijas [4]. Yra pasiūlymų daugiamačių duomenų vizualizacijos uždavinius skirstyti į kelias esmines kategorijas: pasikartojimų / tendencijų (angl. *pattern*) aptikimas, sąsajų paieška, elgesio / charakteristikų palyginimas, narystės išskyrimas klasteriuose – kadangi triukšmas trukdo pamatyti dėsniumus, metodo tikslas turėtų būti išskirti kuo aiškesnes duomenų savybes [1].

Dimensijų mažinimo metodai, kuriantys projekcijas iš daugiamatės į mažiau dimensijų turinčią erdvę, skirstomi į tiesinius ir netiesinius (žr. 1 lentelę). Taikant tiesinius metodus, daroma prielaida, kad duomenys yra išsidėstę daugiamatės erdvės tiesiniame poerdvyje (arba netoli jo); netiesiniai metodai šia prielaida nesiremia, todėl geba identifikuoti sudėtingesnes duomenų rekonstrukcijas daugiamatėje erdvėje [5]. Tiesinių dimensijų mažinimo metodų rezultatas yra pradinių kintamųjų tiesinės kombinacijos. PCA metodas gražina didžiausios dispersijos kryptis, nepriklausomų komponentų analizė (ICA) – maksimalias statistiškai nepriklausomas kryptis, tiesinė diskriminantinė analizė – didžiausias tarpklasines ir mažiausias klasės viduje esančias dispersijas, kanoninė koreliacijų analizė (CCA) – maksimalias koreliacijų tarp požymių kryptis [6].

Netiesiniai metodai dar gali būti skirstomi į tipus: 1) metodai, siekiantys projekcijose išlaikyti globalią originalių duomenų struktūrą, 2) metodai, siekiantys išlaikyti lokalią pradinių duomenų struktūrą, 3) metodai, kurie atlieka globalų tiesinių modelių rinkinio derinimą. Metodai, išlaikantys lokalią struktūrą, remiasi prielaida, jog tiksliai atvaizdavus atstumus iki artimų taškų, globali duomenų struktūra bus išlaikyta savaime; globalūs metodai dažniausiai stengiasi išlaikyti porinius atstumus tarp taškų [5]. Trečioji grupė metodų suranda tam tikrą skaičių lokaliai tiesinių modelių ir atlieka jų globalų derinimą. Netiesiniai dimensijų mažinimo metodai duomenų rinkiniui suranda atvaizdavimo funkciją arba koordinates, rezultatai vertinami pagal atstumus gautame duomenų rinkinyje. Skirtingi metodai išryškina skirtingo tipo atstumus: klasikinis daugiamačių skalių metodas išlaiko porinius atstumus, *Sammon* vaizdavimas – mažus atstumus, *Isomap* – atstumus tarp klasterių, save organizuojantys žemėlapiai (SOM) atvaizdavimą kuria reguliariame tinkelyje [6].

1 lentelė. Dimensijų mažinimo metodai duomenų vizualizavimui

Tipas	Potipis	Metodas
Tiesiniai	-	Principinių komponentų analizė (PCA) [7] Faktorinė analizė [5] Nepriklausomų komponentų analizė (ICA) [6] Tiesinė diskriminantinė analizė (LDA) [6] Kanoninė koreliacijų analizė (CCA) [6]
	Spindulinė projekcija	Žvaigždžių koordinatės [20] <i>RadViz</i> [20]
Netiesiniai	Globalūs	Daugiamačių skalių metodai (MDS) [5] <i>Isomap</i> [5] Branduolinis PCA [5] Difuzijos žemėlapis [5] Daugiasluksniai autoenkoderiai [5]
	Lokalūs	<i>Laplaciano</i> tikrinių reikšmių vaizdavimai (angl. <i>Laplacian eigenmaps</i>) [5] Lokali tiesinė rekonstrukcija (LLE) [5] <i>Hessian</i> LLE [5] Stochastinė kaimyno rekonstrukcija (SNE), t-SNE [23]
	Globalus tiesinių modelių derinimas	Lokaliai tiesinė koordinacija (LLC) [5]
Neprojektuoja į mažai dimensijų turinčią erdvę	-	Lygiagrečiosios koordinatės [9] <i>Andrews</i> kreivės [9] Taškinių grafikų matrica [9]

Lyginant tiesinius ir netiesinius metodus, ryškus tiesinių metodų privalumas – efektyvumas skaičiavimo požiūriu, nes, apskaičiavus tiesinę transformaciją, kiekvieno taško vaizdavimas Dekarto koordinatėse atliekamas dauginant matricą ir vektorių. Tačiau tiesiniai metodai nesugeba pavaizduoti sudėtingų struktūrų, todėl gali atsirasti vaizdo iškraipymai, ypač dirbant su sudėtingomis daugialypėmis struktūromis. Nors buvo pasiūlytos alternatyvos, kuriomis problemą bandyta spręsti, vis dėlto tiesiniai metodai šiuo aspektu dažniausiai veikia prasčiau už netiesinius metodus [8].

Lokalūs ir globalūs metodai taip pat pasižymi skirtingomis savybėmis. Lokalumo sąvoka vartojama dviem skirtingoms daugiamačių projekcijų metodų savybėms apibūdinti: lokaliai modeliavimui ir lokaliai atvaizdavimui. Pirmoji savybė apibūdina metodus, kuriais pirmiausia siekiama iš duomenų išgauti lokalią geometrinę arba topologinę informaciją, o paskui apskaičiuoti globalų vaizdavimą, išsaugant lokalią informaciją. Tokie metodai yra lokalus tiesinis rekonstravimas (LLE), stochastinis kaimyno rekonstravimas (SNE). Antroji savybė, lokalus atvaizdavimas, taip pat remiasi informacija apie kaimyninius taškus, tačiau nesukuria vieno globalaus vaizdavimo. Vietoj to sukuriama lokalių vaizdavimų rinkinys. Kiekvienas lokalus vaizdavimas projektuoja atitinkamą duomenų poaibį, išsaugodamas lokalią struktūrą. Tokių metodų pavyzdžiai: iš dalies *Laplacianu* paremta projekcija (angl. *Piecewise Laplacian-based projection* – PLP), lokali afininė daugiamatė projekcija (angl. *Local affine multidimensional projection* – LAMP) [8].

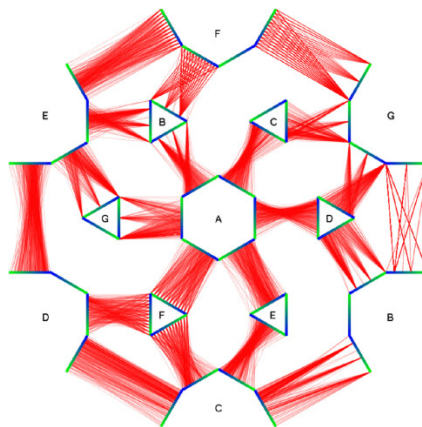
1.1.1. Neprojektuojantys duomenų į mažesnių dimensijų erdvę metodai

Dauguma daugiamačio vizualizavimo metodų veikia kurdami duomenų projekcijas iš daugiamatės erdvės į dvimatę / trimatę erdvę, vis dėlto yra ir kitokiu principu veikiančių algoritmų. Taškinių (sklaidos) grafikų matrica – vienas paprasčiausių daugiamačių duomenų vizualizavimo metodų, kai kintamieji poromis brėžiami taškiniuose grafikuose. Tačiau turint daug kintamųjų, grafikų skaičius tampa didelis, todėl tampa svarbu atrinkti tik informatyvius taškinius grafikus. Yra pasiūlyti metodai, automatiškai atrenkantys reikšmingus taškinius grafikus, naudojant keletą metrių (koreliacija, atskiriamumas, grupavimas) [4].

Lygiagrečiųjų koordinatinių metodas leidžia lygiagrečiomis vertikaliomis linijomis pavaizduoti skirtingas dimensijas (požymius), o kiekvienas individualus stebėjimas vaizduojamas linija, jungiančia taškus esančius ant lygiagrečių koordinatinių linijų. Tačiau esant didesniai dimensijų kiekiui, iš grafiko pamatyti informaciją darosi sudėtingiau. Panašus į jį yra *Andrews* kreivių metodas, leidžiantis daugiamačią tašką vaizduoti kaip tam tikros funkcijos išraišką. Šis metodas gana gerai vizualizuoja dimensijų sąryšius, savybes [9, 10].

Taip pat gali būti naudojamas lygiagrečiųjų koordinatinių plėtinys trimatėje erdvėje; remiamasi prielaida, jog tuomet vienu metu galima perteikti daugiau ryšių, todėl vizualizacija parodo daugiau informacijos apie duomenų struktūrą. Tačiau buvo nustatyta, kad siekiant nustatyti ryšius tarp dviejų kintamųjų, dvimatės koordinatės pasiekia geresnių rezultatų [11].

Buvo pasiūlytas daugiareikšmių lygiagrečiųjų koordinatinių grafikų, kuris skirtas padėti vizualiai tirti ryšius tarp kintamųjų: jame kintamieji išdėstomi ratu, kai kurie yra dubliuojami (išdėstant juos priešingose grafiko pusėse), siekiant atvaizduoti visus kintamųjų ryšius. Parenkamas vienas centrinis kintamasis, kuris nedubliuojamas [12]. Tokios vizualizacijos pavyzdys matomas 1 pav.



1 pav. Daugiareikšmių lygiagrečiųjų koordinatinių grafikų [12]

Dar viena lygiagrečiųjų koordinatinių variacija – tekstilės grafikas, kur ašių išsidėstymas bei mastelis parenkami taip, kad linijos, iš kurių kiekviena žymi vieną stebėjimą, būtų išdėstytos kuo horizontaliau. Grafike susidaro „mazgai“, padedantys interpretuoti duomenis. Šis metodas remiasi tekstilės audinio gamyba: audinys laikomas kokybišku, jei siūlai jame yra išdėstę horizontaliai. Kadangi naudojamas horizontalumo kriterijus, tekstilės grafikas ne tik palengvina galimų ryšių tarp gretimų ašių (t. y. kintamųjų arba požymių) supratimą, bet ir leidžia nustatyti tiesinius ryšius arba ortogonalumą, kurie gali egzistuoti tarp duomenų vektorių [13].

1.1.2. Daugiamačių projekcijų metodai

Daugiamačių projekcijų metodai sudaro grupę metodų, skirtų daugiamatėje erdvėje apibrėžtų taškų projektavimui į mažai dimensijų turinčią vizualią erdvę; taškai projektuojami siekiant išlaikyti porinius atstumus tarp jų [14]. Vizualizacijos kontekste, projektuojama į dvimatę arba trimatę erdvę. Daugiamačių skalių metodai (vizualizavimo kontekste dažniau vadinami daugiamačių projekcijų metodais) panašius taškus vaizduoja arti vienas kito, o nepanašius – toli. Panašumo bei nepanašumo (skirtumo) sąvokos neturi standartinio apibrėžimo. Intuityvus panašumo apibrėžimas: dydis, vertinantis dviejų objektų panašumą – kuo didesnė jo reikšmė, tuo vienodesni yra objektai; kuo mažesnė reikšmė – tuo mažiau panašūs, turintys daugiau skirtumų objektai [8].

Klasikinis daugiamačių skalių metodas siekia, kad mažų dimensijų erdvėje suprojektuoti atstumai kuo tiksliau atitiktų originalius. Metrinis daugiamačių skalių metodas apibendrina klasikinį metodą. Atliekamas klaidos funkcijos minimizavimas: ieškomas sprendinys, kai skirtumai (angl. *dissimilarities*) yra proporcingi tarp projekcijos ir pradinių taškų [15]. Vienas pavyzdžių – *Sammon* vaizdavimas, surandantis duomenų projekciją išlaikant geometrinius sąryšius tarp duomenų vektorių [16]. Esant sudėtingesnei (kreivinei, ritinio formos) duomenų struktūrai, Euklido atstumais paremtas daugiamačių skalių metodas veikia prasčiau [5], todėl buvo pristatytas *Isomap* – tam tikra klasikinio skalių metodo variacija, originalius atstumus pakeičianti geodeziniais [14]. Nemetrinis daugiamačių skalių metodas ieško monotoninių sąryšių skirtumų tarp duomenų taškų daugiamatėje ir mažų dimensijų erdvėje, atliekamas ranginis normavimas vietoj metrinio [15].

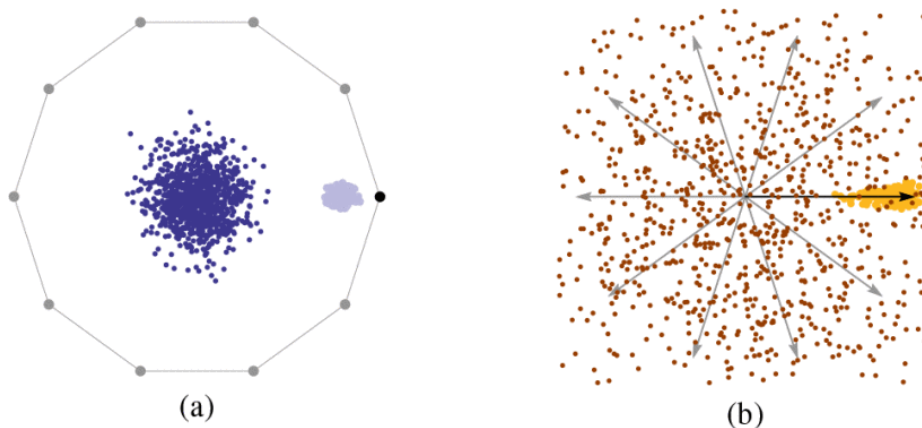
Saviorganizuojantis žemėlapis (SOM) – tai neuroninis tinklas, išsaugantis duomenų topologinę struktūrą; dažnai naudojamas daugiamačių duomenų vizualizavime. Aibė daugiamatės erdvės iš n dimensijų vektorių yra klasterizuojami į mažesnę aibę n -mačių taškų [2]. Neuroninis tinklas yra netiesinis duomenų projekcijos metodas, galintis sudaryti apibendrintą neišreikštą vaizdavimo funkciją [15]. Modelyje kaimyninės neuronų tinklo ląstelės konkuruoja tarpusavyje dėl šoninės sąveikos ir vystosi kaip specifiniai skirtingų signalų detektoriai. Šios kategorijos mokymasis vadinamas mokymusi be mokytojo, arba saviorganizuojamuoju [17]. Saviorganizacija yra fundamentalus pasikartojimų atpažinimo procesas, kurio metu išmokstami duomenims būdingi vidiniai sąryšiai. SOM gebėjimas išsaugoti topologiją svarbus siekiant išgauti ir vizualizuoti reikšmingus sąryšius. ViSOM – SOM patobulinimas, kuris labiau akcentuoja lokalių atstumų išsaugojimą vaizdavime, yra tinkamas duomenims, kuriuose labai didelis netiesiškumas. SOM metodas sukuriama rezultatais yra panašus į daugiamačių skalių metodą [15]. Dar viena SOM modifikacija - poliarinis saviorganizuojantis žemėlapis (PolSOM). PolSOM sudarytas iš dvimačio poliarinio vaizdavimo su dviem kintamaisiais – spinduliu ir kampu, kurie atitinka duomenų svorį ir požymį. Palyginus su tradiciniais algoritmais, projektuojančiais duomenis Dekarto koordinatėse ir naudojančiais Euklido atstumą kaip vienintelį kintamąjį, PolSOM ne vien išsaugo duomenų topologiją ir atstumą tarp neuronų, bet ir vizualizuoja klasterių skirtumus pagal svorį ir požymius [18].

1.1.3. Spindulinė vizualizacija

Spindulinės vizualizacijos metodai kintamuosius išdėsto apskritimu, taip pat gali būti sutinkamos elipsės ar spiralės formos projekcijos. Šie metodai kartais apibrėžiami kaip projekcijos iš Dekarto į polinę koordinatėse sistemą [19]. Vieni populiariausių tarp spindulinės vizualizacijos metodų yra *RadViz* ir žvaigždžių koordinatės. Šie metodai taikomi įvairiose srityse (bioinformatika, inžinerija,

finansai) ir gali praversti atliekant tiriamąją duomenų analizę: nustatant galimus klasterius, išskirtis. Esminis skirtumas tarp šių dviejų metodo – *RadViz* normavimo žingsnis, paverčiantis metodą netiesiniu [20].

Spindulinės vizualizacijos metodas *RadViz* projektuoja daugiamacių taškus į dvimatę erdvę. Tai netiesinis metodas, kuriame dimensijos išdėstomos vienodais atstumais ant vienetinio apskritimo, o stebėjimų taškai vaizduojami apskritimo viduje, remiantis jėgos veikiamų spyruoklių principu. Jeigu visų dimensijų koordinatės turi vienodą reikšmę, taškas bus piešiamas apskritimo centre; taškas, kurio viena ar kelios koordinatės yra didesnės nei likusios, bus piešiamas arčiau didesnę reikšmę turinčių kintamųjų. Šis metodas pirmiausia buvo pasiūlytas DNR sekų vizualizavimui. Metodo privalumas – parodo, kuris kintamasis daro didžiausią įtaką duomenų taško pozicijai, tuo pačiu ir klasterio formavimuisi. *RadViz* gerai išskiria retus duomenis: dažni taškai brėžiami arti centro, o reti – toliau. Retų duomenų vizualizacija matoma 2 pav. (a), kur dažni taškai vaizduojami centre, o šviesūs taškai, priklausantys retų duomenų grupei, piešiami arti apskritimo taško; vizualizacija lyginama su žvaigždžių koordinatinių metodų gautu grafiku (2 pav. b), kur geltona spalva vaizduojami reti duomenų taškai visai neišskiriami. Naudojant šį metodą, neretai susiduriama su taškų persidengimo trūkumu; taip pat svarbu atsižvelgti į algoritmo taikomą normavimą, nes jis gali lemti netiesinių iškreipimų duomenyse atsiradimą, netikslumų vaizduojant atstumus tarp taškų ar aptinkant išskirtis [2, 9, 20, 21].



2 pav. *RadViz* (a) ir Žvaigždžių koordinatinių (b) metodų palyginimas retų duomenų vizualizavimui [20]

Yra sukurta *RadViz* patobulinimų, pavyzdžiui, *RadViz++*, išdėstantis atraminius taškus pagal kintamųjų panašumą (naudojamas Pearson'o koreliacijos koeficientas ir vidutinės jungties aglomeracinis hierarchinis klasterizavimas) [19]. Vektorizuotas *RadViz* naudojamas klasterių vizualizavimui, atlieka duomenų tiesinimą, vėliau duomenys paverčiami į dvejetainį formatą, padidėja dimensijų skaičius – jų išdėstymas ant apskritimo krašto tampa lankstesnis, kas lemia geresnę klasterių vizualizavimą [22]. *RadViz Deluxe* vizualizacijos rezultatas – atstumais paskirstytas išdėstymas, su netolygiai perimetre išdėstytais atraminiais taškais. Įgyvendinamas trijų pakopų tobulinimo algoritmas: pirmiausia, atraminiai taškai išdėstomi pagal apytikslį Hamiltono ciklo sprendinį, toliau paklaida sumažinama taikant iteracinį algoritmą, galiausiai koreguojama duomenų taškų padėtis, remiantis jėgos krypties išdėstymu [19].

Žvaigždžių koordinatinių metodas yra tiesinis, jis leidžia vizualizuoti daugiamacių duomenis koordinatinių ašis išdėstant ant apskritimo dvimatėje erdvėje. Ašių skaičius atitinka dimensijų (kintamųjų) skaičių [2]. Koordinatinių centras sutampa su apskritimo centru, o daugiamatis taškas

atvaizduojamas sumuojant kiekvienos koordinatės vienetinius vektorius ir dauginant iš daugiamačio taško reikšmės. Taip pat sukurtas šio metodo plėtinys trimatėje erdvėje – šis metodas geriau gali atvaizduoti klasterius daugiamačiuose duomenyse [9].

Žvaigždžių koordinatės ypač tinka hierarchinę struktūrą turinčioms duomenų aibėms vizualizuoti. Dar viena taikymo sritis – sprendimų priėmimas; vizualizacija gali būti sudėliojama tokiu būdu, kad teigiamus veiksnius lemiantys kintamieji atsidurtų vienoje pusėje, neigiamus – priešingoje. Normavimas modifikuojamas atsižvelgiant į veiksmų svarbą (svarbesniems padidinamas). Tokiu būdu gautame grafike galima apžvelgti kintamųjų išsidėstymą, bendras sąveikas [10].

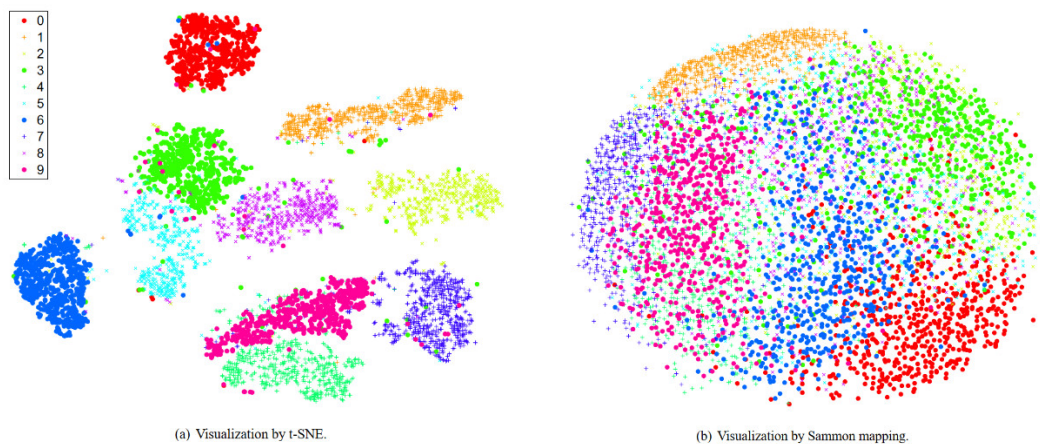
1.1.4. Principinių komponentių analizė

Principinių komponentių analizė (PCA) – tiesinis dimensių mažinimo metodas. Konstruojamas toks duomenų vaizdavimas mažai dimensių turinčioje erdvėje, kuris aprašo kuo daugiau dispersijos. Dimensijos sumažinamos iki nurodyto komponentių skaičiaus, pirmosios komponentės atspindi didžiausią dalį dispersijos [2]. PCA sėkmingai taikomas daugelyje sričių, sprendžiant, pavyzdžiui, veido atpažinimo, monetų klasifikavimo uždavinius [5]. Jei turimi duomenys yra sudėtingos struktūros, jo taikymo galimybės yra ribotos. PCA metodas negali atpažinti netiesinių, sudėtingesnių duomenų ryšių, aprašomų aukštesne nei antrojo laipsnio statistika. Yra sukurta plėtinių, tinkančių ir netiesiniams atvejams – apibendrintas PCA (angl. *generalised PCA*), branduolinis PCA (angl. *kernel PCA*), lokali tiesinė rekonstrukcija (angl. *local linear embedding* - LLE) [15]. Dar vienas trūkumas atsiranda esant dideliame dimensių skaičiui pradiniuose duomenyse: kovariacijų matricos dydis yra proporcingas dimensioms, todėl tikrinių reikšmių skaičiavimas gali tapti neįmanomas (tokiu atveju siūloma naudoti PCA alternatyvas, pavyzdžiui, tikimybinį PCA) [5].

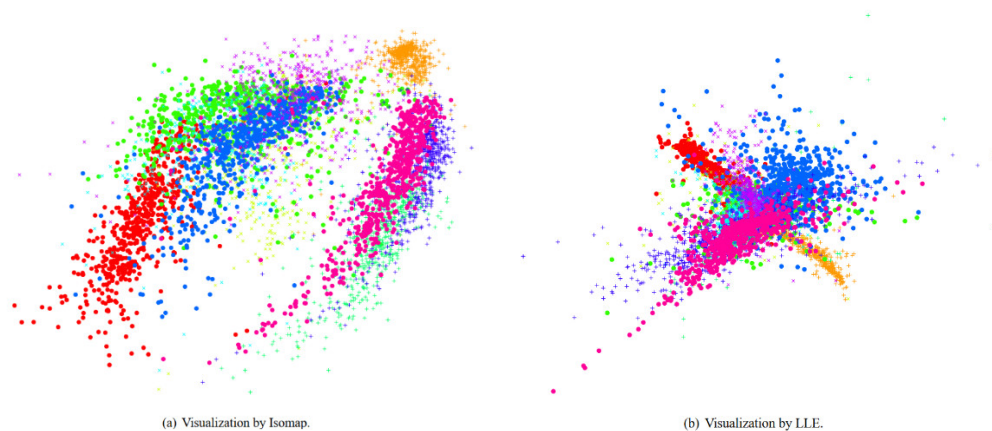
1.1.5. *t* paskirstyta stochastinė kaimyno rekonstrukcija

t paskirstyta stochastinė kaimyno rekonstrukcija (angl. *t-distributed stochastic neighbor embedding* – t-SNE) yra netiesinis dimensių mažinimo metodas, skirtas duomenų vizualizavimui dvimatėje arba trimatėje erdvėje. Sukurtame vaizdavime atsispindi ir lokali, ir globali duomenų struktūra. t-SNE yra stochastinės kaimyno rekonstrukcijos (SNE) modifikacija, kuri lengviau optimizuojama bei mažiau linkusi sutelkti taškus į grafiko centrą. Pagrindiniai skirtumai: t-SNE naudojama simetrinė SNE klaidos funkcijos versija bei Stjudento *t* skirstinys (vietoj Gauso skirstinio SNE atveju) panašumų skaičiavimui mažų dimensių erdvėje [23].

Buvo palyginti t-SNE su kitais metodais: *Sammon*, *Isomap*, LLE [23]. Galima matyti, kad *Sammon* metodo klaidos funkcija didelę reikšmę skiria mažų atstumų išlaikymui (3 pav., b), *Isomap* labiau akcentuojami dideli atstumai (4 pav., a), LLE didžiausias trūkumas – kovariacijos apribojimas, kuris neleidžia visiems taškams susigrūsti į vieną vietą (4 pav., b); apribojimą lengva apeiti, jei dauguma taškų yra vienoje krūvoje ir tik keletas yra nutolę dideliu atstumu; tokiu būdu pasiekama didelė kovariacija [23].



3 pav. (a) t-SNE, (b) *Sammon* metodų vizualizacijos MNIST duomenų rinkiniui [23]



4 pav. (a) *Isomap*, (b) LLE metodų vizualizacijos MNIST duomenų rinkiniui [23]

Metodas yra tinkamas vizualizavimui, nes gerai veikia mažinant dimensijas iki dviejų ar trijų, tačiau dėl naudojamo Stjudento t skirstinio nėra aišku, kaip t-SNE veikia projektuojant į didesnę nei trimatę erdvę (priešingai nei PCA, kurį galima taikyti įvairiam dimensijų skaičiui gauti). Taip pat pažymėtina, kad t-SNE dimensijas mažina pagal lokalias duomenų savybes, jei duomenys turi labai didelį dimensijų skaičių (daugiau nei 100) metodas veikia sunkiau. t-SNE klaidos funkcija nebūtinai konverguoja į globalų optimumą. Funkcija nėra išgaubta, optimizuojant turi būti parenkami keletas parametru, nuo kurių priklauso rezultatas. Dažnai skirtingiems atvejams tinka tie patys parametrai, o surastas optimumas nesikeičia, todėl šis trūkumas nėra laikomas svarbiu [23]. Galima rasti t-SNE metodo patobulinimų: naudojant sprendimų medžiais paremtus algoritmus, sukurtas pagreitinimas, leidžiantis dirbti su milijonus stebinių turinčiomis duomenų imtimis [24]. Sukurtas t-SNE variantas su laiko vėlinimu, pritaikytas retai pasitaikančių perėjimų molekulinėje sistemoje vizualizavimui. Ši t-SNE modifikacija su laiko vėlinimu veiksmingai atskiria būsenas pagal atstumą laike, galima vizualizuoti pagrindines tiriamų sistemų būsenas [25].

1.1.6. Klasterizavimo metodai daugiamačiams duomenims

Klasterizavimo metodai dirbant su daugiamačiais duomenimis praverčia siekiant sugrupuoti panašius stebėjimus į poaibius (klasterius), remiantis tam tikru kriterijumi. Kiekvieno klasterio viduje esantys stebėjimai vienas su kitu yra panašūs, bet skiriasi nuo kituose klasteriuose esančių taškų [9]. Klasterizavimo metodai skirstomi į hierarchinius, tankio, padalinimo, modelių paremtus, tinklelių paremtus. Hierarchiniai metodai sukuria struktūrą, panašią į medžio: pradžioje laikoma, kad

kiekvienas elementas sudaro atskirą klasterį, tuomet klasteriai apjungiami pagal tam tikrą kriterijų (vienos jungties, *Ward*, pilnosios jungties ir pan.). Rezultatas vizualiai vaizduojamas dendrograma [2].

1.2. Atraktorius fazinėje plokštumoje

Rekonstrukcija fazinėje plokštumoje yra vienas galingiausių netiesinių sistemų analizės metodų [26]. Šis metodas praverčia, kai sistema sudėtinga, jos neišeina tiesiogiai išmatuoti ar aprašyti diferencialinėmis lygtimis. Atvaizdavimai fazinėje plokštumoje, galintys atskleisti sistemos dinamiką iš geometrinės perspektyvos, vadinami faziniais portretais, arba keistais atraktoriais (dėl neprognuojamų trajektorijų) [27]. Chaoso reiškinys realiame pasaulyje yra chaotiškos laiko eilutės duomenys. Chaotiškos eilutės laiko atžvilgiu yra stochastinės, tačiau fazinės erdvės struktūra atskleidžia determinuotą elgseną, todėl chaotiškas laiko eilutes fazinėje erdvėje galima analizuoti ir prognozuoti [28].

Netiesinės dinaminės sistemos gali būti rekonstruojamos fazinėje erdvėje, tam dažniausiai naudojami vienos dimensijos stebėjimai. Dažniausiai naudojamas laiko vėlinimų metodas, kuris išplečia vienmatę laiko eilutę į daugiamatę erdvę ir tokiu būdu atskleidžia paslėptą dinaminės sistemos informaciją [29]. Dinaminės sistemos būsena laiko momentu t gali būti atvaizduojama rekonstruotoje fazinėje erdvėje vektoriumi $X(t)$, kuris atitinka vienmatį stebėjimą iš originalios laiko eilutės. Rekonstruotos erdvės savybės priklauso nuo dimensijų skaičiaus m ir vėlinimų τ parinkimo [27].

Galimas atraktoriaus kūrimas paremtas išvestinių skaičiavimu, vadinamas diferencialiniu rekonstravimu. Taikant šį metodą yra lengviau parinkti parametrus (kaip daugiamatės koordinatės vietoj vėlinimų naudojamos išvestinės). Tačiau praktikoje metodas taikomas retai, nes esant triukšmingiems duomenims yra sudėtinga įvertinti išvestines (tam naudojami centrinių skirtumų, diskrečiųjų *Legendre* daugianarių ir kiti metodai) [30].

Būsenų erdvės (angl. *state-space*) rekonstrukcijos metodas yra panašus į fazinės erdvės, tačiau tai yra įterptų duomenų geometrinė analizė siekiant nustatyti ryšį tarp praeities ir ateities laiko eilutės taškų. Taip pat naudojami laiko vėlinimai, netiesiškumai aprašomi lokaliomis tiesinėmis aproksimacijomis. Šis metodas leidžia atlikti trumpalaikę laiko eilutės būsimos elgsenos prognozę, naudojant praeities reikšmėmis pagrįstą informaciją [28].

1.2.1. Dimensijų ir vėlinimo vektoriaus parinkimo metodai

Dinaminės sistemos rekonstravimui svarbu tinkamai parinkti parametrus: dimensijų skaičių ir vėlinimų dydį. Tai nelengvas uždavinys, o netinkamai parinkus, gali būti klaidingai vaizduojama netiesinė sudėtinga struktūra, nors sistema yra tiesinė [31]. Dauguma laiko eilučių duomenų yra baigtiniai ir triukšmingi, tai šių parametų parinkimą apsunkina. Šie parametrai gali būti parenkami nepriklausomai vienas nuo kito [32], arba priklausomai – yra sukurta įvairių parinkimo metodų [31].

Dimensijų parinkimui gali būti naudojamas netikro artimiausio kaimyno algoritmas. Pagrindinė mintis yra ta, kad artimi kaimynai gali tapti tolimais taškais, kai dimensijų skaičius padidėja [26]. Dimensijų skaičius yra svarbus parametras, tiek per didelė, tiek per maža dimensija gali neigiamai paveikti rezultatus. Dėl skaičiaus parinkimo remiamasi Takens'o [33] teorema, kuri nurodo minimalią dimensijų skaičiaus ribą, reikalingą atkurti originalios sistemos dinamikai: įrodyta, kad dimensija m fazinėje erdvėje turėtų tenkinti sąlygą $m > 2d + 1$, čia d – pradinių duomenų dimensija.

Tačiau tai yra pakankama sąlyga: priklausomai nuo duomenų, atraktorius gali būti atkurtas net tada, kai m yra mažesnis [28]. Dažnai patariama dimensijų skaičių išlaikyti kuo įmanoma mažesnę [27]. Vis dėlto, jei m per mažas, rekonstrukcija būtų topologiškai neteisinga. Rinktis didesnes bei būtina dimensijas trukdo skaičiavimų dydis, taip pat atsiranda triukšmo problema: esant per didelei rekonstrukcijos dimensijai, duomenyse esantis triukšmas sumažins atraktoriaus taškų tankį. Triukšmas, priešingai nei deterministinė duomenų komponentė, atkurtame atraktoriuje užpildo kiekvieną fazinės plokštumos dimensiją. Taigi, padidinus rekonstrukcijos dimensiją daugiau, nei minimaliai reikia, be reikalo padidėja duomenų užterštumo lygis [28].

Vėlinimai turėtų būti parenkami tokie, kad rekonstruotos erdvės vektoriai būtų kuo labiau nepriklausomi, tačiau kartu ir atspindėtų pradinės sistemos koreliacijas. Vėlinimų reikšmių optimizavimas (šių dviejų savybių suderinimas) yra pagrindinis šio metodo uždavinys [34]. Vėlinimų parinkimas gali būti atliekamas įvairiai. Takens'o [33] teorema paremta begaliniais, triukšmo neturinčiais duomenimis, todėl praktikoje sunkiai pritaikoma laiko vėlinimų dydžio parinkimui. Renkant vėlinimų parametrų nepriklausomai nuo dimensijos, vienmatėms laiko eilutėms yra pasiūlytas metodas rinktis vėlinimą, atitinkantį pirmą bendros informacijos funkcijos minimumą. Taip pat naudojamas autokoreliacijos funkcijos metodas [27]. Tačiau bendros informacijos funkcijos naudojimas susilaukia ir kritikos dėl sunkiai atliekamų skaičiavimų, autokoreliacijos funkcija atsižvelgia tik į tiesines priklausomybes, todėl esant netiesiškumams jie nebūtų tinkamai įvertinti [31]. Vėlinimams nustatyti yra pasiūlyta naudoti ir artimiausio kaimyno metodą [26].

Yra atlikta tyrimų, kuriais nustatyta, jog laiko eilutėms esant baigtinėms ir triukšmingoms, dimensijos ir vėlinimo parametrai yra priklausomi vienas nuo kito [31]. Vėlinimų ir dimensijų skaičiaus priklausomybė išreiškiama [32]:

$$\tau_w = (m - 1)\tau; \tag{1}$$

čia τ – vėlinimo dydis; τ_w – optimalus vėlinimo langas, atitinkantis globalų minimumą; m – dimensijų skaičius. Laiko vėlinimų langą siūloma parinkti C-C metodu, skaičiuojant koreliacijų integralą. C-C metodas neturi stipraus teorinio pagrindimo, tačiau praktikoje veikia gerai siekiant parinkti vėlinimų laiką ir rekonstrukcijos dimensiją [32]. Tai vienas populiariausių metodų, surandantis kartu abu reikiamus parametrus. C-C metodas yra nesudėtingas skaičiavimų prasme, pasižymi patikimumo esant baigtinėms imtims [31]. Įprastinis C-C metodas gali įvertinti tik tikslų vėlinimo laiką, tačiau netinka optimaliam laiko langui įvertinti; tam naudojamas patobulintas C-C metodas [32].

Tyrimė [31] pasiūlytas atraktoriaus parametrų parinkimo metodas, paremtas simboline dinamika ir informacijos teorija. Pagrindinė simbolinės dinamikos idėja – padalinti fazinę erdvę į baigtinį poaibių skaičių ir kiekvieną poaibį pažymėti abėcėlės raide. Simbolizavimas gali padidinti informacijos, skirtos laiko modeliams apibūdinti, paieškos ir kiekybinio įvertinimo efektyvumą. Laiko eilučių simbolizavimo procesas grindžiamas jau minėtu Takens'o laiko vėlinimo koordinacių metodu. Kaip parametrų atrankos priemonė naudojama simbolinė entropija. [31].

Vėlinimai gali būti parenkami reguliarūs arba nereguliarūs. Reguliaraus vėlinimo atveju parenkami vėlinimai yra tos pačios konstantos atkartojimas visoms dimensijoms: $[\tau, 2\tau, 3\tau, \dots, (m - 1)\tau]$ [29]. Vėlinimų vektorius X fazinėje plokštumoje laiko momentu t , su vėlinimu τ ir dimensija m aprašomas išraiška [31]:

$$X_m(t) = (X_t, X_{t+\tau}, X_{t+2\tau}, \dots, X_{t+(m-1)\tau}); \tag{2}$$

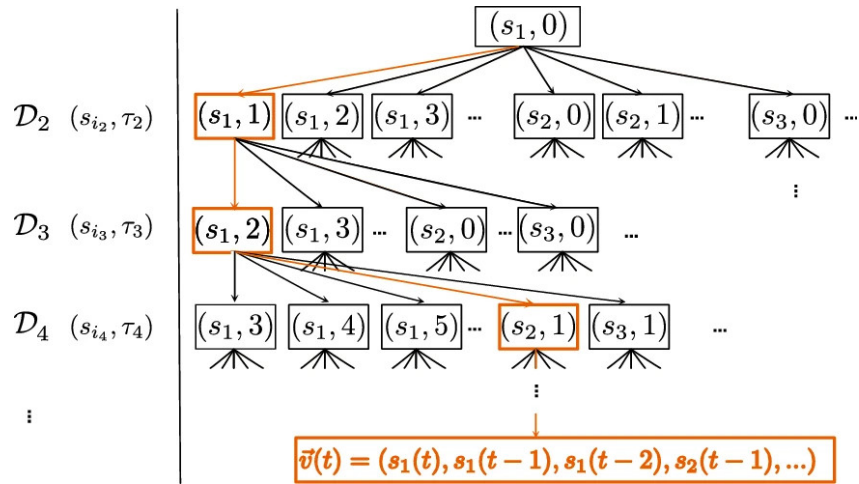
čia τ – vėlinimo dydis; m – dimensijų skaičius. Tokiu būdu iš pradinės laiko eilutės stebėjimo x_t , sukuriama daugiamatė rekonstrukcija su m dimensijų. Nereguliaraus vėlinimo atveju parenkami vėlinimai gali būti skirtingo dydžio kiekvienai dimensijai: $[\tau_1, \tau_2, \dots, \tau_{m-1}]$.

1.2.2. Atraktoriaus kūrimas daugiamatėse laiko eilučių atveju

Dauguma fazinės erdvės rekonstrukcijų yra atliekama vienmačių laiko eilučių atveju, o šio metodo naudojimas daugiamatėms eilutėms yra ribotas, galimai dėl duomenų sudėtingumo, taip pat dėl didelio triukšmo, skirtingų matavimų skalių [34]. Dirbant su daugiamatėmis eilutėmis, praktinių siūlymų dėl parametru parinkimo yra mažai: singuliarių reikšmių dekompozicijos (angl. *singular value decomposition* – SVD) metodo taikymas tiesinėms priklausomybėms eliminuoti, *Gamma* testas, skirtas nereguliarioms laiko eilučių rekonstrukcijoms parinkti; šio testo taikymas gali būti išplečiamas ir daugiamatėse laiko eilutėse dažnai naudojamas reguliarių vėlinimų vektorius, tačiau daugiamatėse atveju jis sudėtingai pritaikomas, vietoj to dažnai pasirenkami nereguliarūs vėlinimai. Dirbant su daugiamatėmis duomenimis svarbu atraktoriaus kūrimui atrinkti reikšmingus, kuo mažiau priklausomus tarpusavyje kintamuosius [29].

Optimaliam vėlinimų parinkimui buvo pasiūlytas ir Monte Karlo sprendimų medžių paieška paremtas metodas, tinkantis tiek vienmatėms, tiek daugiamatėms laiko eilutėms. Rekonstrukcijos procesas modeliuojamas kaip medis, kurio lapai atitinka tikslo funkcijos reikšmes. Naudojant Monte Karlo algoritmą, medžio lapų reikšmės užpildomos įvairiais rekonstrukcijos parametru variantais, tuomet pasirenkamas sprendinys, su kuriuo tikslo funkcija įgyja mažiausią reikšmę. Algoritmas išvengia lokalių minimumo taškų, gali būti naudojamas su įvairiomis tikslo funkcijomis, pavyzdžiui, L statistika, netikro artimiausio kaimyno (FNN) statistika. Metodas garantuoja pasirinktos tikslo funkcijos optimizavimą randant optimalius rekonstrukcijos parametrus, jei duomenys medžiui atrenkami tinkamai [36].

Metodo pritaikymo schema daugiamatėms laiko eilutėms pateikiama 5 pav., kur visos įmanomos rekonstrukcijos vizualizuojami medžio schema. Naudojama tikslo funkcija (L arba FNN statistika), padedanti parinkti tinkamas vėlinimo reikšmes, atliekama medžio paieška. Kadangi kiekviename žingsnyje renkantis labiausiai tikslo funkciją minimizuojančias reikšmes tikėtinas lokalaus minimumo taško suradimas, atliekami papildomi žingsniai globalaus minimumo radimui: medžio poaibio atrinkimas atsitiktiniu būdu, suskaičiuojami tikslo funkcijos pokyčiai kiekviename cikle, galutinė reikšmė perduodama atgal į buvusius lapus – sukuriamas tiesioginis sklidimas atgal (angl. *backpropagation*) [36].



5 pav. Optimalių rekonstrukcijos parametų paieška su Monte Karlo sprendimų medžio algoritmu, čia D_i – rekonstravimo ciklas, s_i – laiko eilutės [36]

Automatizuotas vėlinimų ir dimensijos parinkimo metodas pasiūlytas [34], naudojant kaip įvestį vienmatę arba daugiamatę laiko eilutę. Skaičiuojami L statistika ir tęstinumo (angl. *continuity*) statistika. L statistika yra tikslo funkcija, kuri kiekybiškai įvertina rekonstrukcijos tinkamumą (kokybę), nepriklausomai nuo pasirinkto rekonstrukcijos metodo. Tęstinumo statistikoje optimalaus vektoriaus paieška išreiškiama kaip rekonstrukcijos vektoriaus komponentų funkcinė nepriklausomybė. L statistikos užtektų siekiant rasti optimalų atraktoriaus rekonstrukcijos vektorius, jei būtų naudojamas jėgos (angl. *brute-force*) metodas, tikrinant visas įmanomas vėlinimų kombinacijas kartojant skirtingoms statistikos parametro reikšmėms. Kadangi daugeliu atveju toks būdas skaičiavimų laiko prasme nėra įgyvendinamas, siūloma naudoti tęstinumo statistiką išankstiniam galimų vėlinimo reikšmių parinkimui, o L statistikos kriterijus naudojamas parinkti vieną iš šių variantų [34].

1.3. Vizualizacijų kokybės vertinimas

Taikant daugiamatį projekcijų metodus ir sukuriant vaizdavimą vizualioje (dvimatėje ar trimatėje) erdvėje galima stebėti duomenų taškų grupavimąsi su kaimyniniais taškais. Vis dėlto, ne visuomet projekcijoje matomos struktūros ir ryšiai atitinka pradinių daugiamatį duomenų turėtą informaciją. Izometrinis atvaizdavimas iš daugiamatės erdvės į mažiau dimensijų turinčią įmanomas tik tam tikromis sąlygomis, todėl sukurtose projekcijose dažnai pasitaiko klaidų, iškraipymų. Šiais metodais paremta vizualinė analizė daro prielaidą, jog artimi atstumai grafikuose išreiškia panašumus, tačiau jei grafikas yra su iškraipymais, toks vertinimas gali būti netikslus. Todėl svarbu analizuoti tokių iškraipymų priežastis bei apibrėžti kiekybines iškraipymų ir klaidų dydžio vertinimo metrikas [8]. Kokybės metrika vizualizacijos kontekste apibrėžiama kaip metrika, suskaičiuojama vizualizavimo etape, fiksuojanti savybes, reikalingas naudingos informacijos gavimui iš duomenų [3].

1.3.1. Vizualizacijų iškraipymo priežastys

Projekcijose esantys netikslumai gali būti įvairūs, atsirandantys skirtinguose proceso etapuose. Vizualizavimo etape atsiradusius netikslumus daugiausiai lemia grafiko skiriamoji geba ir kontrastas sklaidos diagramose. Taip pat verta paminėti, kad paslėptų horizontalių ir vertikalų ašių vienetiniai vektoriai, naudojami taškams braižyti, ekrane turi būti vienodo ilgio – tai svarbu dėl tikslesnio

panašumų atvaizdavimo. Modeliavimo etape atsirandantys netikslumai gali atsirasti dėl esamų neeuklidinių duomenų skirtumų – tuomet regimoji Euklido erdvė negali išreikšti visų jų savybių bei variacijų. Kitas variantas, kai duomenys yra euklidiniai, tačiau nėra išdėstyti plokštumoje, kuri būtų įterpta į daugiamatę Euklido erdvę (galbūt greta egzistuoja netiesinės ar netrivialios topologijos struktūros). Projektuojant tokias struktūras vykdomas tam tikras atstumų ištempimas arba suspaudimas [8].

Atliekant projekcijas susiduriama su dingusio kaimyno ir netikro kaimyno reiškiniiais, sukeliančiais iškraipymus vizualizacijoje. Dingusio kaimyno reiškinių metu taškai, esantys šalia vienas kito daugiamatėje erdvėje, projektuojami toli vienas nuo kito. Netikro kaimyno reiškinys yra priešingas pirmajam – toli vienas kuo kito originalioje erdvėje esantys taškai projektuojami greta. Kuo dažniau pasireiškia šie atvejai, tuo labiau didėja vizualizacijos neapibrėžtumas (daugėja netikslumų ir iškraipymų) [8].

1.3.2. Vizualizacijų tikslumo vertinimo metrikos

Vizualizacijų vertinimo metrikos siekia kiekybiškai įvertinti iškraipymus ir netikslumus (pavyzdžiui, netikro kaimyno arba dingusio kaimyno reiškinių dažnį). Metrikos gali būti grupuojamos į veikiančias lokaliai ir globaliai, taip pat pagal iškraipymų tipus kuriuos konkreti metrika matuoja. Galima rasti metrikų apžvalgų, klasifikuojančių jas į klasterizavimo, išskirčių, sudėtingų atsikartojimų (angl. *patterns*), savybių išsaugojimo – šių grupių pavadinimai atitinka metrikos akcentuojamus elementus [3, 8].

Klaidos funkcijomis paremtos metrikos veikia globaliai, lyginami poriniai panašumai tarp skirtingų dimensijų erdvių taškų. Kitos globalios metrikos: Spearman'o koreliacija, topografinis produktas (lyginamos nepanašumo proporcijos), globalios kokybės metrika. NIEQA kriterijus sujungia tiek lokalių, tiek globalių netikslumų analizę, konstruojant artimiausių kaimynų grafą ir lyginant jame esančius atstumus globaliame kontekste skaičiuojant koreliacijas. Pagal iškraipymo tipus metrikos gali būti skirtos nepanašumams, koreliacijai, geometriniam neatitikimams vertinti. Koreliacijomis vertinama, kiek sistemingi pokyčiai vienoje dimensijoje daro įtaką kitoms dimensijoms [3, 8].

Pažymėtina, kad globalūs metodai iš esmės vertina kiek nepanašumai ar jų koreliacijos yra paveiktos dimensijų mažinimo proceso. Lokalūs metodai geba įvertinti platesnį iškraipymų spektrą. Neretai sutinkamai tikimybiniai vertinimo kriterijai (*Kullback – Leibler* divergencijos metrika). *Procrustes* metrika remiasi prielaida, kad taškai yra įterpti į Dekarto erdvę [8]. Lokalaus tęstinumo (angl. *local continuity* – LC) metakriterijus (skaičiuojama procentinė tikrų kaimynų dalis kaimynystėje), į jį panašus Patikimumo / Tęstinumo kriterijus [37].

Yra pasiūlyta ir kitų metodų, kurie nesiremia netikrų bei dingusių kaimynų reiškiniiais. Kokybinis daugiamatinių projekcijų vertinimas gali būti atliekamas su Shepard'o diagrama. Ji naudinga, kai lyginami metodai labai skiriasi kokybe bei duomenys turi informaciją apie klases. Vertinama, kiek projekcija išsaugo klasių struktūras vizualioje erdvėje [8].

1.4. Mechaninių sistemų diagnostika

Mechaninės sistemos laikui bėgant dėvisi ir galiausiai sugenda. Diagnostikai šioje srityje naudojamų metodų tikslas – iš anksto numatyti degradacijos progresavimą, kad būtų galima imtis veiksmų prieš įvykstant gedimui, kuris neretai turi katastrofiškų padarinių. Dažniausiai stengiamasi prognozuoti

likusią naudingą veikimo trukmę (angl. *remaining useful life* – RUL), t.y. laiką, likusį iki gedimo. Tikslus orlaivių variklių likusio naudingo veikimo laiko (RUL) prognozavimas yra labai svarbus aviacijos bei kosmoso srityse, nes nuo to priklauso orlaivio saugumas. Lėktuvų varikliai yra labai sudėtingos besisukančios mašinos, ilgą laiką veikiančios ekstremaliomis aplinkos sąlygomis. Tikslios RUL prognozės gali veiksmingai sumažinti variklių techninės priežiūros išlaidas ir kartu padidinti saugumą – didžiąją dalį visų orlaivių gedimų lemia turbininio variklio gedimas [38].

Mechaninių sistemų diagnostika gali būti atliekama identifikuojant smulkius sistemos pokyčius, atsirandančius dėl degradacijos. Tam pasitelkiami sensorių duomenys (vibracijos, temperatūros, slėgio ir kt.) analizuojami laiko arba dažnių srityje [38]. Siekiant prognozuoti RUL, yra pasiūlyta įvairių algoritmų, jie gali būti skirstomi į modelių bei duomenimis paremtus metodus; kartais gali būti sutinkama ir trečioji, hibridinių metodų, grupė. Modelių paremtiems metodams reikia sukurti tikslius fizikinius / matematinius variklio modelius kartu su istoriniais prognozavimo duomenimis ir gedimo būdais. Yra pasiūlyta tokių modelių, pavyzdžiui, naudojant Kalmano filtrą. Tokio modelio privalumai: jis yra tikslus (jei pavyksta tinkamai aprašyti sistemą), nesunkiai validuojamas. Vis dėlto sukurti tikslų fizikinį modelį, tinkantį efektyviam RUL prognozavimui, yra sudėtinga, nes reikia suprasti mechaninės sistemos fizikinius dėsnius, skaičiavimai yra brangūs ir ilgai trunkantys, modelio panaudojimo antrą kartą galimybės ribotos, reikalauja paprastinančių prielaidų tyrimo [39, 40].

Dėl aukščiau išvardintų sunkumų, kylančių dirbant su fizikiniais modeliais, dažniau naudojami duomenimis grįsti modeliai. Jie siekia iširti duomenų tarpusavio sąryšius, išgauti informaciją, kuri turi įtakos lėktuvo variklio eksploatacinių savybių pablogėjimui. Pagrindiniai privalumai – greitas įgyvendinimas ir diegimas, nebrangus kūrimas, gali būti pritaikomi ir kitoms panašioms sistemoms, gali dirbti su daugiamachišiais triukšmingais duomenimis. Tačiau yra ir trūkumų – jiems reikalingas neapibrėžtumo valdymas, tikslumas gali suprastėti dėl persimokymo, naujai sukurtos sistemos dažniausiai neturi pakankamai duomenų, reikalingų jiems apmokyti (reikalingas didelis duomenų kiekis), prognozės skaičiuojamos palyginti trumpam laikui į priekį, daugiamachišiams duomenims tikslumas gali būti mažesnis [40].

Hibridinių modelių grupė mažiausiai populiarūs ir sutinkama rečiausiai, nors šie modeliai integruoja abiejų (modeliais ir duomenimis paremtų) modelių grupes. Hibridinis metodas, sukurtas su tam tikru duomenų kiekiu ir gana tiksliais matematiniais modeliais, paprastai geba modeliuoti didesnę RUL prognozavimo tikslumą. Vis dėlto hibridiniam metodui būdingi abiejų grupių trūkumai ir su jais gerokai sudėtingiau pasiekti sprendimą [40].

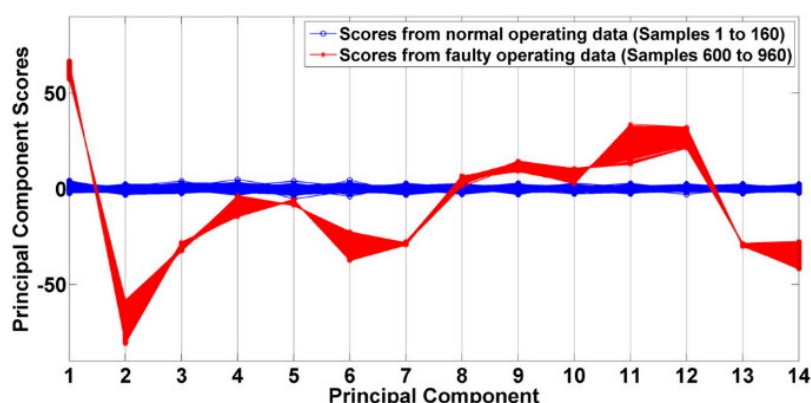
Iš duomenimis paremtų modelių, pagrinde naudojami statistiniai ir mašininio mokymosi metodai [39]. Laiko eilučių analizės metodai gali būti kombinuojami su mašininio mokymosi metodais: yra pasiūlytas autoregresinis integruotas slenkančio vidurkio metodas (ARIMA) kartu su atraminių vektorių regresija, gebantis prognozuoti RUL kelis laiko vienetus į priekį [41]; dažnai aptinkami gilus mokymosi metodai, neuroniniais tinklais paremti modeliai [38]. Buvo pasiūlytas integruotas klasterizavimo ir ilgos trumpalaikės atminties neuroninio tinklo (LSTM) algoritmas ILSTMC, tikslumu aplenkiantis LSTM tinklus, rekurentinius neuroninius tinklus (RNN) bei tiesinio programavimo metodus. Pasiūlytame algoritme įgyvendinamas daugiapakopis procesas, duomenys normuojami, kvadratinės paklaidos sumos metodu nustatomas optimalus klasterių skaičius, duomenys klasterizuojami, apmokomas LSTM tinklas ir prognozuojamas RUL [39]. Gali būti naudojami ir sąsūkų neuroniniai tinklai – buvo pasiūlyta tinklo architektūra su sąsūkos, *average* – *pooling* bei faktoriniu sluoksniu; autoregresinius modelius yra pasiūlyta derinti su paslėptaisiais

Markovo modeliais, aptinkami ir Bajeso modeliai. Taip pat gali būti naudojamas algoritmas, sudarytas iš mokymosi su priežiūra ir mokymosi be priežiūros etapų bei kombinuojant kelis neuroninių tinklų tipus. Be priežiūros apmokomas variacinis autoenkoderis, o su priežiūra – sąsūkų neuroninis tinklas ir LSTM neuroninis tinklas [42].

Tyrime [38] pasiūlytas metodas, skirtas orlaivio variklio RUL prognozuoti, pagrįstas nauju modifikuotu k-artimiausių kaimynų interpoliavimu (kNNI) su posterioriniu mažiausių kvadratų glotninimu (LSS). kNNI optimizuojamas naudojant genetinį algoritmą, integruojant jį su posterioriniu mažiausių kvadratų metodu. LSS turi tokį patį poveikį, kaip kaupiamasis ir slenkančio vidurkio filtrai [38].

1.4.1. Vizualizavimo metodų taikymas mechaninių sistemų gedimų diagnostikoje

Gedimų diagnostikai, nors ir rečiau aptinkami, gali būti pasitelkiami ir daugiamačio vizualizavimo metodai, kadangi mechaninę sistemą aprašantys duomenys dažnai būna daugiamačiai. Tyrime [43] pasiūlytas daugiamačio vizualizavimo metodas skirtas gedimų diagnostikai, naudojant PCA ir istorinius duomenis. Principinių komponentų vizualizacijoms naudojamos lygiagrečiosios koordinatės (žr. 6 pav.), gedimų diagnostikai naudojamas atsitiktinio miško mašininio mokymosi metodas (dėl atsparumo išskirtims, lygiagretizavimo galimybių) [43].



6 pav. Gedimo vizualizavimas lygiagrečiosiomis koordinatėmis: mėlyna spalva vaizduojama normaliai veikianti sistema, raudona – gedimas [43]

Kitas mechaninių sistemų diagnostikai pasiūlytas algoritmas [44], sukurtas planetinių pavarų dėžių gedimo aptikimams, naudoja t-SNE metodą dimensijų mažinimui ir gedimų vizualizavimui, atpažįstami net kelių tipų gedimai. Naudojami laiko srities (vidurkis, standartinis nuokrypis, maksimumas ir kt.) ir dažnių srities (spektrinės energijos, dažnių juostos energijos pasiskirstymo, spektrinio maksimumo padėties pokyčio, spektrinės linijos dispersijos ir pagrindinės juostos padėties pokyčio įvykus gedimams) duomenys. Klasterizavimo rezultatai lyginami su PCA, LLE, SNE. t-SNE metodas veikia tiksliausiai, klasterizuodamas skirtingus gedimus net 100 proc. tikslumu [44].

Tyrime [45] pristatoma nauja diagnostikos sistema, kurioje laiko ir erdvės modelių (angl. *spatiotemporal pattern*) tinklo (STPN) metodas derinamas su sąsūkų neuroniniais tinklais (CNN), siekiant sukurti hibridinį ST-CNN algoritmą. Algoritmas yra ne mažiau (arba labiau) tikslus nei negilieji metodai (atraminių vektorių mašina ir atsitiktinis miškas) ir vienmatis CNN. Naudojamas t-SNE metodas; juo vizualizuojant aktyvacijas nustatyta, kad erdviniai požymiai, naudojami modeliuose, gali padidinti diagnostikos tikslumą. Dar gilesnė požymių analizė atliekama

vizualizuojant su klasės aktyvacijos žemėlapiu (CAM) ir gradientiniu klasės aktyvacijos žemėlapiu (Grad-CAM) – sukuriama šilumos žemėlapių, leidžiantys palyginti skirtingų laiko / erdvės požymių aktyvacijas [45].

Yra pasiūlytas automatinis vibracijos šaltinių išskyrimo ir požymių vizualizavimo metodas [46], skirtas laivų dyzelinių variklių gedimams nustatyti. Naudojama Stockwell'o transformacija laiko ir dažnio signalui sukurti; patobulintu branduoliniu nepriklausomų komponentų analizės metodu buvo atskirtas tik su gedimu susijęs šaltinis. Tada gedimo požymiams išskirti ir vizualizuoti panaudotas t-SNE. Galiausiai, siekiant aptikti variklio gedimus ir jų tipus, sukurtas ekstremalaus mokymosi mašina (angl. *extreme learning machine*) pagrįstas klasifikatorius. Buvo palygintos LLE, PCA ir t-SNE vizualizacijos. Algoritmas geriausiai veikia naudojant t-SNE, nes gali vizualiai atskirti vibracijos šaltinį, reikalingą atskirų gedimo požymių išskyrimui [46].

Sukurtas akustinis diagnostikos metodas [47], pabrėžiantis akustinio vaizdavimo svarbą besisukančios mašinos diagnostikai, kai jis tinkamai derinamas su gedimų nustatymo metodais. Mašinos skleidžiamam akustiniam laukui įrašyti naudojamas mikrofonų rinkinys; gaunama erdvės funkcija – kintantys mašinos skleidžiami laiko signalai. Spektrinis ekscesas, kuris išryškina signalo impulsyvumą kaip dažnio funkciją, ir akustinio vaizdavimo derinys leidžia vizualizuoti impulsyvumą kaip erdvės ir dažnio funkciją. Tokia vizualizacija leidžia lokalizuoti impulsinius šaltinius. Sugebėję mašinos komponentai pasižymi impulsyviu elgesiu, tad jie išskiriami [47].

Gedimų diagnozavimo problema besisukančiuose guoliuose sprendžiama klasifikavimo ir atpažinimo būdu, naudojant prižiūrimą dimensijų mažinimo metodą: pėdsakų santykių (angl. *trace ratio*) tiesinę diskriminantinę analizę (TR-LDA). Tai LDA variantas, kurio apskaičiuota projekcijos matrica yra ortogonalė, o duomenų taškų panašumai vertinami pagal Euklido atstumą. Vizualizacijos lyginamos su kitais dimensijų mažinimo metodais (PCA, LDA ir kt.): TR-LDA pasiekia aukščiausią tikslumą [48].

1.5. Baigiamojo projekto temos ir uždavinių pagrindimas

Daugiamačių duomenų vizualizavimo metodų įvairovė yra nemaža, todėl tiriant duomenis reikia parinkti tam atvejui tinkamiausius, daugiausiai informacijos apie duomenis atskleidžiančius metodus. Mechaninių sistemų diagnostikos kontekste siekiama parinkti tokius metodus, kurie kuo anksčiau leistų pamatyti atsiradusį (ar ateityje atsirasantį) gedimą. Gedimus diagnozuoti yra svarbu, nes dauguma jų, jei laiku nepastebėti, gali baigtis nelaimingais atsitikimais. Kuo anksčiau identifikuojami mechaninės sistemos darbo pasikeitimai, tuo greičiau ir su mažesnėmis išlaidomis ji gali būti taisoma.

Tyrime dirbama su daugiamačiais orlaivių variklių duomenimis, siekiant sukurti tokias jų vizualizacijas, kad būtų galima išvelgti mechaninės sistemos likusios naudingos darbo trukmės sumažėjimą. Orlaivių srityje ypač svarbu, kad variklis veiktų tinkamai, nes gedimai skrydžio metu gali turėti katastrofiškų padarinių. Darbe apžvelgiami ir praktiškai ištiriami klasikiniai daugiamačių duomenų vizualizavimo metodai, siekiant įvertinti, kaip jie padeda pastebėti mechaninėje sistemoje atsiradusius gedimus. Taip pat yra pasiūloma plėtinių, pritaikomas rekonstrukcijos fazinėje plokštumoje metodas. Atraktoriaus rekonstrukcija gali padėti išvelgti dinaminės sistemos pokyčius; tyrime vertinama atraktoriaus rekonstravimo įtaka vizualizacijoms – ar padeda vizualiai labiau išskirti likusios naudingos darbo trukmės sumažėjimą.

2. Duomenys ir tyrimo metodai

2.1. Duomenys

Tyrime naudoti turboventiliatorinio variklio degradacijos simuliacijos duomenys iš Nacionalinės aeronautikos ir kosmoso administracijos (NASA, angl. *Ames Prognostics Center of Excellence*) [49]. Juose pateikiama sumodeliuota 100 lėktuvų variklių sensorių informacija. Varikliai bandymo pradžioje yra skirtingo susidėvėjimo, tačiau visi laikomi normaliai veikiančiais. Modeliuoti keturi skirtingi rinkiniai, kuriuose buvo skirtingi darbo sąlygų ir gedimų režimų deriniai. Tyrime naudojamas „FD001 Train“ duomenų rinkinys, jame pateikiamos laiko eilutės iki variklio gedimo. Duomenų rinkinio kintamieji pateikiami 2 lentelėje, o išsamesni kiekvieno sensoriaus aprašymai bei duomenų modeliavimo proceso metodika pateikiami straipsnyje [50].

2 lentelė. Duomenų rinkinio kintamieji ir jų reikšmės

Kintamojo pavadinimas	Reikšmė, matavimo vienetai
<i>unit_number</i>	Variklio numeris
<i>time_units</i>	Laikas, vienetais
<i>os1, os2, os3</i>	Veikimo parametrai (angl. <i>operational settings</i>)
<i>T2, T24, T30, T50</i>	Temperatūros sensorių matavimai, °R
<i>P2, P15, P30, Ps30</i>	Slėgio sensorių matavimai, psia (svarai kvadratiniam coliui)
<i>Nf, Nc, NRf, NRC, Nf_dmd, PCNfR_dmd</i>	Greičio matavimai, rpm (apsisukimai per minutę)
<i>epr, BPR, farB</i>	Santykis, atitinkamai: variklio slėgio, apėjimo, degiklio kuro / oro
<i>htBleed</i>	Nutekėjimo entalpija (angl. <i>bleed enthalpy</i>)
<i>W31_HPT, W32_LPT</i>	Vėsintuvo nutekėjimas, lbm/s (svarai per sekundę)

Tyrime buvo palikti tik labiau varijuojantys kintamieji, t.y. iš duomenų rinkinio pašalinti *os3, T2, P2, P15, epr, farB, Nf_dmd, PCNfR_dmd* stulpeliai, nes įgyja tik vieną (*P15* – dvi) reikšmę visiems stebėjimams ir yra mažai informatyvūs.

RUL reikšmė laiko momentu t apskaičiuojama kiekvienam varikliui atskirai iš laiko vienetų kintamojo *time_units*, pagal formulę:

$$RUL_t = \max(x) - x_t; \quad (3)$$

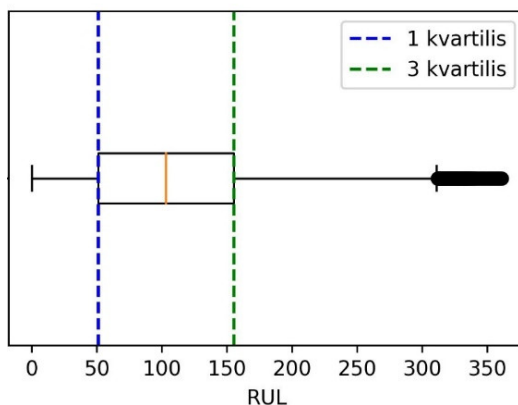
čia x – *time_units* kintamojo reikšmės; t – stebėjimų indeksas varikliui (praėję laiko vienetai nuo stebėjimų pradžios).

Taip pat RUL reikšmėms kiekvienu stebėjimo momentu priskiriamos kategorijos:

$$Kategorija = \begin{cases} \text{Aukšta, jei } RUL > 155; \\ \text{Vidutinė, jei } 51 < RUL \leq 155; \\ \text{Žema, jei } RUL \leq 51. \end{cases} \quad (4)$$

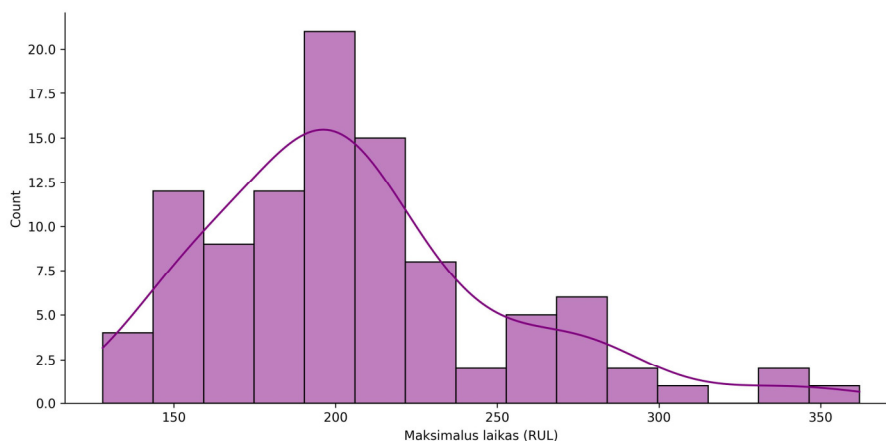
Kategorijos skirstomos pagal kvartilius (žr. 7 pav.): pirmi 25 proc. žemiausių RUL reikšmių priklauso kategorijai „Žema“, tai reiškia, jog varikliui liko 51 arba mažiau veikimo laiko vienetų. „Vidutinė“

kategorija apima nuo 1 ir 3 kvartilų reikšmes (nuo 25 iki 75 proc.), o „Aukšta“ veikimo trukmė apima paskutinio ketvirčio reikšmes, didesnes nei 155 laiko vienetai.



7 pav. Variklių veikimo laiko (RUL) kvartilų grafikas

Duomenys aprašo variklių veikimą iki gedimo momento, priklausomas kintamasis – likęs naudingas darbo laikas (RUL).



8 pav. Variklių veikimo laiko (RUL) pasiskirstymo diagrama

Iš variklių pasiskirstymo pagal pradiniu momentu buvusią veikimo trukmę diagramos (žr. 8 pav.) galima matyti, kad daugumos variklių duomenys apima 200 laiko vienetų iki gedimo, tačiau taip pat yra keletas ilgiau nei 300 laiko vienetų veikusių variklių.

2.2. Duomenų apdorojimas: filtravimas ir normavimas

Filtravimui išbandomi medianos ir slenkančio vidurkio filtrai, naudojant skirtingo dydžio langus. Slenkančio vidurkio filtras skirtas signalui glotninti. Jis veikia vidurkindamas pradinio signalo taškus pasirinktam slenkančio lango dydžiui, tokiu būdu sumažinant triukšmą. Formulė, aprašanti filtro veikimą, vidurkinimui naudojant taškus tik iš vienos filtruojamo taško pusės [51]:

$$y_i = \frac{1}{M} \sum_{j=0}^{M-1} x[i + j]; \quad (5)$$

čia x – įvesties signalas; y – išvesties signalas; M – slenkančio lango dydis.

Medianos filtras veikia panašiu slenkančio laiko principu, tik vietoj vidurkio skaičiuojama medianos reikšmė.

Duomenims atliktas normavimas dviem skirtingais būdais:

1. Standartinis normavimas su *Python sklearn.preprocessing.StandardScaler()* funkcija: atimamas vidurkis ir dalinama iš standartinio nuokrypio (toliau: *StandardScaler*).
2. Su rankiniu būdu suprogramuota normavimo funkcija, kuriai taikomas *MinMax* normavimo principas, minimalias ir maksimalias reikšmes parenkant tik pagal „Aukštos“ kategorijos (gerai veikiančio variklio) RUL duomenis (toliau: *MinMaxRUL*). *MinMaxRUL* normavimo formulė:

$$x_{norm} = \frac{x - \min(x_{RUL})}{\max(x_{RUL}) - \min(x_{RUL})}; \quad (6)$$

čia x_{norm} – normuoti duomenys; $\min(x_{RUL})$ – mažiausia x įgyjama reikšmė, imant tik tuos stebėjimus, kur RUL kategorija yra „Aukšta“; $\max(x_{RUL})$ – didžiausia x įgyjama reikšmė, imant tik tuos stebėjimus, kur RUL kategorija yra „Aukšta“.

2.3. Atraktoriaus rekonstrukcija fazinėje plokštumoje

Atraktoriaus kūrimas fazinėje plokštumoje gali padėti analizuoti dinaminės sistemos savybes ir vizualiai anksčiau pamatyti gedimus. Siekiant rekonstruoti atraktorių fazinėje plokštumoje naudojant laiko vėlinimus, pagrindiniai etapai yra:

- pradinių duomenų transformavimas (jei dirbama su daugiamačiu duomenų rinkiniu);
- parametrų parinkimas: dimensijos skaičiaus nustatymas ir vėlinimų vektoriaus optimizavimas.

Daugiamačiu atveju dažnai pasirenkama kurti nereguliarus vėlinimus (reguliaraus tipo vėlinimai dažniau aptinkami vienmačiu atveju) [29].

Tyrime bus atliekamas nereguliarus vėlinimas, t.y. vėlinimo reikšmės τ skirtingoms dimensijoms gali būti skirtingos, vėlinimų vektorius aprašomas išraiška:

$$v = [\tau_1, \tau_2, \dots, \tau_{m-1}]; \quad (7)$$

čia v – vėlinimų vektorius; τ – vėlinimo dydis; m – atraktoriaus dimensijų skaičius.

Tuomet sukurtas vaizdavimas fazinėje plokštumoje kiekvienam laiko momentui (stebėjimo indeksui t) atitinka m -matį tašką:

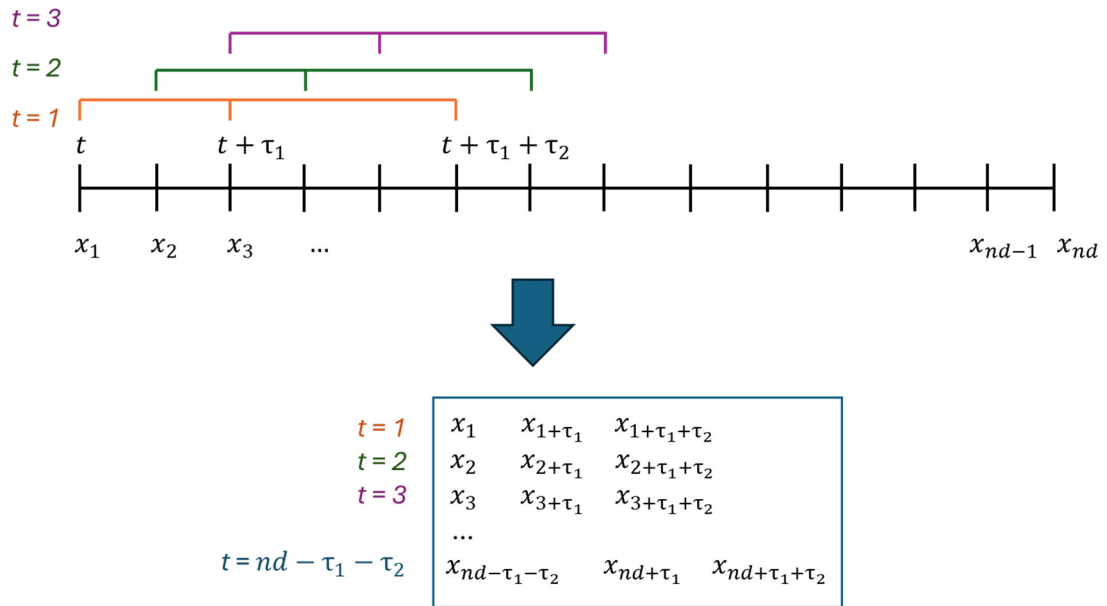
$$X_m(t) = (X_t, X_{t+\tau_1}, X_{t+\tau_1+\tau_2}, \dots, X_{t+\tau_1+\tau_2+\dots+\tau_{m-1}}); \quad (8)$$

čia X – pradinių (transformuotų į vienmatę erdvę) duomenų rinkinys (laiko eilutė).

Atraktoriaus rekonstrukcijos \mathcal{R} išraiška:

$$\mathcal{R} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1,m} \\ x_{21} & x_x & \dots & x_{2,m} \\ \vdots & & & \vdots \\ x_{t,1} & x_{t,2} & \dots & x_{t,m} \end{bmatrix}; \quad (9)$$

čia x – koordinatė fazinėje plokštumoje; m – atraktoriaus dimensijos skaičius; t - laiko indeksas. \mathcal{R} matricos eilutės atinka $X_m(t)$ esant skirtingiems laiko indeksams, stulpeliai – rekonstruotos dimensijos.



9 pav. Rekonstrukcijos fazinėje plokštumoje schema 3 dimensijų atveju

Rekonstrukcijos schema pateikiama 9 pav. galima matyti, kaip parinktu vėlinimo vektoriumi, vis perkeltiant jį vienu žingsniu į priekį, formuojami taškai fazinėje plokštumoje. Šiuo atveju konstruojami trimačiai taškai, naudojant du vėlinimus (τ_1, τ_2).

2.3.1. Rekonstrukcija fazinėje erdvėje daugiamačiu atveju

Rekonstrukcijos fazinėje plokštumoje dažniausiai atliekamos su vienmatėmis laiko eilutėmis. Daugiamačiai duomenys yra tyrinėti mažiau, su jais dirbant svarbu parinkti, koku būdu formuoti atraktorių [34]. Šiame tyrime dirbama su daugiamačiais duomenimis, atliekama jų tiesinimo transformacija naudojant *Python* funkciją *numpy.ndarray.flatten()*, rezultatas – vienmatis vektorius, sudarytas iš visų dimensijų duomenų:

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1,d} \\ x_{21} & x_{22} & \dots & x_{2,d} \\ \vdots & \vdots & \dots & \vdots \\ x_{n,1} & x_{t,2} & \dots & x_{n,d} \end{bmatrix} \rightarrow [x_{11}, x_{12}, \dots, x_{1d}, x_{21}, x_{22}, \dots, x_{2d}, \dots, x_{n-1,d}, x_{n,d}]; \quad (10)$$

čia x – duomenų taškai; n – stebėjimų skaičius; d – pradinių duomenų dimensija.

2.3.2. Rekonstruotos erdvės dimensijos parinkimas

Siekiant rekonstruoti atraktorių fazinėje plokštumoje, svarbu parinkti dimensijos ir vėlinimų parametrus, nes pagal juos sudaromas vėlinimų vektorius. Dimensijų skaičius parenkamas eksperimentiniu būdu, išbandant įvairias reikšmes ir vertinant atraktorių vizualiai. Bandomos reikšmės iš intervalo $[3, 2d + 1]$, čia d – pradinių duomenų dimensija. Trijų dimensijų rekonstrukcijos privalumas – galima tiesiogiai vizualizuoti trimatėje erdvėje, netaikant papildomų projektavimo į

mažesnę erdvę metodų. Viršutinei intervalo rimbai brėžti remiamasi Takens'o [33] teorema, nurodančia minimalų dimensijų skaičių ribą, sistemos dinamikos atkūrimui:

$$m > 2d + 1; \tag{11}$$

čia d – pradinių duomenų dimensija; m – atraktoriaus dimensija.

Verta paminėti, kad, nors Takens'o teorema nustato minimalią ribą, t.y. dvigubai didesnę atraktoriaus dimensijų skaičių nei pradiniuose duomenyse, tačiau rekonstrukcijai gali užtekti ir mažiau dimensijų [28], priklausomai nuo turimų duomenų; bendrai dimensijų skaičių rekomenduojama rinktis kuo įmanoma mažesnę [27].

Dirbant su daugiamačiais duomenimis, siūloma atraktoriaus kūrimui atrinkti kuo mažiau priklausomus tarpusavyje kintamuosius [29]. Dimensijų skaičiui parinkti tyrime panaudotas ir genetinis algoritmas. Optimizuojant vėlinimų vektorių buvo pastebėta, kad kai kurių dimensijų vėlinimai tampa lygūs 0. Tokios vėlinimų reikšmės tik atkartotų ankstesnio kintamojo reikšmes, naujos informacijos nepridėtų. Dėl šios priežasties genetiniu algoritmu gautame vektoriuje bandyta pašalinti 0 reikšmę turinčius vėlinimus, tokiu būdu sumažinant dimensijų skaičių ir paliekant tik informatyvias dimensijas.

2.3.3. Vėlinimų vektoriiaus optimizavimas genetiniu algoritmu

Vėlinimų vektorius v sudaromas iš τ reikšmių rinkinio $[\tau_1, \tau_2, \dots, \tau_{m-1}]$, kurio elementų skaičius atitinka parinktą rekonstrukcijos dimensijos skaičių m . τ reikšmių parinkimui konkrečioje m -matėje erdvėje atliekamas vektoriiaus v optimizavimas genetiniu algoritmu.

Genetinis algoritmas paremtas natūralia atranka (evoliucija), vykstančia populiacijoje. Ilgainiui rūšys, turinčios tinkamą genų derinį, tampa dominuojančiomis savo populiacijoje. Kartais genuose gali įvykti atsitiktinių pokyčių (mutacijų). Jei šie pokyčiai suteikia papildomų pranašumų, iš senųjų rūšių išsivysto naujos, priešingu atveju pokyčius pašalina natūralioji atranka [52].

Viena svarbiausių genetinio algoritmo savybių – gebėjimas išvengti lokalaus optimumo. Galimi sprendiniai yra laikomi genais, kurie derinami siekiant padidinti jų efektyvumą. Sprendinių vektorius yra laikomas individu (arba chromosoma). Algoritmas dirba su chromosomų rinkiniu, vadinamu populiacija, kuris dažniausiai inicializuojamas atsitiktinai. Vykstant paieškai, populiaciją sudaro vis tinkamesni sprendiniai, kol galiausiai ji konverguoja (ima dominuoti vienas sprendinys) [52]. Pasiūlytų sprendimų tinkamumas, vertinamas pagal prisitaikymo (angl. *fitness*) funkciją. Individai, pasižymintys geriausiais rezultatais (aukščiausia funkcijos reikšme), išsaugomi ir sujungiami siekiant sukurti naują kartą – sprendinių rinkinį [41].

Algoritmas paremtas genetiniais procesais [41]:

- Reprodukcija: naujos kartos sprendinių aibė sukurama remiantis ankstesnės kartos individualais. Dėl šios priežasties vienos kartos individai turi daugumą ankstesnės kartos individų savybių.
- Kryžminimas: genetinio algoritmo kryžminimo operatorius imituoja biologinį reprodukcinį mechanizmą, naudojant du sprendinių aibės individus kitos kartos individo kūrimui.
- Mutacija: mutacijos operatorius naudojamas sprendinių įvairovei išlaikyti. Procesas panašus biologinę mutaciją, kai nauji palikuonys gimsta su atsitiktiniais pokyčiais, neturinčiais nieko bendra su jų tėvų chromosomomis [41].

Genetinio algoritmo žingsniai [52]:

3. Pradinės populiacijos P_l generavimas (N sprendinių) ir jos tinkamumo įvertinimas, laikas $t = l$.
4. Kryžminimas: kuriama palikuonių populiacija Q_t . Pagal tinkamumo įvertinimus pasirenkami du sprendinius x ir y iš P_t , naudojamas kryžminimo operatorius ir gautas sprendinys pridodamas prie Q_t .
5. Mutacija: kiekvienas sprendinys mutuojamas pagal iš anksto nustatytą intensyvumą.
6. Tinkamumo įvertinimas: kiekvienam sprendiniui $x \in Q_t$, remiantis tikslo funkcijos reikšme, priskiriama tinkamumo vertė.
7. Atranka: Parenkami N sprendinių iš Q_t (pagal jų tinkamumą) ir perkeliama į P_{t+1} .
8. Paieška stabdoma, jei tenkinamas paieškos kriterijus. Kitu atveju nustatoma $t = t+1$ ir grįžtama į 2 punktą [52].

Tikslo funkcija tyrime apibrėžiama skaičiuojant sukurto atraktoriaus tūrį ir siekiant parinkti tokį vektorių, kad tūris būtų kuo didesnis [53]. Remiamasi prielaida, jog esant didesniai tūriui, rekonstrukcija informatyvesnė. Tūris skaičiuojamas sumuojant kiekvieno taško atstumą (modulį) iki centro:

$$V = \sum_{t=1}^T \sum_{i=1}^d |x_{i,t}| \rightarrow \max; \quad (12)$$

čia V – atraktoriaus tūris; T – laiko eilutės paskutinis indeksas; d – dimensijų skaičius pradinuose duomenyse; x – pradinių duomenų taškai. Optimizuojama pagal dar nesugėdusio, gerai veikiančio variklio duomenis.

2.4. Daugiamačių duomenų vizualizacijos metodai

Šiame skyrelyje aprašomi klasikiniai daugiamačių duomenų vizualizacijų metodai, kurie tyrime naudojami vizualizuoti pradiniam bei rekonstruoto fazinėje plokštumoje atraktoriaus duomenims. Taip pat aprašomas pasiūlytas PCA metodo plėtinys.

2.4.1. Principinių komponentių analizė

Principinių komponentių analizė (PCA) yra plačiai naudojamas klasikinis dimensijų mažinimo metodas. Pradinių duomenų koreliuoti kintamieji yra transformuojami į nekoreliuotų ortogonalų kintamųjų rinkinį – pagrindines komponentes, tuo pačiu siekiant išsaugoti svarbiausią pradinių duomenų informaciją. Metodas siekia paaiškinti kintamųjų kovariacinę struktūrą, naudojant tiesines kintamųjų kombinacijas. Matematiškai, PCA yra koreliacijos matricos kovariacijų dekompozicija. Prieš taikant metodą duomenys normuojami [43].

Pirmajai principinei komponentei keliamas reikalavimas turėti didžiausią įmanomą dispersiją. Antroji komponentė skaičiuojama taip, kad turėtų ne vien didžiausią dispersiją, bet dar reikalaujama ir ortogonalumo su pirmąja komponente. Kitos komponentės skaičiuojamos atitinkamai tokiu pačiu būdu. Šių naujų suskaičiuotų kintamųjų reikšmės stebėjimams vadinamos faktorių balais, o šiuos faktorių balus galima interpretuoti geometriškai kaip stebinių projekcijas į pagrindines komponentes [7].

Komponentės ieškomos naudojant SVD metodą, faktorių balų matrica F išreiškiama:

$$X = P\Delta Q^T; \quad (13)$$

$$F = P\Delta = P\Delta Q^T Q = XQ; \quad (14)$$

čia X – pradinių duomenų matrica; P – kairiųjų singuliarių vektorių matrica; Δ – singuliarių reikšmių diagonalinė matrica; Q – dešiniųjų singuliarių vektorių matrica, dar vadinama faktorių skaičiavimo koeficientų matrica.

2.4.2. Spindulinė vizualizacija *RadViz*

Spindulinės vizualizacijos metodas *RadViz* leidžia pavaizduoti daugiamačius duomenis dvimačiame grafike, išdėstant taškus apskritime. *RadViz* modeliuoja fizikinę spyruoklę. Kintamųjų reikšmės originalioje erdvėje turi būti neneigiamos, todėl kintamieji normuojami į intervalą $[0, 1]$. Normuojama atskirai kiekviena dimensija. Kintamieji lemia spyruoklių lankstumą. Pagal Huko dėsnį, taško reprezentacijos dvimatėje erdvėje vieta yra tokia, kurioje spyruoklių jėgų suma lygi 0 (t. y. taškas yra pusiausvyros padėtyje) [20].

Kiekvieną dimensiją atitinka atramos taškas, kurie visi yra išdėstomi ant vienetinio apskritimo krašto. Dimensijų atraminių taškų vietos randamas skaičiuojant kiekvieną iš jų atitinkantį kampą θ_i ir išdėstant tolygiai ant vienetinio apskritimo krašto:

$$\theta_j = \frac{2\pi j}{d}, \text{ kai } j = 1, \dots, d; \quad (15)$$

čia d – dimensijų skaičius pradiniuose duomenyse.

Atliekamas perėjimas iš polinių į Dekarto koordinates, tuomet atraminių taškų vektorius išreiškiamas formule:

$$v_i = (r \cos(\theta_i), r \sin(\theta_i)); \quad (16)$$

čia r – apskritimo spindulys. Dirbama su vienetiniu apskritimu, todėl $r = 1$.

Generuojamos netiesinės projekcijos dvimatėje erdvėje, kiekvienam daugiamačiui duomenų taškui priskiriant tašką dvimatį tašką, kuris žymi vietą apskritime [20]:

$$y_i = \frac{\sum_{j=1}^d x_{i,j} v_j}{\sum_{j=1}^d x_{i,j}}; \quad (17)$$

čia x – pradinių duomenų taškai; y – dvimatės erdvės taškai; v – atraminių taškų vektorius; d – daugiamatės erdvės dimensijų skaičius; $i = 1, \dots, n$; n – stebėjimų skaičius pradiniuose duomenyse.

Įstačius (16) formulę į (17), gaunama projekcijos galutinės duomenų x taškų reprezentacijos dvimatėje erdvėje $y = (y(1), y(2))$ formulės [20]:

$$y(1)_i = \frac{\sum_{j=1}^d x_{i,j} \cos \theta_j}{\sum_{j=1}^d x_{i,j}}; \quad (18)$$

$$y(2)_i = \frac{\sum_{j=1}^d x_{i,j} \sin \theta_j}{\sum_{j=1}^d x_{i,j}}. \quad (19)$$

RadViz projekcijoms būdinga, jog kuo tam tikro kintamojo reikšmės daugiamatėje erdvėje yra didesnės, palyginus su kitais kintamaisiais, tuo projekcija y bus arčiau to kintamojo atraminio taško. Jei kintamųjų reikšmės yra daugmaž vienodos, toks taškas bus projektuojamas apskritimo centre [20].

2.4.3. t-SNE vizualizavimo metodas

t-SNE yra metodas, skirtas daugiamatųjų duomenų vizualizavimui trimatėje ar dvimatėje erdvėje. Tai netiesinis metodas, išsaugantis tiek lokalią, tiek globalią duomenų struktūrą [23].

t-SNE veikia Euklido atstumus tarp daugiamatės erdvės taškų versdamas sąlyginėmis tikimybėmis, kurios reprezentuoja panašumus. Panašumas taško x_j į tašką x_i yra tikimybė p_{ij} , kad x_i pasirinktų x_j savo kaimynu. Kaimynai renkami pagal jų tikimybų tankio funkciją (naudojamas Gauso skirstinys, centruotas x_i taške). Gretimiems taškams, p_{ji} yra didelė, o plačiai išsibarsčiusiems – be galo maža [23]:

$$p_{i,j} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma^2)}{\sum_{k \neq l} \exp(-\|x_k - x_l\|^2 / 2\sigma^2)}; \quad (20)$$

čia x – duomenų taškai daugiamatėje erdvėje; y – duomenų taškų x projekcijos mažų dimensijų erdvėje; σ – Gauso skirstinio dispersija.

Mažų dimensijų erdvėje tarp taškų y_i ir y_j (daugiamatėje erdvėje atitinkančių x_i ir x_j) skaičiuojama panaši sąlyginė tikimybė q_{ij} (naudojant Stjudento t skirstinį su vienu laisvės laipsniu) [23]:

$$q_{i,j} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}}. \quad (21)$$

Jei grafiko taškai y_i ir y_j teisingai vaizduoja panašumus tarp daugiamatės erdvės taškų x_i ir x_j , tai tikimybės p_{ij} ir q_{ij} turėtų sutapti. t-SNE minimizuoja jų nesutapimus, skaičiuojant *Kullback – Leibler* (KL) divergencijų sumą. t-SNE baudos funkcija C yra simetrinė, nes $p_{ij} = p_{ji}$ ir $q_{ij} = q_{ji}$, $\forall i, j$. Taip pat pažymima $p_{ii} = q_{ii} = 0$. Minimizuojama gradientinio nusileidimo algoritmu [23].

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}; \quad (22)$$

čia P – tikimybinis skirstinys daugiamatėje erdvėje; Q – tikimybinis skirstinys mažų dimensijų erdvėje; p_{ij} – poriniai panašumai daugiamatėje erdvėje; q_{ij} – poriniai panašumai mažų dimensijų erdvėje.

2.5. Skaitinis vizualizacijų tikslumo vertinimas

2.5.1. Vidutinė kvadratinė paklaida

Vidutinė kvadratinė paklaida (angl. *mean squared error* – MSE), dar vadinama vidutiniu kvadratinu skirtumu, yra viena populiariausių kiekybinio vertinimo metriku. Ji gali būti naudojama siekiant palyginti skirtingų algoritmų efektyvumą. MSE plačiai taikoma optimizavime bei įvairių signalų apdorojimo metodų vertinime, įskaitant filtrų kūrimą, signalų suspaudimą, atkūrimą, triukšmo pašalinimą, klasifikavimą [54].

MSE naudojama siekiant palyginti du signalus, suteikiant jiems kiekybinę įvertį, apibūdinanti panašumo arba iškraipymų lygį tarp signalų. Tarkime, kad $x = \{x_i | i = 1, 2, \dots, N\}$ ir $y = \{y_i | i = 1, 2, \dots, N\}$ yra du vienodo ilgio diskretūs signalai. Tuomet MSE jiems būtų apibrėžiama [54]:

$$MSE(x, y) = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2; \quad (23)$$

čia N – signalo ilgis; x_i – i -tasis x signalo stebėjimas; y_i – i -tasis y signalo stebėjimas.

MSE turi aiškia fizikinę prasmę – tai yra vienas iš būdų apibrėžti klaidos signalo energijai. Klaidos signalas išreiškia skirtumą tarp originalaus ir iškreipto vaizdo ir nusakomas formule [54]:

$$e_i = x_i - y_i. \quad (24)$$

MSE, kaip klaidos signalo energijos matas, yra išsaugoma po bet kokios ortogonalios tiesinės transformacijos (pavyzdžiui, *Furje*). Ši energijos išsaugojimo savybė garantuoja, kad signalo iškraipymo energija transformacijos srityje išliks tokia pati kaip ir pradiniam signalo [54].

Kitos MSE būdingos naudingos savybės: išgaubtumas, simetriškumas ir diferencijuojamumas [54].

2.5.2. Vaizdų evoliucijos laike stebėjimas

Tyrime vertinama, kaip vizualizacijos kinta laike. Slenkančio lango principu įvairiais laiko momentais fiksuojamas vaizdas, tuomet skaičiuojama MSE – skirtumas tarp dviejų gretimų fiksuotų vaizdų. Toliau aprašomi konkretūs atliekami žingsniai.

Pirmiausia parenkamas slenkančio lango dydis L , lygus pusei turimo signalo ilgio, suapvalinto iki sveiko skaičiaus su *Python* funkcija *round()*. Tarkime vertinamas signalas yra N ilgio, tuomet:

$$L = \text{round}\left(\frac{N}{2}\right). \quad (25)$$

Slenkantis langas pirmiausia yra signalo pradžioje, jis perkeliamas per 1 dydžio žingsnį, kol pasiekiamas $(N-L)$ -tasis stebėjimas. Kiekviename žingsnyje fiksuojami *RadViz*, t-SNE ir PCA vaizdai $I = \{I_i | i = 0, 1, 2, \dots, N - L\}$, vizualizujantys konkretaus intervalo taškus.

Gautiems vaizdams I skaičiuojamos MSE reikšmės, lyginant du gretimus vaizdus ir skaičiuojant jų kvadratinį skirtumą. Gautos reikšmės brėžiamos grafike.

2.6. Metodinė schema

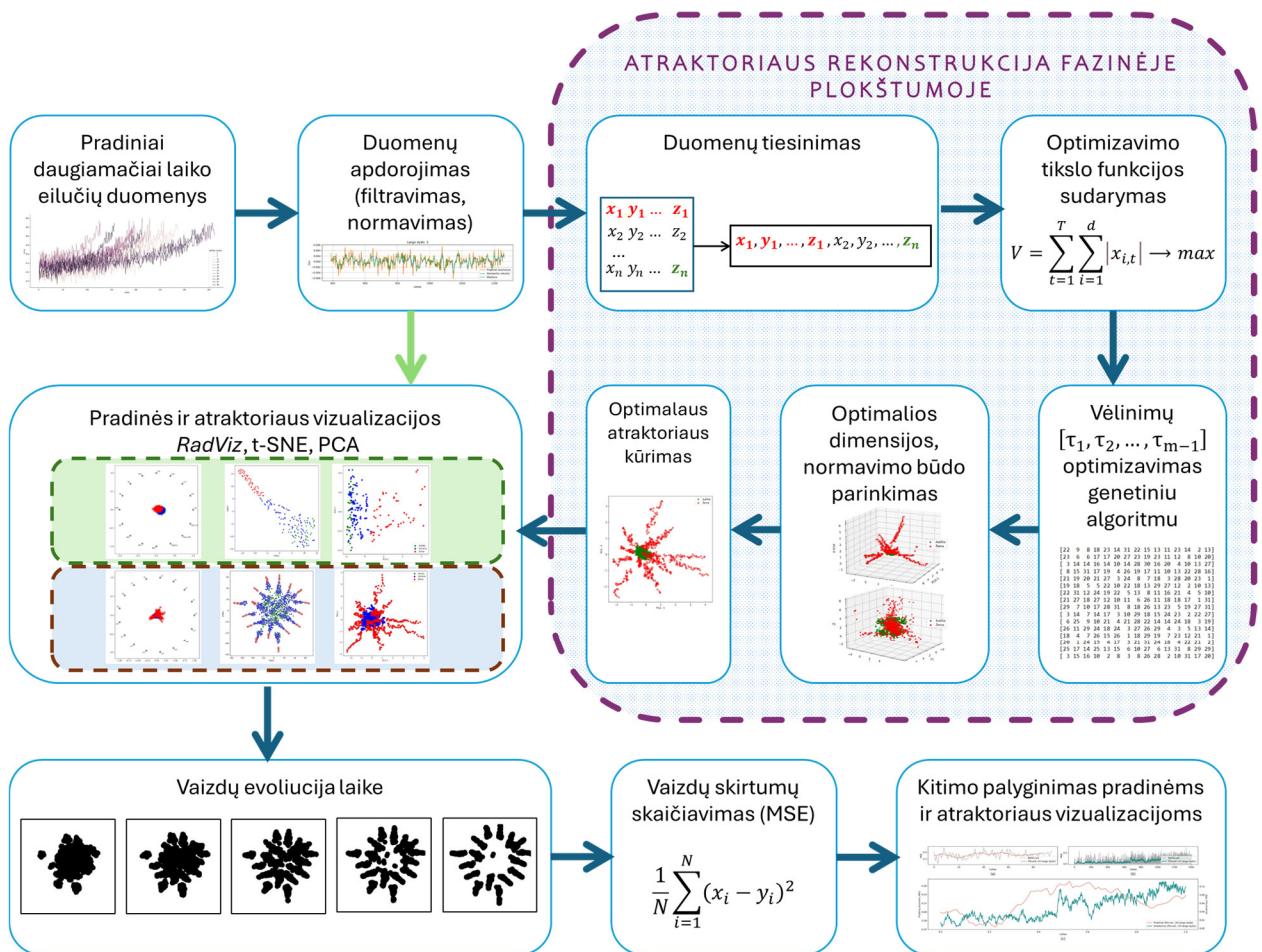
Metodinė darbo schema matoma 10 pav. Schemoje pavaizduotas procesas, apimantis daugiamačių duomenų metodų tyrimą lėktuvų variklių gedimams vizualizuoti. Pirmiausia pradiniai duomenys apdorojami (normavimas ir triukšmo šalinimas). Tuomet paruošti duomenys analizuojami dviem būdais.

Pirmiausia sukuriama pradinių duomenų vizualizacijos klasikiniai daugiamačių duomenų vizualizavimo metodais:

- PCA (tiesinis metodas);
- t-SNE (netiesinis metodas, išsaugantis lokalią bei globalią duomenų struktūrą);
- *RadViz* (netiesinis spindulinės vizualizacijos metodas).

Toliau kuriamas rekonstrukcija fazinėje plokštumoje paremtas vizualizavimo metodas. Atliekama atraktoriaus rekonstrukcija. Kadangi dirbama su daugiamačiais duomenimis, reikalingas duomenų tiesinimo žingsnis. Atraktoriaus parametrai (dimensija, vėlinimų skaičius) optimizuojami naudojant genetinį algoritmą. Išbandomi keli normavimo būdai. Šio etapo rezultatas – sukurta optimali rekonstrukcija fazinėje plokštumoje. Klasikiniais metodais vizualizuojami rekonstruoto fazinėje erdvėje atraktoriaus taškai. Galiausiai atliekamas skaitinis vaizdų evoliucijos laike vertinimas

skaičiuojant MSE gretimiems vaizdams, gautiems su slenkančio lango būdu. Vertinamos pradinė duomenų ir fazinėje erdvėje rekonstruoto atraktoriaus *RadViz*, t-SNE, PCA metodais. Stebima, kurie metodai jautriau reaguoja į sistemos pokyčius.



10 pav. Tyrimo metodinė schema

2.7. Programinė įranga

Darbas atliktas naudojant *Python* programinę įrangą. Klasikinių daugiamačių vizualizavimo metodų tyrimui naudotos šios funkcijos: PCA – *sklearn.decomposition.PCA()*; t-SNE – *sklearn.manifold.TSNE()*; *RadViz* – *pandas.plotting.radviz()*.

Vėlinimų vektoriaus optimizavimui genetiniu algoritmu naudojama *pygad* bibliotekos funkcija *pygad.GA()*.

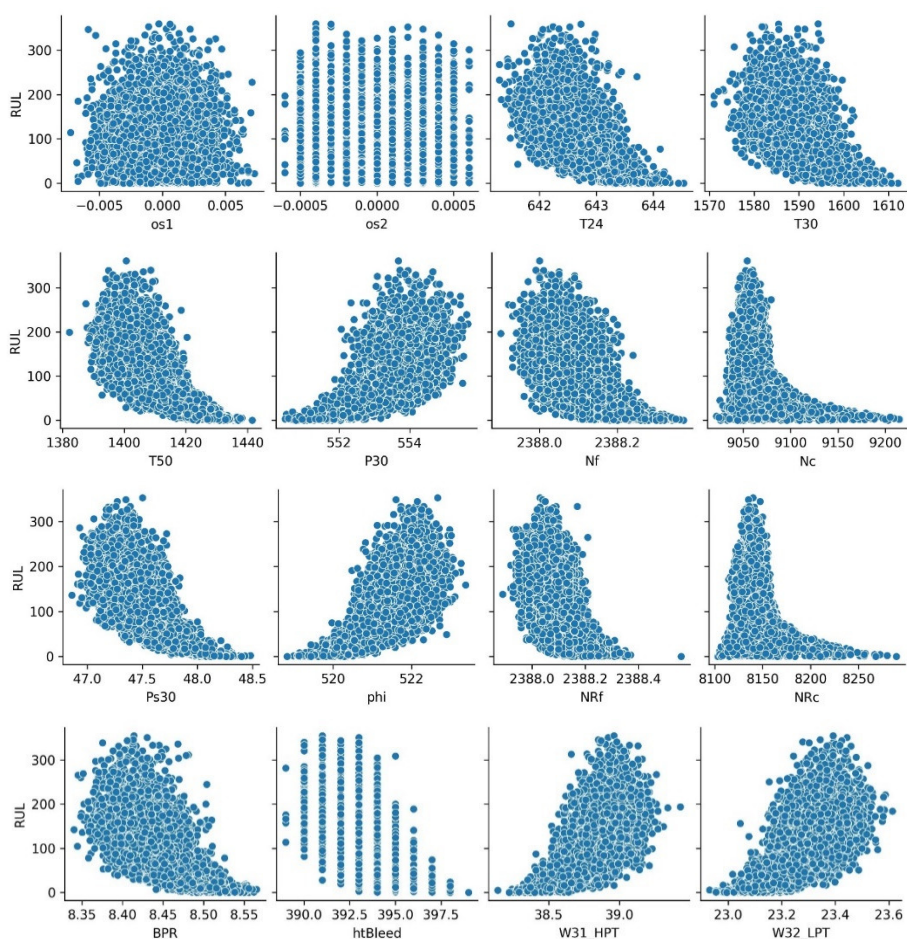
3. Tyrimų rezultatai ir jų aptarimas

Šioje dalyje aprašomi tyrimo rezultatai, pradedant duomenų tiriamąja analize, filtravimu; toliau aprašomos klasikinių daugiamačių vizualizacijos metodų pritaikymas lėktuvų variklių naudingam likusiam veikimo laikui vizualizuoti. Aprašomi atraktoriaus rekonstrukcijos proceso rezultatai, pateikiami optimalaus atraktoriaus ir pradinių duomenų klasikinių vizualizacijų vertinimai ir palyginimai.

3.1. Pradinių duomenų apdorojimas ir vizualizacijos

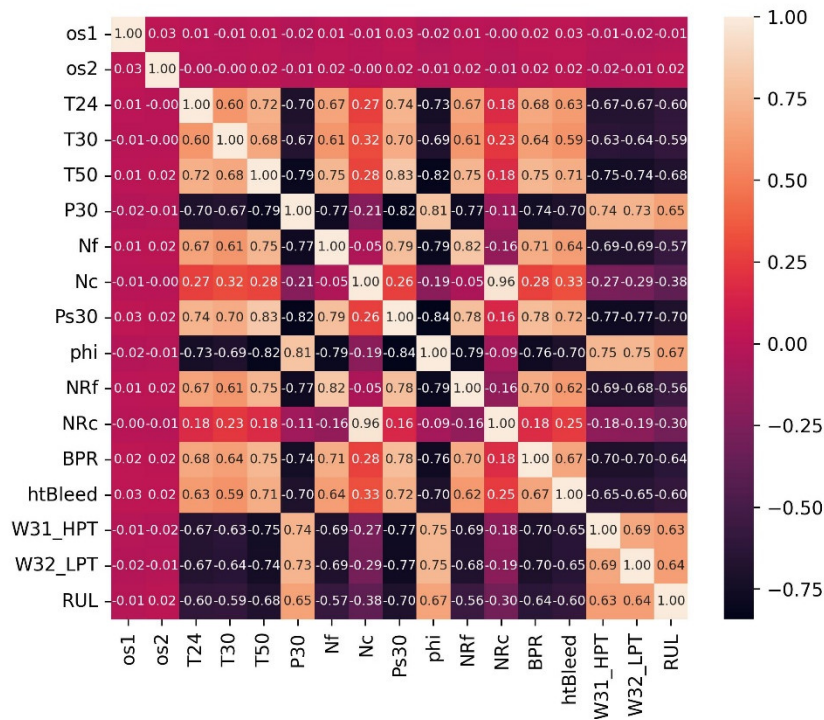
3.1.1. Tiriamoji duomenų analizė

Siekiant išsiaiškinti esamus sąryšius, tiriamiems kintamiesiems braižomi taškiniai grafikai ir skaičiuojamos koreliacijos koeficiento reikšmės su priklausomu kintamuoju RUL. Sklaidos grafikai tarp kiekvieno kintamojo ir RUL matomi 11 pav., galima pastebėti, kad dauguma kintamųjų turi gana aiškius sąryšius su RUL (mažiau sąryšių turi *os1* ir *os2* kintamieji), vieni iš jų koreliuoja teigiamai, kiti neigiamai.



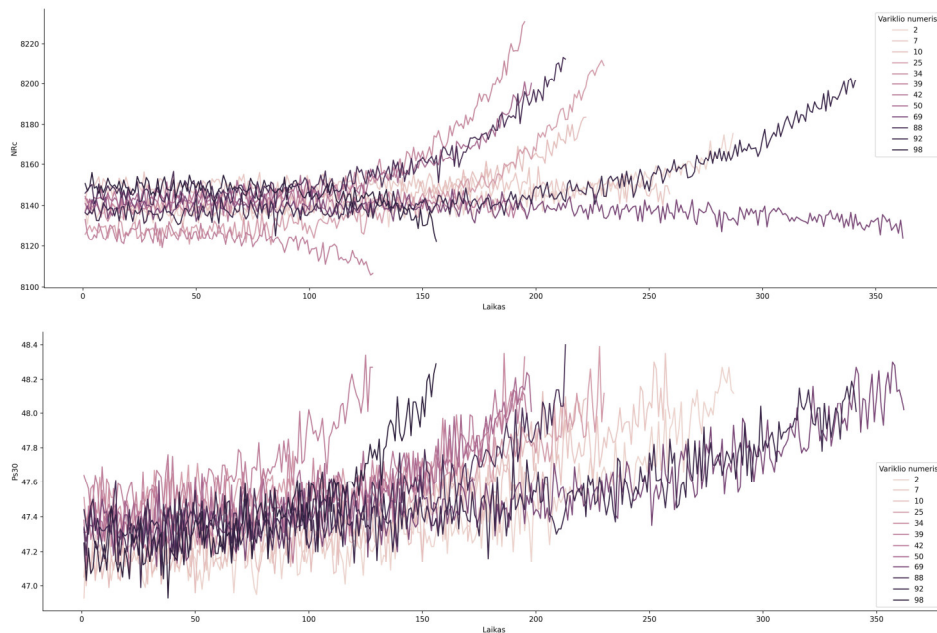
11 pav. Sklaidos grafikai priklausomiems kintamiesiems ir RUL

Koreliacijų matrica (žr. 12 pav.) rodo, kad *os1* ir *os2* kintamieji mažiausiai koreliuoja, nemaža dalis turi gana stiprią teigiamą (iki 0,67) arba neigiamą (iki -0,70) koreliaciją su RUL.



12 pav. Koreliacijų matricos grafikas pradiniam duomenims

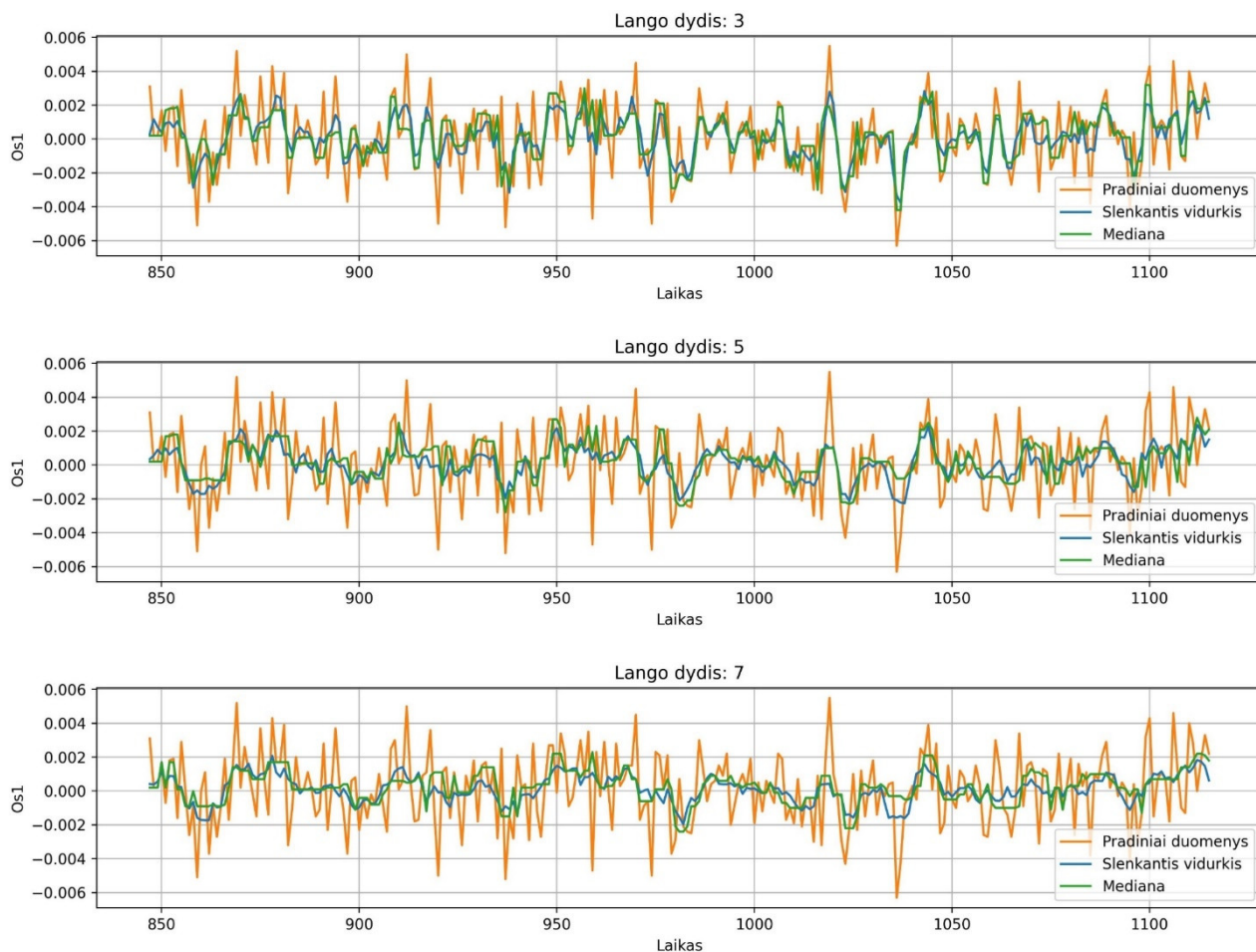
Duomenyse aprašyta variklių veikimo informacija pradeda normaliai veikiančiu varikliu, baigiant gedimo atsiradimu. Duomenų kintamieji bėgant laikui ima keistis – galima stebėti (žr. 13 pav.), kad atsirandant gedimui *NRC* kintamojo reikšmės skirtingiems varikliams ima didėti arba mažėti, o *Ps30* kintamojo reikšmės visiems varikliams gendant ima didėti.



13 pav. Pradinių duomenų vizualizacijos skirtingiems varikliams, *NRC* ir *Ps30* kintamiesiems

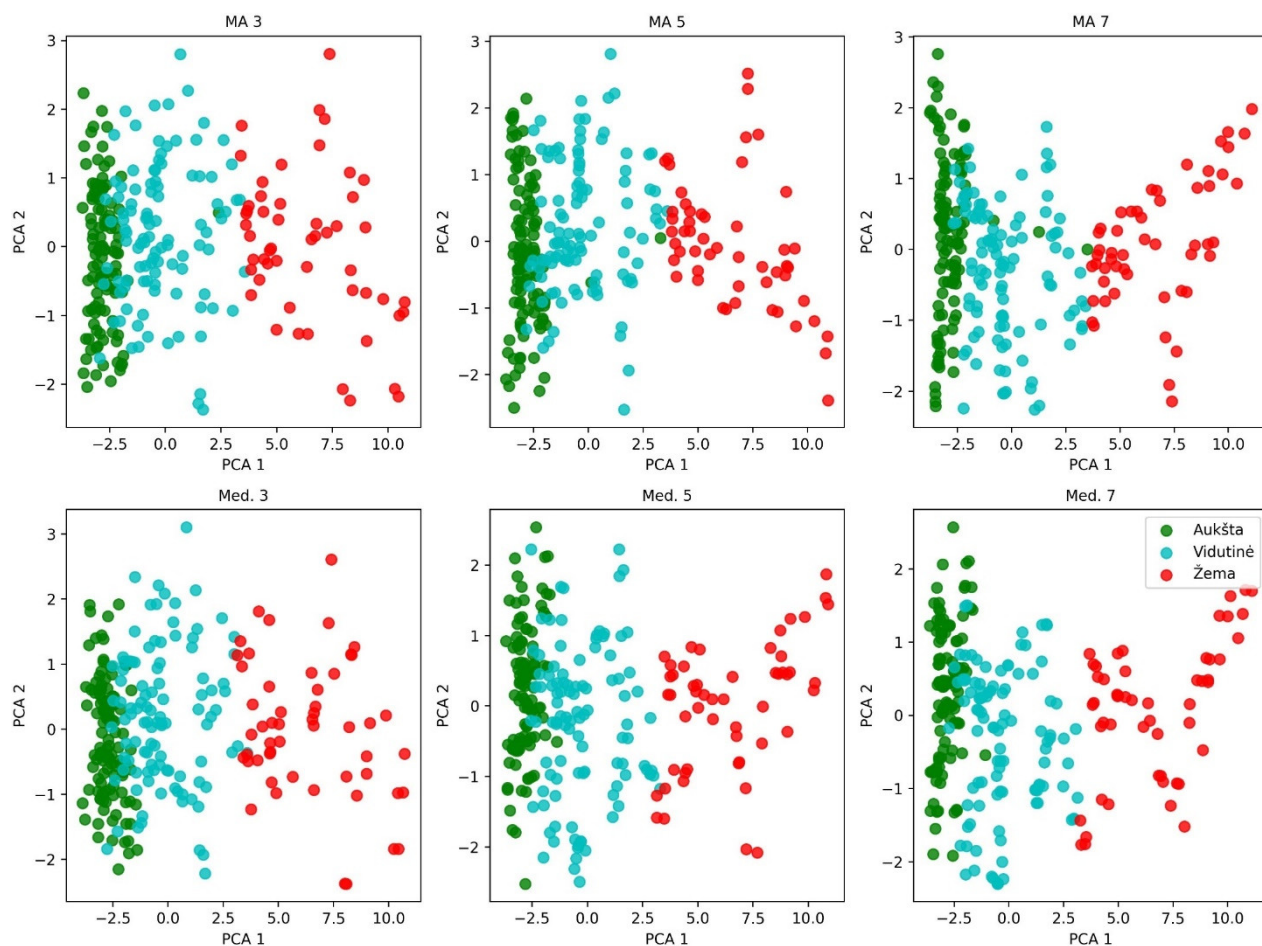
3.1.2. Triukšmo filtravimas

Triukšmo šalinimui išbandyti slenkančio vidurkio ir medianos filtrai, su slenkančio lango dydžiais 3, 5, 7. Rezultatai vizualizuoti *osI* kintamajam, 5 variklio duomenims: 14 pav. matomi pradiniai duomenys bei filtruoti abiem filtrais su skirtingais parametrais. Galima matyti, kad duomenys tampa glotnesni tiek su vienu, tiek su kitu filtru.



14 pav. Triukšmo filtravimas skirtingiems slenkančio lango dydžiams su medianos ir slenkančio vidurkio filtrais, *osI* kintamajam

Su šiais parametrais filtruotiems duomenims atliktos PCA vizualizacijos (žr. 15 pav.), spalvos atitinka RUL kategorijas. Slenkančio vidurkio (MA) atveju, galima matyti keletą išskirčių: daugiausiai jų MA 7 atveju, t.y. naudojant lango dydį 7, mažiausiai lango dydžio 3 (MA 3) atveju. Medianos filtro atveju išskirčių yra mažiau, skirtingų lango reikšmių atvejai nelabai vizualiai skiriasi.

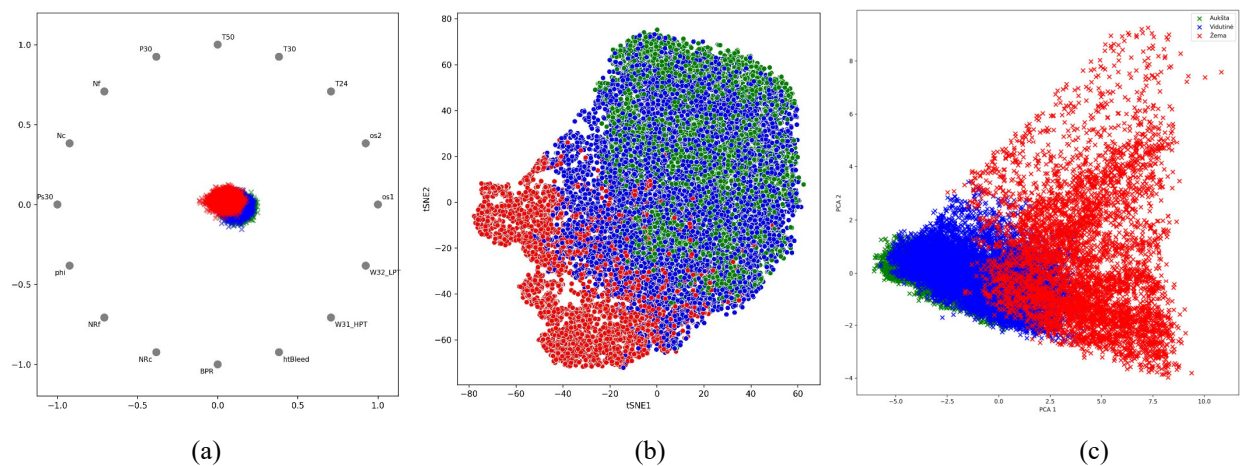


15 pav. PCA metodas su dviem komponentėmis filtruotiems duomenims su 3, 5 ir 7 slenkančio lango dydžiais su slenkančio vidurkio (MA) ir medianos (Med.) filtrais

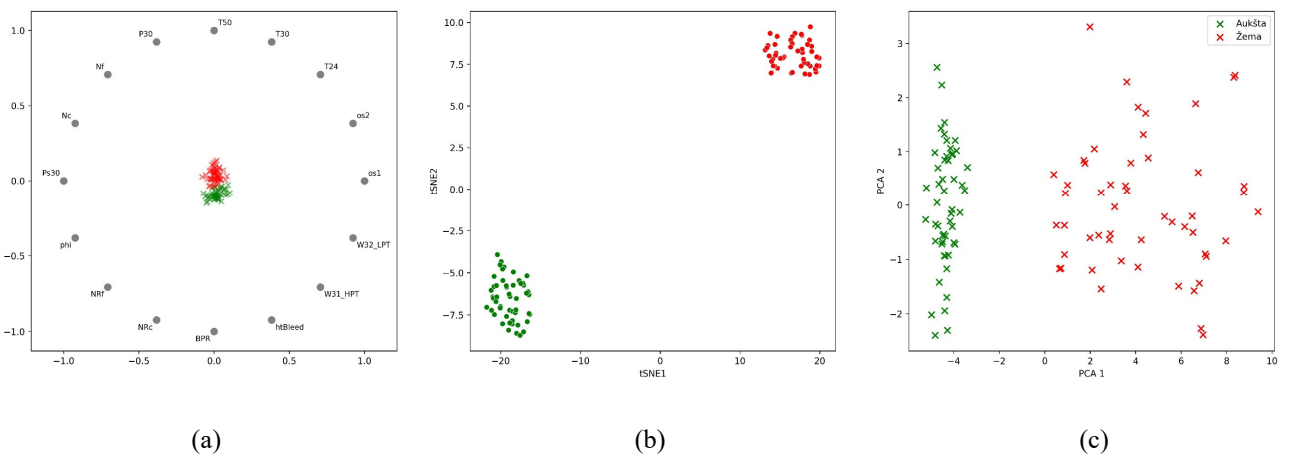
Tolimesniems tyrimams pasirinkti duomenys, filtruoti medianos filtru, su lango dydžiu 3.

3.1.3. Pradinių duomenų vizualizacijos dimensijų mažinimo metodais

Normuotiems standartiniu būdu (*StandardScaler*) ir pagal ankstesniame skyrelyje aprašytą parinktą triukšmo mažinimo metodą filtruotiems duomenims braižomos vizualizacijos, duomenis projektuojant į dvimatę erdvę. 16 pav. matomi visų variklių duomenys kartu, vizualizuoti trimis skirtingais metodais (*RadViz*, t-SNE, PCA). Galima pastebėti, kad *RadViz* metodas RUL išskiria gana prastai, didelė dalis taškų tarp „Aukštos“ ir „Žemos“ RUL kategorijų yra persidengę. t-SNE žali bei raudoni taškai yra labiau atskirti. PCA šiuo atveju sukuria geriausią vizualizaciją, nes „Žemos“ ir „Aukštos“ kategorijų RUL taškai atskirti ryškiausiai.



16 pav. Pradinių duomenų vizualizacijos visiems 100 variklių, visų kategorijų („Aukšta“, „Vidutinė“, „Žema“) RUL, metodai: (a) – *RadViz*, (b) – t-SNE, (c) – PCA



17 pav. Pradinių duomenų vizualizacijos 15 variklio duomenims, „Aukštos“ ir „Žemos“ RUL kategorijoms, metodai: (a) – *RadViz*, (b) – t-SNE, (c) – PCA

17 pav. matomos tais pačiais metodais sukurtos vizualizacijos, bet vaizduojama tik vieno variklio duomenys, piešiant tik „Aukštos“ ir „Žemos“ RUL kategorijų taškus. Šiuo atveju galima stebėti, kad visi metodai taškus atskiria gana neblogai, nors *RadViz* atveju skirtingos kategorijos vaizduojamos labai arti viena kitos. Geriausių vizualizaciją sukuria t-SNE, kur skirtingos RUL grupės vaizduojamos priešinguose grafiko kampuose.

3.2. Atraktoriaus rekonstrukcija fazinėje erdvėje ir jos vizualizavimas

Atraktoriaus rekonstrukcija fazinėje plokštumoje gali padėti sukurti informatyvesnes vizualizacijas. Toliau aprašomas rekonstrukcijos procesas: vėlinimų vektoriaus optimizavimas, skirtingų normavimo būdų įtakos rezultatams palyginimas. Išbandomas atraktorių kūrimas pavieniams kintamiesiems, siekiant išsiaiškinti kintamųjų koreliacijos su RUL įtaką vizualizacijoms. Kuriamas atraktorius iš daugiamatės duomenų aibės, parenkamas optimalus dimensijų skaičius, atliekami tikslumo vertinimai skaitmeniniams vaizdams.

3.2.1. Vėlinimų vektoriaus optimizavimo genetiniu algoritmu rezultatai

Vėlinimų dimensijos parenkamos eksperimentiniu būdu, išbandant atraktorius su 3, 16, 33 dimensijomis. Kadangi formuojant atraktorių, pirmoji reikšmė yra be vėlinimų, tai kuriamo vektoriaus ilgis turi būti vienetu mažesnis, nei rekonstruotų dimensijų skaičius. Parinktomis dimensijoms siekiama surasti optimalų vėlinimų vektorių, tam naudojamas genetinis algoritmas.

Optimizavimui svarbu apibrėžti tikslo funkciją. Buvo išbandyta tikslo funkcija skaičiuojant kiekvieno taško Euklido atstumą iki centro. Tačiau paskui įvesta modifikacija – kvadratas pakeistas moduliu. Taip padaryta dėl esamų duomenų specifikos: reikšmėms kintant intervale (0,1) kvadratinės funkcijos skaičiavimas reikšmę sumažina, o turint didesnes nei 1 reikšmes – padidina. Siekiant vienodai traktuoti visas reikšmes, kaip atstumas iki centro skaičiuojamas modulis (žr. 12 formulę metodinėje dalyje).

Optimizavimo parametrai ir jų reikšmės visoms trimis dimensijoms pateikiami žemiau, 3 lentelėje. Optimizavimui naudoti 51 variklio, „Aukštos“ RUL kategorijos, t.y. gerai veikiančio variklio, duomenys.

3 lentelė. Genetinio algoritmo optimizavimo parametrai ir jų reikšmės; čia n – vėlinimų vektoriaus ilgis, įgyja reikšmes: 2, 15, 32.

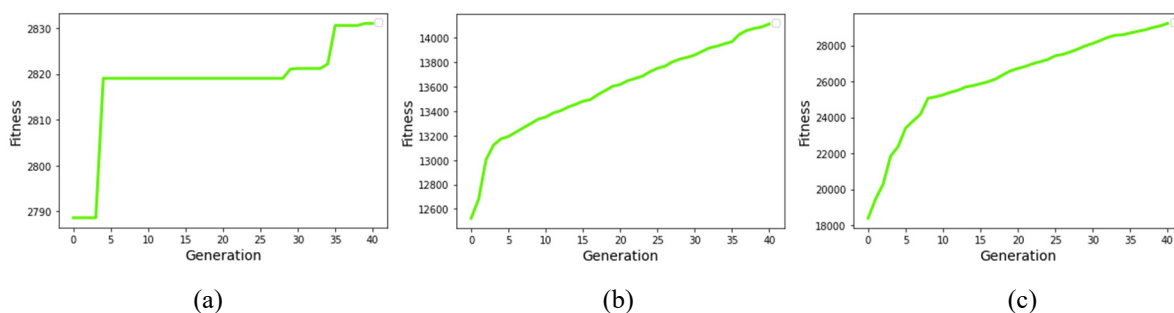
Parametro pavadinimas	Parametro reikšmė
<i>num_generations</i>	40
<i>num_parents_mating</i>	4 (kai $n = 2;15$), 10 (kai $n = 32$)
<i>sol_per_pop</i>	$2*n$
<i>num_genes</i>	n
<i>gene_type</i>	int
<i>init_range_low</i>	1
<i>init_range_high</i>	32
<i>parent_selection_type</i>	"sss"
<i>keep_parents</i>	1
<i>crossover_type</i>	"single_point"
<i>mutation_type</i>	"random"
<i>mutation_percent_genes</i>	10

Optimizavimas atliekamas naudojant standartiniu *StandardScaler* ir nestandartiniu *MinMaxRUL* normuotus duomenis. 4 lentelėje matomi optimizavimo rezultatai su *StandardScaler* apdorotiems duomenims. Lentelėje parinkta dimensija D_{prad} apibrėžia kuriamo vėlinimų vektoriaus ilgį n , kuris yra vienetu mažesnis už D_{prad} . Sukurtas vėlinimų vektorius – tai genetinio algoritmo optimizavimo rezultatas, pasiekęs lentelėje nurodytą tikslo funkcijos reikšmę. Pastebėtina, kad sukurtuose vektoriuose, ypač esant didesniai dimensijų skaičiui ($D_{prad} = 16; 33$), optimaliuose sprendiniuose atsiranda nemažai 0 reikšmių. Tokie vėlinimai tik pakartotų prieš tai buvusią reikšmę, nepridėdami naujos informacijos, tad yra pašalinami. Galutinis vėlinimų vektorius V sukuriamas išbraukus 0 reikšmes, o galutinė kuriamo atraktoriaus dimensija apskaičiuojama prie V vėlinimų vektoriaus ilgio pridėdant 1, kuris žymi pirmąją nevēlintą koordinatę. Šiuo atveju, pradiniam dimensijų skaičiui esant 16 ir 33, galutinės dimensijos atraktoriui atitinkamai lygios 8 ir 12.

4 lentelė. Genetinio algoritmo optimizavimo rezultatai, naudojant standartiniu būdu normuotus duomenis

Parinkta dimensija D_{prad}	Vėlinimų vektoriaus ilgis n ; $n = D_{prad} - 1$	Sukurtas vėlinimų vektorius	Tikslo funkcijos reikšmė	Galutinis vėlinimų vektorius V	Galutinė atraktoriaus dimensija D ; $D = V \text{ ilgis} + 1$
3	2	[4, 7]	2831,0371694562637	[4, 7]	3
16	15	[0, 6, 0, 9, 0, 0, 6, 6, 11, 0, 19, 0, 0, 17, 0]	14113,954088371818	[6, 9, 6, 6, 11, 19, 17]	8
33	32	[0, 9, 2, 10, 0, 0, 4, 3, 0, 0, 11, 1, 0, 0, 2, 0, 0, 0, 0, 0, 18, 1, 0, 10, 0, 0, 0, 0, 0, 0, 0]	29220,664871992536	[9, 2, 10, 4, 3, 11, 1, 2, 18, 1, 10]	12

18 pav. matomas tikslo funkcijos kitimo procesas, kurio metu buvo gauti 4 lentelėje aprašyti rezultatai. Galima stebėti, kad 3 dimensijų (a) grafike optimizavimo procesas pabaigoje stabilizuojasi, likusiuose (b) ir (c) stebimas funkcijos kilimas. Didesnė funkcijos reikšmė rodo geresnį sprendinio tinkamumą.



18 pav. Tikslo funkcijos kitimo grafikas optimizavimo proceso metu, naudojant standartiškai normuotus duomenis: (a) – $n = 2$, (b) – $n = 15$, (c) – $n = 32$, čia: x ašis – „*Generation*“ – žymi iteracijų skaičių (skirtingas populiacijos kartas), y ašis – „*Fitness*“ – žymi tikslo funkcijos reikšmę

Verta pastebėti, kad tinkamumas tokiu būdu gali būti lyginamas tik konkrečios dimensijos aplinkoje, tikslo funkcijos reikšmės nėra lyginamos tarp skirtingų dimensijų.

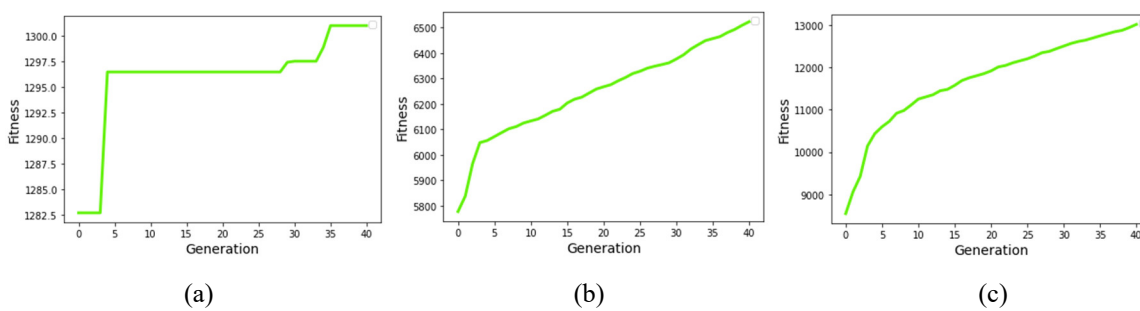
5 lentelė. Genetinio algoritmo optimizavimo rezultatai, naudojant nestandartiniu būdu normuotus duomenis

Parinkta dimensija D_{prad}	Vėlinimų vektoriaus ilgis n ; $n = D_{prad} - 1$	Sukurtas vėlinimų vektorius	Tikslo funkcijos reikšmė	Galutinis vėlinimų vektorius V	Galutinė atraktoriaus dimensija D ; $D = V \text{ ilgis} + 1$
3	2	[6, 6]	1301,0400423875153	[6, 6]	3
16	15	[0, 7, 0, 4, 0, 0, 1, 6, 15, 0, 19, 0, 0, 17, 0]	6522,452829911603	[7, 4, 1, 6, 15, 19, 17]	8

33	32	[0, 9, 2, 18, 9, 0, 3, 2, 0, 0, 3, 3, 1, 11, 5, 0, 0, 0, 0, 0, 16, 0, 0, 8, 2, 0, 0, 4, 0, 2, 0, 0]	13019,470157745884	[9, 2, 18, 9, 3, 2, 3, 3, 1, 11, 5, 16, 8, 2, 4, 2]	17
----	----	--	--------------------	--	----

5 lentelėje matomi optimizavimo duomenys nestandartiniu *MinMaxRUL* būdu normuotiems duomenims. Optimizavimo procesas atitinka aukščiau aprašytą procesą su standartiniu būdu normuotais duomenimis, tik šiuo atveju naudojami su rankiniu būdu sukurta funkcija normuoti duomenys. Galima pastebėti, kad šiuo atveju pradinės atraktoriaus dimensijos buvo nurodytos tokios pačios (3, 16, 33), tačiau galutinės atraktoriaus dimensijos, kurios gaunamos išbraukiant nulines reikšmes, yra 3, 8 ir 17.

Tikslo funkcijų kitimas optimizavimo proceso metu stebimas 19 pav. Galima matyti, kad procesas vykta panašiai kaip ir su *StandartScaler* apdorotų duomenų atveju, skiriasi galutinės pasiektos tikslo funkcijos reikšmės.



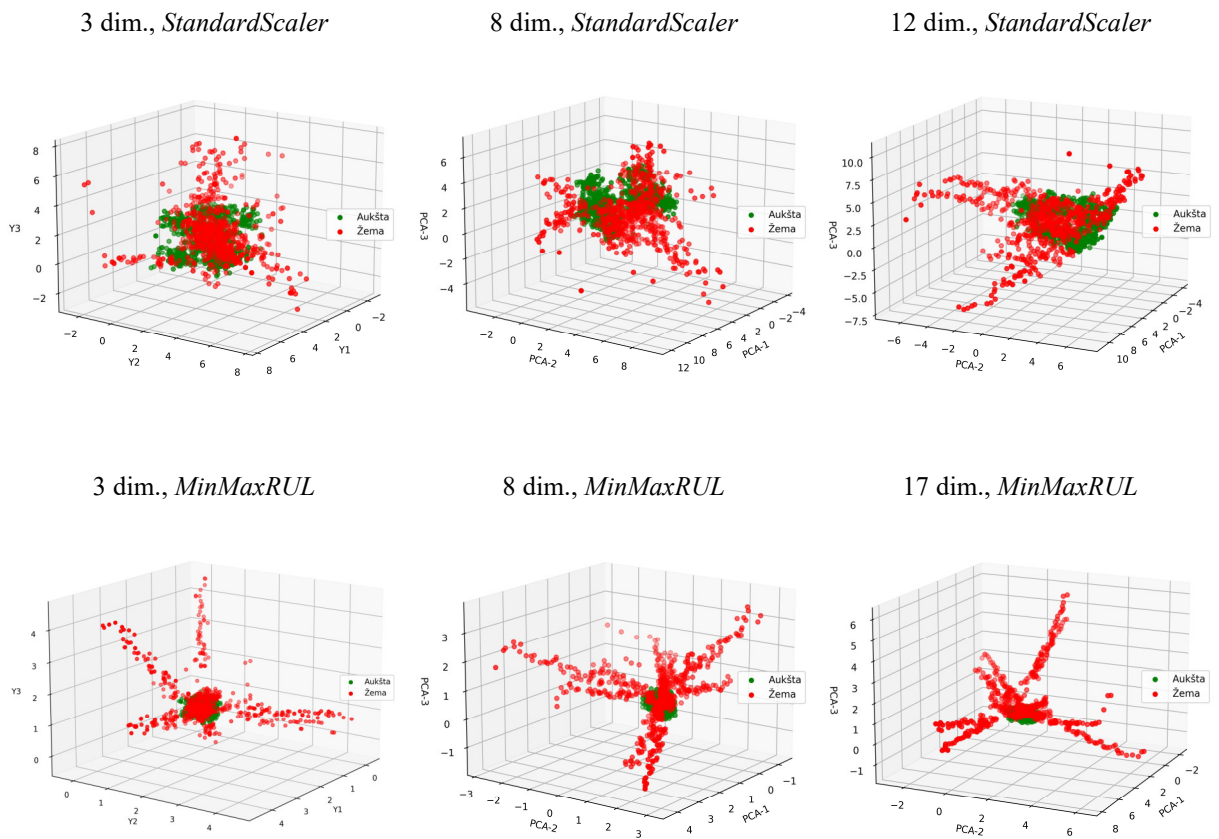
19 pav. Tikslo funkcijos kitimo grafikas optimizavimo proceso metu, naudojant nestandartinio *MinMax* normavimo duomenis: (a) – $n = 2$, (b) – $n = 15$, (c) – $n = 32$, čia: x ašis – „*Generation*“ – žymi iteracijų skaičių (skirtingas populiacijos kartas), y ašis – „*Fitness*“ – žymi tikslo funkcijos reikšmę

Toliau darbe bus naudojami optimizuoti vėlinimų vektoriai (nurodomi anksčiau aptartose 4 lentelėje ir 5 lentelėje pagal normavimo tipą ir galutinę atraktoriaus dimensijų skaičių).

3.2.2. Normavimo įtakos atraktoriaus vizualizacijoms palyginimas

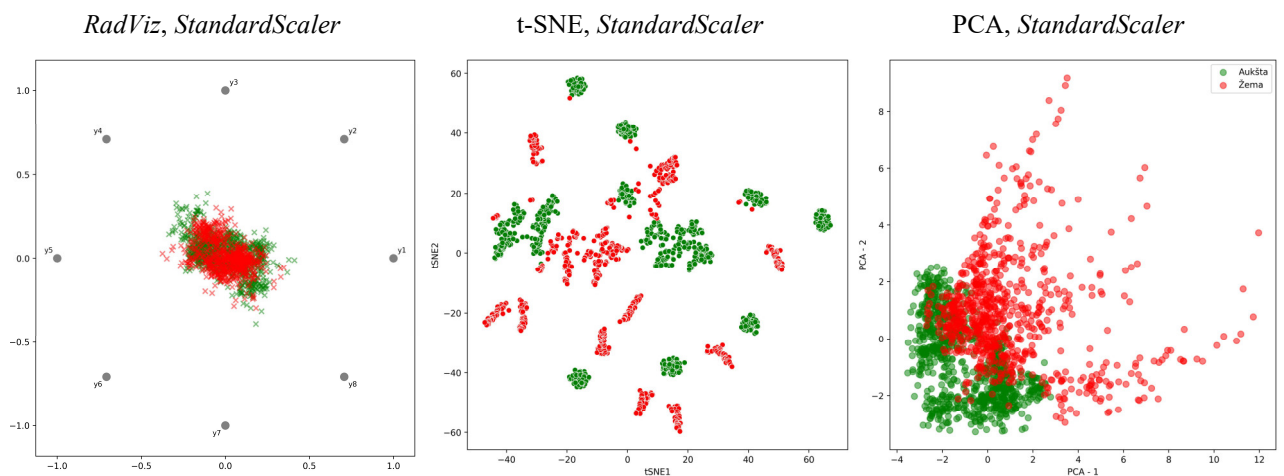
Optimizavimo procesas buvo atliktas dviem skirtingais būdais normuotiems (*StandardScaler*, *MinMaxRUL*) duomenų rinkiniams, kiekvienam sukurta po tris vėlinimo vektorius, rekonstruojančius atraktorių 3, 8 ir 12 dimensijų erdvėje *StandardScaler* normuotų duomenų atveju, bei 3, 8 ir 17 dimensijų erdvėje *MinMaxRUL* normuotų duomenų atveju.

Siekiant išsiaiškinti tinkamesnę atraktoriaus kūrimui normavimą, palyginimas pateikiamas trimačiais taškiniais grafikai (žr. 20 pav.), kur 3 dimensijų atraktorius vizualizuojamas tiesiogiai trimačiame grafike, didesnių dimensijų atvejais naudojama PCA su trimis komponentėmis. Galima matyti, kad *MinMaxRUL* normuotiems duomenims sukurti atraktoriai yra išraiškingesni, labiau išsiskiria gerai veikiančio (žali taškai) ir gendančio (raudoni taškai) variklių duomenys – žali taškai turi daug didesnę tankį nei raudoni. *StandardScaler* atveju žalių taškų sklaida yra didesnė, o raudonų – mažesnė nei *MinMaxRUL* atveju.

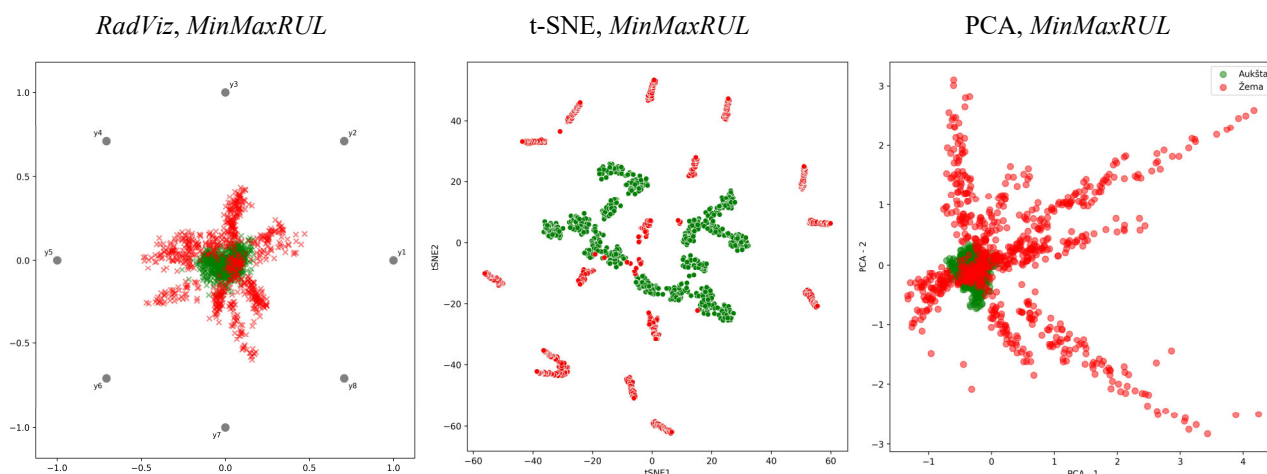


20 pav. Atraktoriaus vizualizacijos skirtingam dimensijų (paveiksle: dim.) skaičiui, 51 varikliui, naudojant standartinę *StandardScaler* ir nestandartinę *MinMaxRUL* normavimo metodus; vėlinimų vektoriai pagal dimensijas iš 4 lentelės bei 5 lentelės

Žemiau matomos vizualizacijos (žr. 21 pav. ir 22 pav.) sukurtiems 8 dimensijų atraktoriaus, naudojant dvimates vizualizacijas. *MinMaxRUL* normuoti duomenys vizualizacijose atrodo geriau: *RadViz* bei PCA metodais „Aukštos“ RUL taškai sugrupuojami kartu, o „Žemos“ RUL turi didelę sklaidą, t-SNE žali taškai taip pat yra labiau sugrupuoti kartu nei *StandardScaler* atveju.



21 pav. *RadViz*, t-SNE, PCA vizualizacijos naudojant standartinę *StandardScaler* normavimo metoda; kuriamas 8 dimensijų atraktoriaus 51 varikliui ($n = 15$, optimalus vėlinimų vektorius [6, 9, 6, 6, 11, 19, 17])

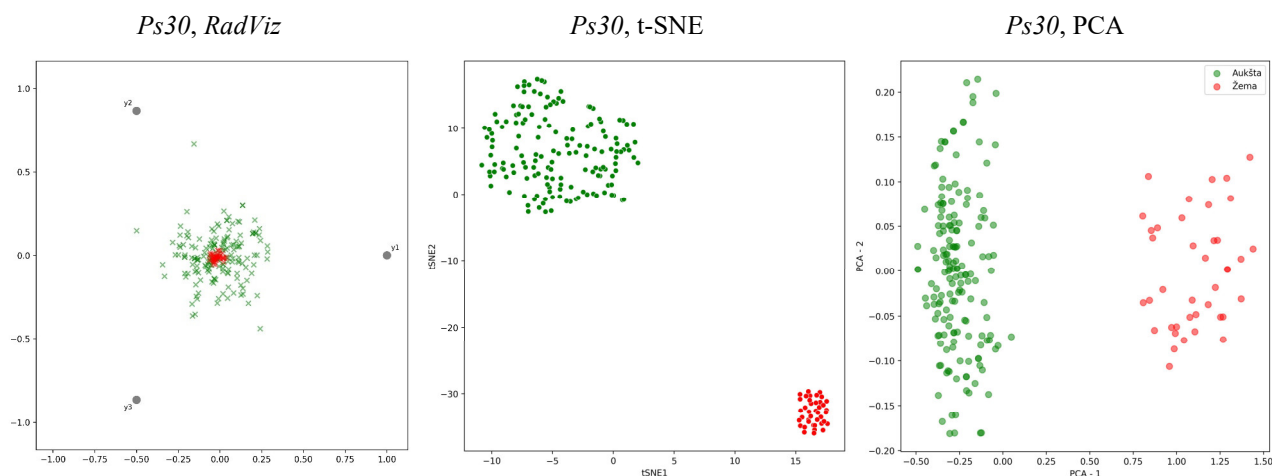


22 pav. *RadViz*, t-SNE, PCA vizualizacijos naudojant nestandartinį *MinMaxRUL* normavimo metodą; kuriamas 8 dimensijų atraktoriaus 51 varikliui ($n = 15$, optimalus vėlinimų vektorius [7, 4, 1, 6, 15, 19, 17])

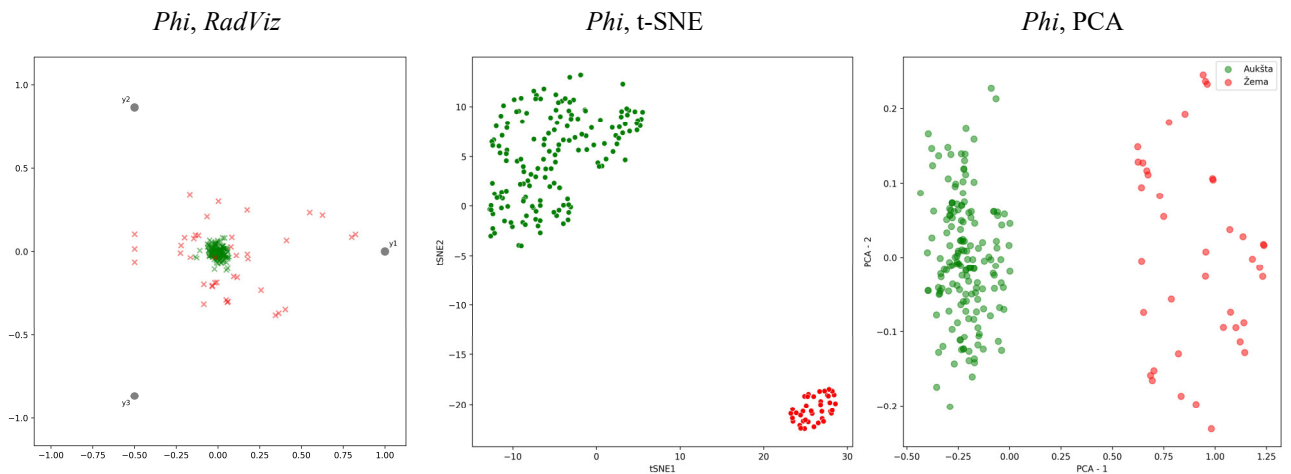
Atsižvelgiant į gautas vizualizacijas, toliau kuriant atraktoriaus bus dirbama su *MinMaxRUL* būdu normuotais duomenimis.

3.2.3. Kintamųjų koreliacijos su RUL įtaka atraktoriaus vizualizacijoms

Kintamųjų sąryšis su RUL yra skirtingas, vieni koreliuoja teigiamai, kiti neigiamai. *Ps30* kintamasis pasižymi didžiausia neigiama koreliacija (-0,69), o *phi* – didžiausia teigiama koreliacija (0,67). Šiame skyrelyje siekiama iširti, kokią įtaką atraktoriaus vizualizacijai turi kintamasis, priklausomai nuo jo koreliacijos su RUL. 23 pav. ir 24 pav. matomos atraktorių, kurtų naudojant tik po vieną iš šių kintamųjų, vizualizacijos. t-SNE ir PCA metodai sukuria tokį patį vaizdą, nepriklausant nuo koreliacijos ženklo, tačiau *RadViz* atveju sudarytas grafikas tampa priešingas, priklausant nuo kintamojo koreliacijos su RUL: *phi* kintamojo atveju žalia spalva (gerai veikiantis variklis, aukštas RUL) vaizduojama centre, o gendančio variklio (raudona spalva, žemas RUL) taškai išsisklaidę aplink, *Ps30* kintamasis, kurio sąryšis su RUL yra priešingas, *RadViz* gerai veikiančio variklio stebėjimų taškus išsklaido, o centre atvaizduojami sugedusio variklio duomenys.

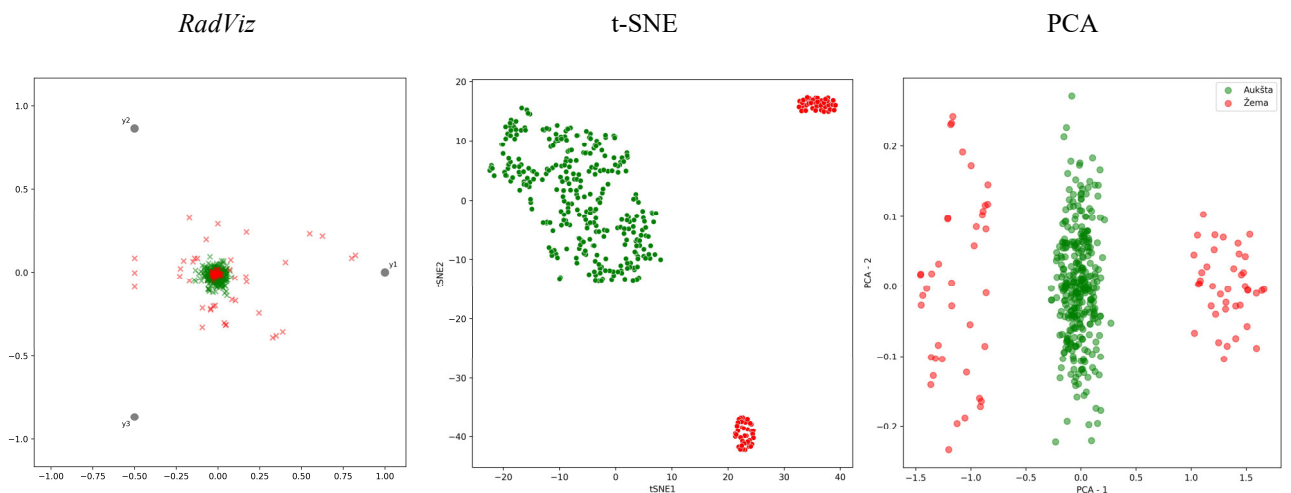


23 pav. Atraktoriaus (3 dimensijų) tik su vienu kintamuoju (*Ps30*) vizualizacijos *RadViz*, t-SNE, PCA metodais; 96 variklis



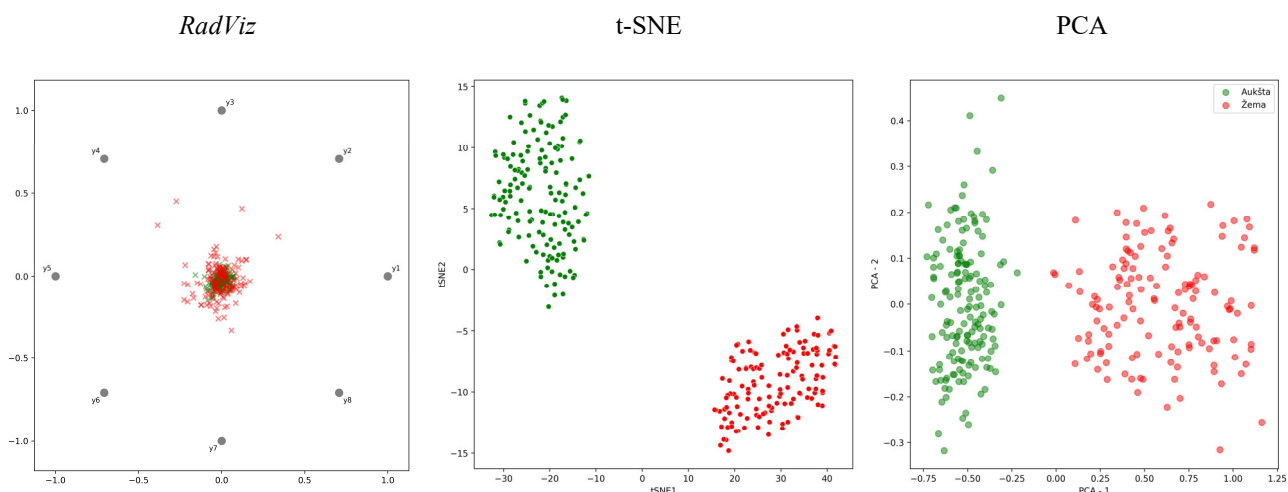
24 pav. Atraktoriaus (3 dimensijų) tik su vienu kintamuoju (*Phi*) vizualizacijos *RadViz*, t-SNE, PCA metodais; 96 variklis

Anksčiau aprašyti atraktoriai su *Ps30* ir *Phi* kintamaisiais buvo vizualizuoti viename grafike kartu, rezultatai matomi 25 pav.: *RadViz* atveju aiškiai matosi priešingas kintamųjų vaizdavimas, priklausantis nuo koreliacijos. t-SNE bei PCA metodų elgsena panaši – gerai veikiančio variklio taškai atvaizduojami kartu, nepriklausomai kuriam kintamajam priklauso, o gendančio variklio taškai išsklaidomi į dvi skirtingas grupes, ką greičiausiai lemia skirtingi kintamųjų pokyčiai RUL mažėjimo atžvilgiu.



25 pav. Atraktoriaus (3 dimensijų) tik su dviem kintamaisiais (*Ps30* ir *Phi*) vizualizacijos *RadViz*, t-SNE, PCA metodais; 96 variklis

26 pav. stebimas atraktoriaus, kurtas parinkus vien teigiamai su RUL koreliuojančius kintamuosius (*Phi*, *P30*, *W31_HPT*, *W32_LPT*). Šiuo atveju *RadViz* labiau išklaido raudonus taškus, nors yra ir keletas labiau išsklaidytų žalių taškų. t-SNE ir PCA metodai visus taškus sugrupuoja į dvi grupes pagal RUL kategorijas.

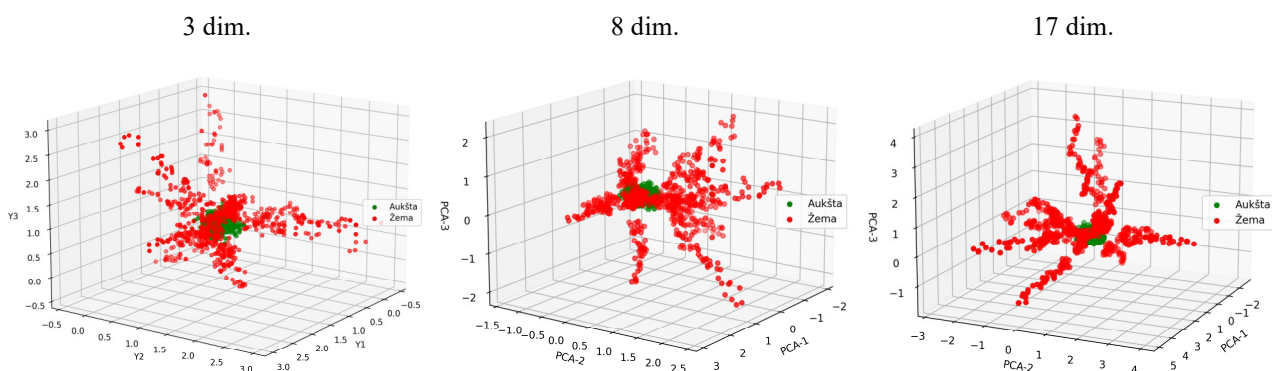


26 pav. Atraktorius (8 dimensijų), kurto tik su teigiamai koreliuojančiais su RUL kintamaisiais (Φ , $P30$, $W31_HPT$, $W32_LPT$) vizualizacijos *RadViz*, t-SNE, PCA metodais; 51 variklis

Taigi, t-SNE ir PCA metodų atveju, gerai veikiančio variklio taškai visada grupuojami kartu, o gendančio išsklaidomi, esant skirtingai koreliuojantiems kintamiesiems. *RadViz* atvejis kitoks – taškai išsklaidomi arba sugrūdami centre, priklausant nuo jų koreliacijos su RUL.

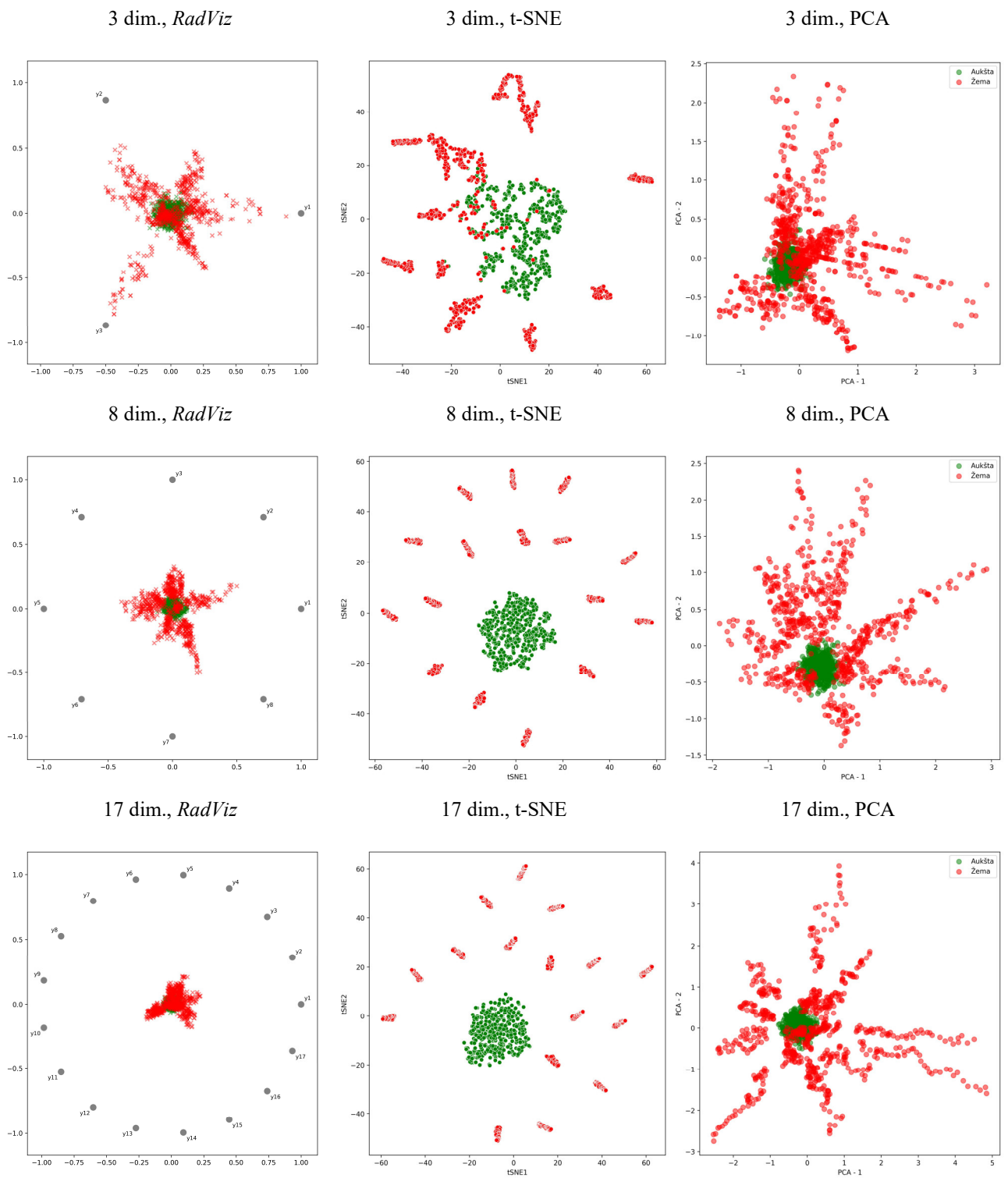
3.2.4. Dimensijų tyrimas ir parinkimas

Vėlinimų vektoriai buvo optimizuoti trims skirtingoms dimensijoms (3, 16, 33), o išbraukus 0 reikšmes gautos galutinės dimensijos 3, 8 ir 17 (*MinMaxRUL* normavimas). Šiame skyrelyje tiriama, kurios dimensijos vizualizacijos yra informatyviausios. 29 pav. matomi trimačiai atraktorių grafikai, sukurti visoms trims rekonstrukcijoms. Galima pastebėti, kad 17 dimensijų atraktorių vizualiai labiausiai išskiria RUL kategorijas – žali taškai yra viename didelio tankio klasteryje, o raudoni vizualiai turi didžiausią sklaidą.



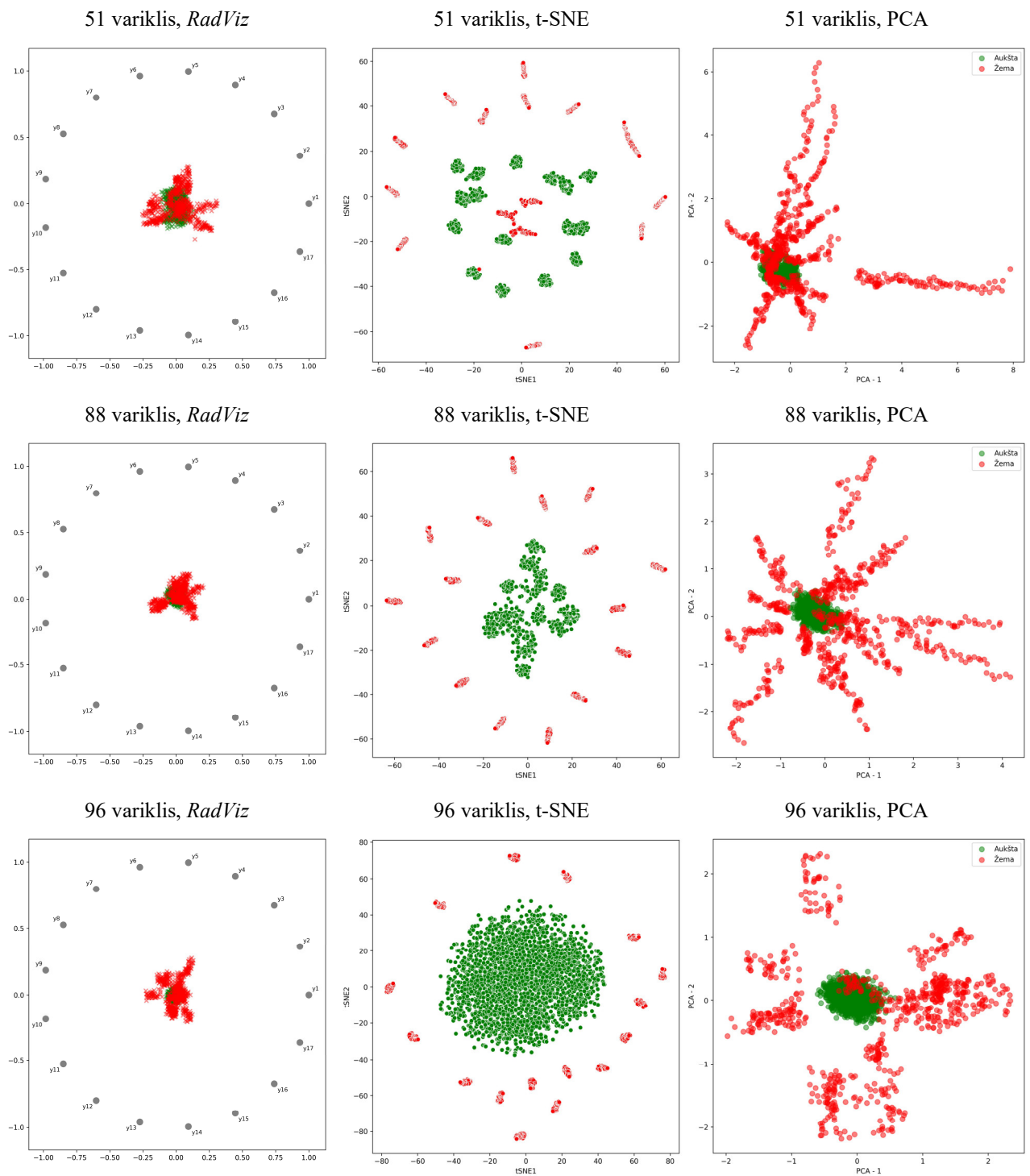
27 pav. Atraktoriaus vizualizacijos 34 varikliui, 3 dimensijų (paveiksle: dim.) atveju – tiesiogiai, 8 ir 17 – trijų komponentių PCA

Vizualizuoti dvimačiai grafikai *RadViz*, t-SNE ir PCA metodais matomi 28 pav. *RadViz* atveju vizualiai RUL kategorijos labiau atskiriamos 3 ir 8 dimensijų atvejais, nes žali taškai grupuojami centre, raudoni – išsklaidomi. Tačiau t-SNE metodas geresnes vizualizacijas sukuria 8 ir 17 dimensijų atveju, nes žali taškai sugrupuojami kartu. PCA metodas geriausią vizualizaciją sukuria 17 dimensijų atveju, nes žali taškai yra grupuojami į didelio tankio grupę, o raudoni labiausiai išsklaidomi.



28 pav. Atraktoriaus dimensijų (paveiksle: dim.) skaičiaus tyrimas (34 variklis, *MinMaxRUL*)

Kaip optimalus dimensijų skaičius pasirinktas 17; atrinkta pagal vizualiai matomą išskirtą formą.

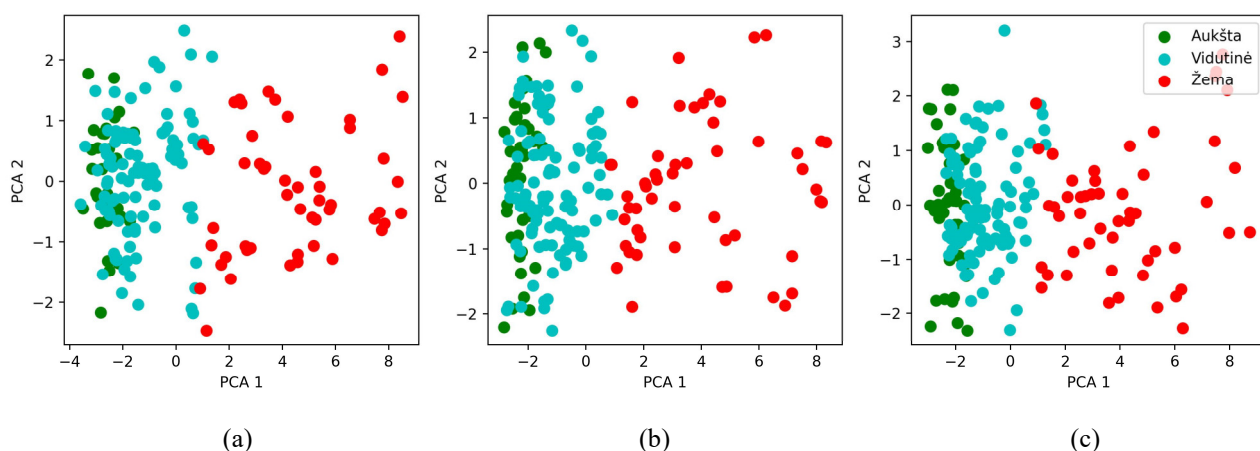


29 pav. Atraktoriaus vizualizacijų palyginimas skirtingiems varikliams (17 dimensijų, *MinMaxRUL*, vėlinimų vektorius: [9, 2, 18, 9, 3, 2, 3, 3, 1, 11, 5, 16, 8, 2, 4, 2])

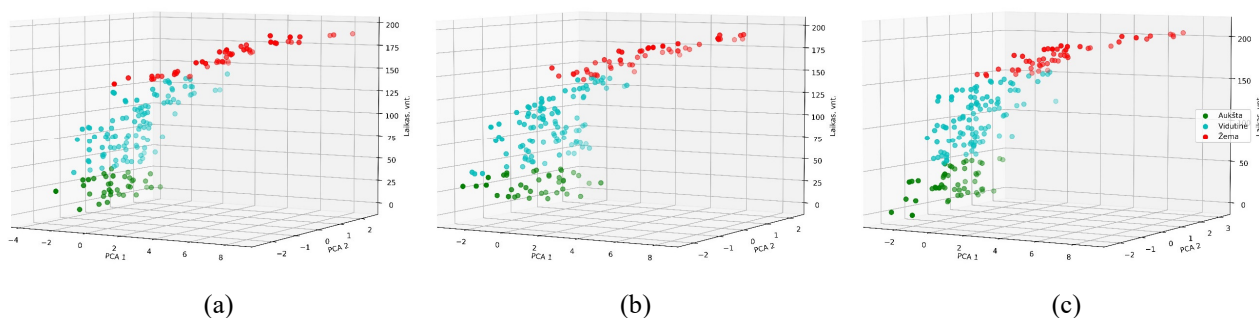
Toliau atliktas su šiuo dimensijų skaičiumi kurtų rekonstrukcijų skirtingiems varikliams palyginimas (žr. 29 pav.); galima matyti, kad *RadViz* atveju didelių skirtumų nėra, PCA – šiek tiek skiriasi RUL sklaidos dydis, t-SNE – šiek tiek keičiasi žalių taškų sugrupavimo kartu intensyvumas.

3.3. Principinių komponentių analizė su laiko dimensija pradiniais duomenimis

Pradiniams duomenims (šiuo atveju taikytas standartinis *StandardScaler* normavimas) pritaikytos dvikomponentės PCA vizualizacijos matomos 30 pav. Galima stebėti, kad skirtingos RUL kategorijos persidengia.



30 pav. Dvimatės RUL kategorijų vizualizacijos su PCA metodu, 1 (a), 34 (b) ir 71 (c) variklių duomenimis



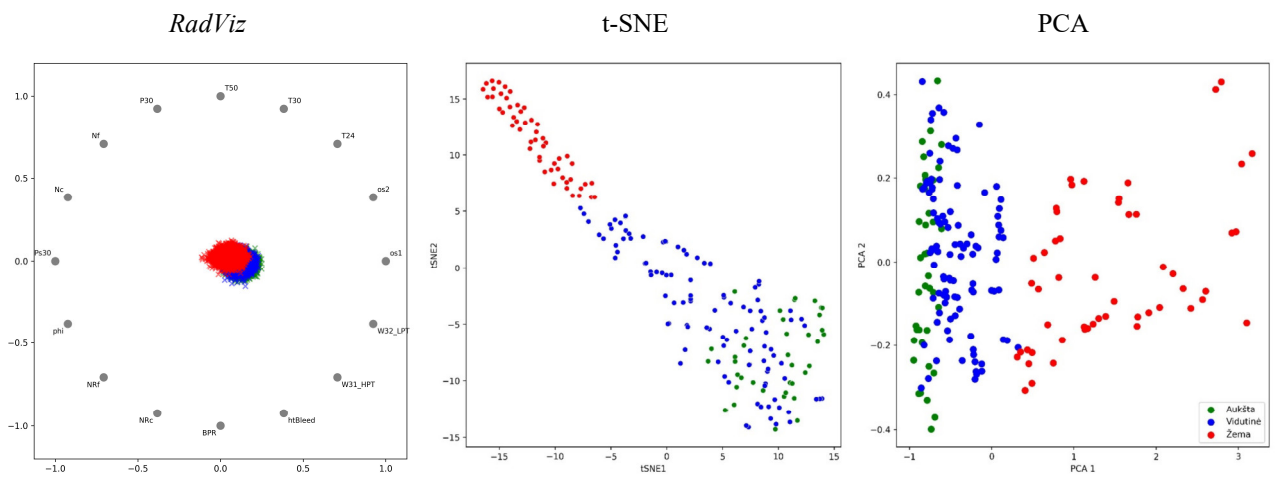
31 pav. Trimatės RUL kategorijų vizualizacijos, 1 (a), 34 (b) ir 71 (c) variklių duomenimis, naudojant dviejų komponentių PCA ir laiko dimensiją bei *StandardScaler* normavimą

31 pav. matomas vizualizacijų plėtinys trimatėje erdvėje, kaip trečioji dimensija atvaizduojamas laikas. Šiuo atveju skirtingos RUL kategorijos nepersidengia, gerai atskiriamos į skirtingas grupes.

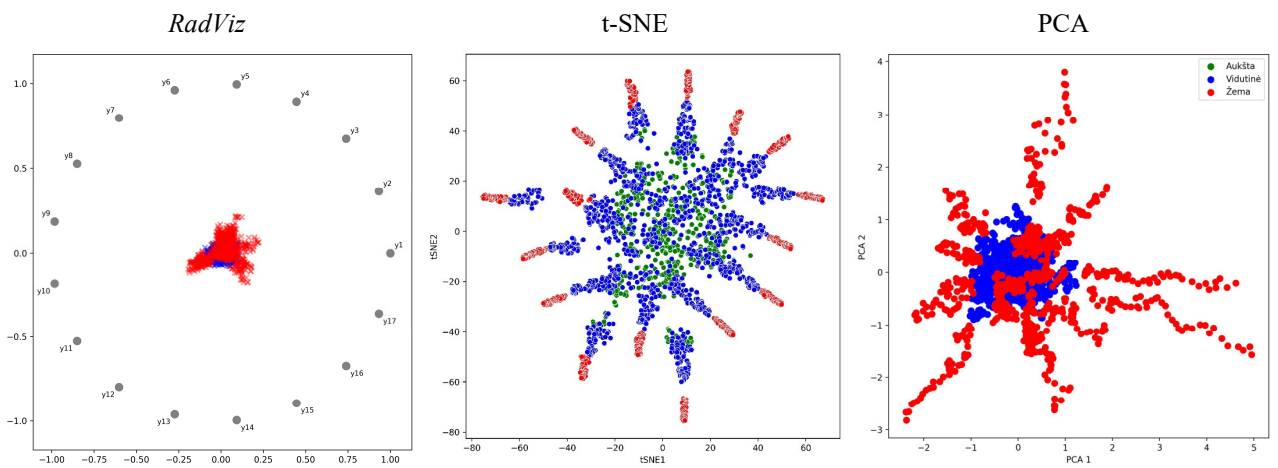
3.4. Vizualizacijų pokyčio laike skaitinis įvertinimas

Siekiant įvertinti vizualizacijų pasikeitimą laike skaitiniu būdu, slenkančio lango principu buvo sukurti paveikslai, fiksuojantys tam tikrą intervalą taškų. Skaičiuojami vidutiniai kvadratiniai skirtumai tarp gretimų vaizdų ir brėžiamas grafikas, siekiant nustatyti staigesnio kitimo laiko momentą, kuris galimai atitiktų gedimo atsiradimą.

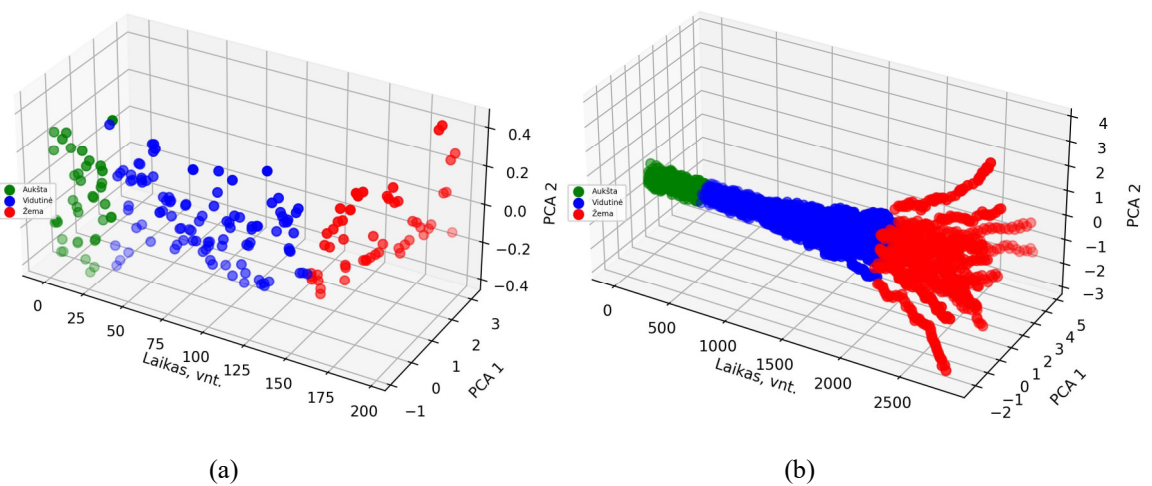
32 pav. galima matyti vizualizacijas, kurtas pradiniais 34 variklio duomenimis, o 33 pav. – to paties variklio duomenų vizualizacijas rekonstruotam atraktoriui; naudojamas parinktas optimalus dimensijų skaičius 17, vėlinimų vektorius: [9, 2, 18, 9, 3, 2, 3, 3, 1, 11, 5, 16, 8, 2, 4, 2]. Vaizduojamas visas signalas, apimantis tris RUL kategorijas: „Aukšta“ – gerai veikiančio variklio, „Vidutinė“ – tarpinis etapas atsirandant gedimui, „Žema“ – atsirandantis gedimas. Galima pastebėti, kad pradinėse vizualizacijos duomenys pereina iš vienos kategorijos į kitą tolygiai tiesine trajektorija, o atraktorius sukuria spindulinį vaizdavimą, žali taškai yra brėžiami didelio tankio klasteryje centre, mėlyni – kiek su didesne sklaida, o raudonų skaida pati didžiausia.



32 pav. Pradinių viso signalo duomenų („Aukštos“, „Vidutinės“ ir „Žemos“ RUL), *RadViz*, t-SNE, PCA grafikai; 34 variklis



33 pav. Atraktoriaus (17 dimensijų), kurto iš viso signalo duomenų („Aukštos“, „Vidutinės“ ir „Žemos“ RUL), *RadViz*, t-SNE, PCA grafikai; 34 variklis

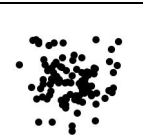

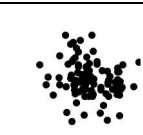


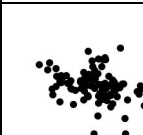
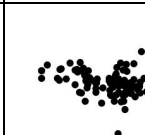



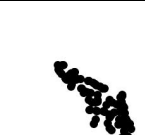
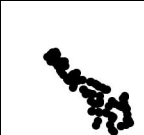
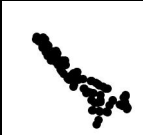
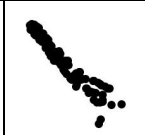


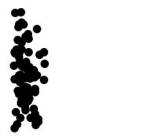
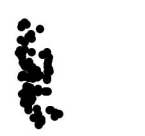





34 pav. Pradinė erdvė (a) ir 17 dimensijų atraktoriaus (b) trimačiai grafikai, vizualizuojantys dviejų komponentių PCA ir laiko dimensiją; 34 variklis

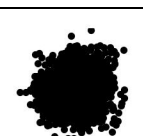
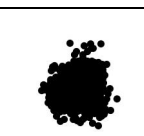
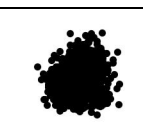
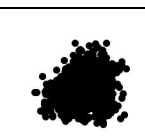
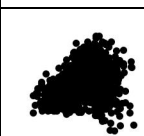
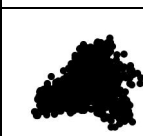
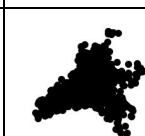














34 pav. (a) matoma pradinių duomenų dvikomponentė PCA su laiko dimensija, o (b) – rekonstruotoje erdvėje kurtas grafikas. Pradinės erdvės taškų trimatė vizualizacija nėra tokia informatyvi, matosi,

kad laikui bėgant taškai keičia pozicijas, tačiau nėra ryškaus sklaidos padidėjimo, iš kurio būtų galima pastebėti gedimą. Atraktoriaus atveju galima matyti, kaip bėgant laikui keičiasi RUL kategorijos, o taškų sklaida vis didėja. Toliau buvo vertinama vaizdų evoliucija. Vizualizacijų pokyčiai laike fiksuojami su slenkančiu langu, kurio dydis lygus pusei turimų stebėjimų taškų ilgio. Slenkantis langas perkeliamas 1 dydžio žingsniu, kiekvieną kartą fiksuojamas vaizdas. Tokia vaizdų evoliucija laike užfiksuojama visais trimis tirtais metodais (*RadViz*, t-SNE, PCA) tiek pradinėje, tiek fazinėje plokštumoje gautoms vizualizacijomis. 6 lentelėje matomi vaizdai, fiksuoti pradinėje erdvėje. Sukurti 96 vaizdai kiekvienam metodui, lentelėje matomi laiko momentais i fiksuoti vaizdai. Galima matyti, kad visų metodų atveju įvyksta tam tikras pokytis, geriausiai tai matosi t-SNE bei PCA vaizduose, *RadViz*, vizualiai vertinant, keičiasi mažiau.

6 lentelė. Vaizdų evoliucija duomenims pradinėje erdvėje *RadViz*, t-SNE ir PCA metodais, kintant laikui; čia i – laiko indeksas, kurio metu fiksuotas vaizdas (0 – eksperimento pradžia, 96 – eksperimento pabaiga)

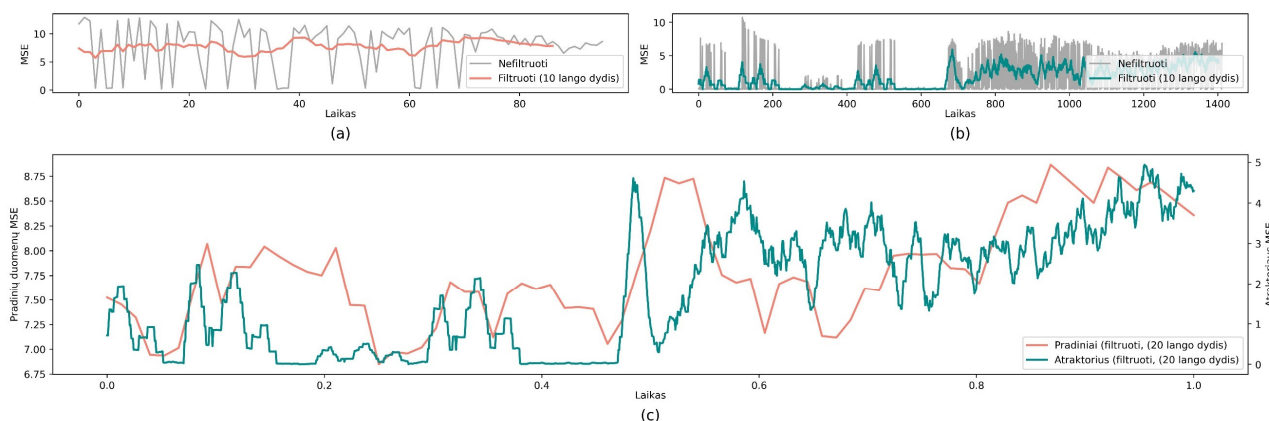
Metodas	Laikas							
	$i = 0$	$i = 16$	$i = 32$	$i = 48$	$i = 64$	$i = 80$	$i = 96$	
<i>RadViz</i>								
t-SNE								
PCA								

7 lentelė. Vaizdų evoliucija fazinėje erdvėje rekonstruotam atraktoriui, vizualizuotam *RadViz*, t-SNE ir PCA metodais, kintant laikui; čia i – laiko indeksas, kurio metu fiksuotas vaizdas (0 – eksperimento pradžia, 1412 – eksperimento pabaiga)

Metodas	Laikas							
	$i = 0$	$i = 235$	$i = 470$	$i = 705$	$i = 940$	$i = 1175$	$i = 1412$	
<i>RadViz</i>								
t-SNE								
PCA								

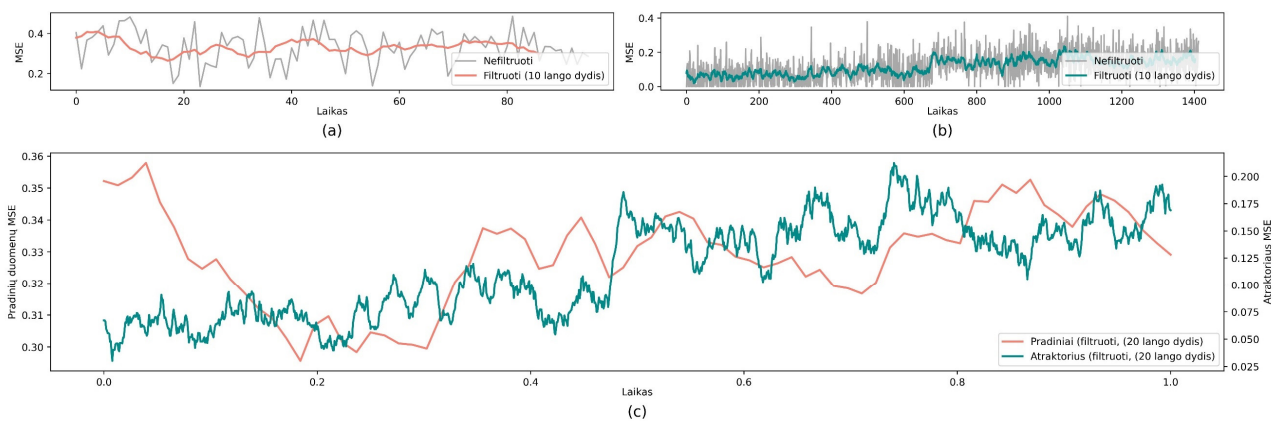
7 lentelėje matoma vaizdų evoliucija atraktoriaus atveju, viso sukurta 1412 vaizdų kiekvienam metodui, lentelėje pavaizduotas tam tikras poaibis. Galima pastebėti, kad *RadViz* evoliucija šiuo atveju labiau išreikšta, nei pradinėje erdvėje. t-SNE ir PCA atveju stebimas sklaidos didėjimas iš centro į kraštus, o pradinėje erdvėje sklaidos didėjimas buvo stebimas labiau kaip poslinkis į šoną.

Toliau užfiksuotiems vaizdams, nagrinėjant po 2 gretimus vaizdus, buvo skaičiuojama MSE paklaida. Vertinama, kokio dydžio pokytis įvyko tarp gretimų fiksuotų vaizdų ir stebima, vaizdai keitėsi tolygiai ar buvo netolygumų. 35 pav. matomi apskaičiuoti MSE rezultatai *RadViz* metodui: (a) – pradinėje erdvėje, o (b) – atraktoriaus. MSE reikšmės gana triukšmingos, todėl taikomas filtravimas su slenkančio vidurkio filtru, parenkant 10 dydžio langą (a) ir (b) grafikuose, o palyginimo tarp pradinės ir atraktoriaus erdvių vizualizacijų naudojamas labiau suvidurkintas (20 dydžio langas) signalas. Galima pastebėti, kad MSE kreivė atraktoriaus erdvėje iš pradžių yra gana tolygi, o tuomet (ties eksperimento viduriu – 0,5) staigiai šokteli (galimai dėl atsirandančio gedimo) ir tuomet iš lėto ima kilti į viršų. Pradinėje erdvėje MSE kinta kitaip, nėra tokios aiškios tendencijos. Ties eksperimento viduriu taip pat stebimas staigus šoktelėjimas į viršų, tačiau pastebėtina, kad atraktoriui fiksuota MSE reaguoja kiek anksčiau nei pradinių duomenų MSE.



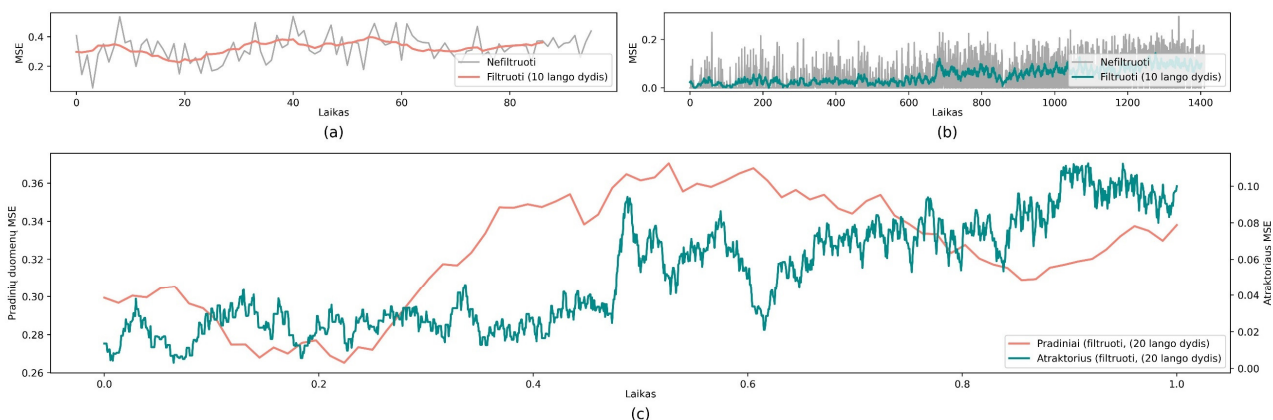
35 pav. Pradinių duomenų ir rekonstruoto 17 dimensijų atraktoriaus *RadViz* vizualizacijų MSE kitimo palyginimas; (a) – originali ir filtruota MSE pradinių vaizdų vizualizacijoms, (b) – originali ir filtruota MSE atraktoriaus vizualizacijoms, (c) – pradinės erdvės ir atraktoriaus vizualizacijų MSE palyginimas

36 pav. matomos t-SNE metodo MSE vizualizacijos, atliktos tokiu pačiu principu kaip ir prieš tai aprašytos *RadViz* MSE. Stebint pradinių duomenų vaizdų MSE kitimus, nesimato jokios tendencijos, kreivė tai kyla, tai leidžiasi. Atraktoriaus atveju, stebimas staigus MSE padidėjimas maždaug eksperimento viduryje – galimai tai žymi gedimo atsiradimą. Taip pat kreivė yra šiek tiek kylanti į viršų.



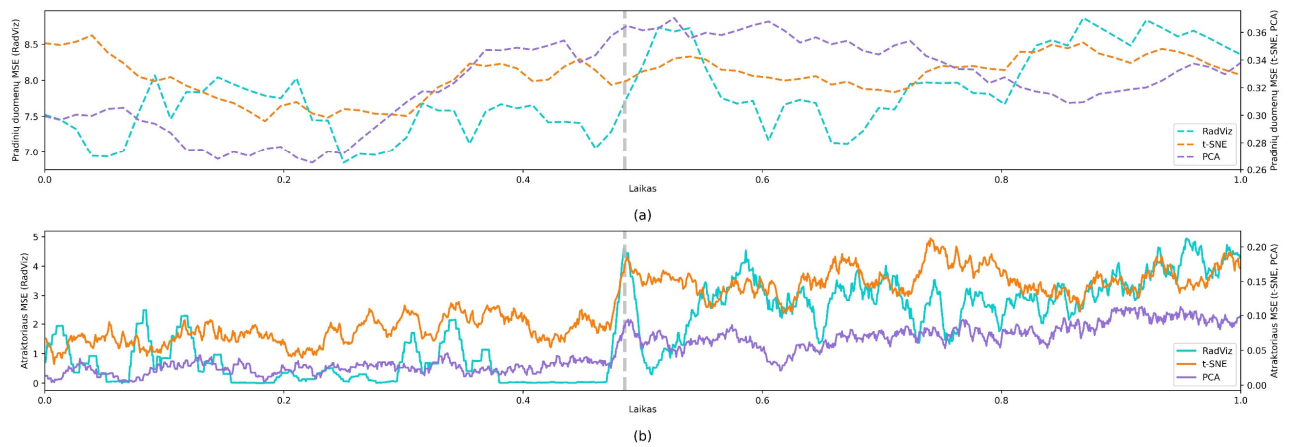
36 pav. Pradinių duomenų ir rekonstruoto 17 dimensijų atraktoriaus t-SNE vizualizacijų MSE kitimo palyginimas; (a) – originali ir filtruota MSE pradinių vaizdų vizualizacijoms, (b) – originali ir filtruota MSE atraktoriaus vizualizacijoms, (c) – pradinės erdvės ir atraktoriaus vizualizacijų MSE palyginimas

PCA metodo rezultatų vertinimas matomas 37 pav., galima matyti, kad pradinių duomenų MSE kreivė tai kyla, tai leidžiasi; tai reikštų, jog esant kilimui didėja skirtumai tarp gretimų vaizdų, o paskui jie ima mažėti. Atraktoriaus atveju kreivė yra pradžioje gana pastovi, o paskui ima kilti. Stebimas staigus šuolis eksperimento viduryje.



37 pav. Pradinių duomenų ir rekonstruoto 17 dimensijų atraktoriaus PCA vizualizacijų MSE kitimo palyginimas; (a) – originali ir filtruota MSE pradinių vaizdų vizualizacijoms, (b) – originali ir filtruota MSE atraktoriaus vizualizacijoms, (c) – pradinės erdvės ir atraktoriaus vizualizacijų MSE palyginimas

Išnagrinėjus MSE kreives kiekvienam metodui atskirai, toliau pateikiamas grafikas (žr. 38 pav.), leidžiantis palyginti skirtingų metodų kreives: (a) grafike matomi pradinių duomenų vaizdų kitimų MSE su *RadViz*, t-SNE ir PCA, o (b) – atraktoriaus vizualizacijų *RadViz*, t-SNE, PCA metodais MSE. Galima matyti, kad pradinių duomenų MSE kinta gana atsitiktinai, tik *RadViz* atveju stebimas staigus šuolis eksperimento viduryje.



38 pav. Pradinių duomenų (a) ir rekonstruoto 17 dimensijų atraktoriaus (b) MSE pokytis visiems metodams kartu

Atraktoriaus erdvėje (žr. 38 pav., b), visų trijų metodų MSE kreivės elgiasi panašiai – pradžioje yra gana tolygios, stebimas staigus šuolis viduryje (grafike žymimas pilka punktyrine linija), antroje pusėje stebimas lengvas kilimas į viršų. Pilka punktyrinė linija žymi atraktoriaus kreivių šoktelėjimą, toje pačioje vietoje ji nubrėžta ir (a) grafike. Galima matyti, kad, nors pradinių duomenų erdvėje *RadViz* ir sureaguoja į pokytį, ši reakcija yra vėlyvesnė nei atraktoriaus erdvėje. Lyginant atraktoriaus MSE kreivių reakciją, galima matyti, kad stipriausiai į galimai atsiradusį gedimą reaguoja *RadViz* metodas.

Apibendrinant vizualizacijų evoliucijos laike tyrimą, galima teigti, kad atsiradę staigūs MSE kreivių šoktelėjimai eksperimento viduryje galimai nusako atsirandantį gedimą. Pradinių duomenų erdvėje geriausiai į pokyčius reaguoja *RadViz* metodas. Vis dėlto rekonstruoto atraktoriaus vizualizacijų MSE kreivės turi aiškesnes tendencijas ir ryškiau reaguoja į pokyčius, nei pradinėje erdvėje. Lyginant atraktoriaus MSE, visos kreivės viduryje eksperimento atvaizduoja didelį pokytį, labiausiai jis išreikštas *RadViz* atveju.

Išvados

1. Atlikus daugiamačių duomenų vizualizacijos metodų apžvalgą, nustatyta, kad šių metodų įvairovė didelė, jie gali būti grupuojami pagal veikimo pobūdį bei taikomi atsižvelgiant į turimus duomenis ir juose esančius sąryšius. Apžvelgti mechaninių sistemų diagnostikoje naudojami modeliai, kuriais stengiamasi aptikti gedimus ar prognozuoti likusią naudingą veikimo trukmę. Pastebėta, kad vizualizacijos metodų taikymo atveju šioje srityje nėra daug, o vienas dažniau sutinkamų metodų – t-SNE.
2. Naudojant daugiamačių lėktuvų variklių duomenų rinkinį, ištirti *RadViz*, t-SNE ir PCA metodai siekiant vizualizuoti likusią naudingą veikimo trukmę. Nustatyta, kad šiose vizualizacijose taškai tarp skirtingų RUL kategorijų kinta tiesine trajektorija. Vizualiai vertinant, „Žemos“ RUL kategorijos taškai geriau išsiskiria naudojant t-SNE ir PCA metodus, nes mažiau persidengia su kitų kategorijų taškais nei *RadViz* atveju.
3. Sudarytas fazinės erdvės rekonstravimo paremtas vizualizavimo metodas, atliekant dimensijų ir vėlinimų optimizavimą genetiniu algoritmu, sukuriant rankinę duomenų normavimo funkciją. Parinktas optimalus rekonstravimo dimensijų skaičius (17) ir vėlinimų vektorius: [9, 2, 18, 9, 3, 2, 3, 3, 1, 11, 5, 16, 8, 2, 4, 2]. Sukurtas atraktorius vizualizuotas *RadViz*, t-SNE ir PCA metodais.
4. Siekiant ištirti vaizdų evoliuciją, dvimatės projekcijos praplėstos su laiko dimensija. Laiko ašimi slenkančio lango principu sukurti vaizdai bei apskaičiuoti skirtumai tarp jų, brėžiant MSE grafikus:
 - Palyginus rezultatus, galima pastebėti, kad pradinių duomenų vizualizacijų kitimo laike MSE kreivėse reakcija į galimai atsirandantį gedimą matoma tik *RadViz* atveju, kitų metodų kreivės kinta be ryškių tendencijų.
 - Atraktoriaus vizualizacijų kitimo kreivės turi ryškesnes tendencijas: pradžioje yra gana stabilios, viduryje signalo stebimas staigus pokytis – padidėjimas, galimai išreiškiantis besivystantį sistemos gedimą, paskui MSE kreivėje stebima kilimo tendencija. Jautriausiai į gedimą reaguoja *RadViz* metodas.
 - *RadViz* metodo reakcija į gedimą rekonstruotoje erdvėje pastebima anksčiau, nei pradinėje erdvėje.

Literatūros sąrašas

1. ETEMADPOUR, R., et al. Choosing Visualization Techniques for Multidimensional Data Projection Tasks: A Guideline with Examples. *Braz, J., Et Al. Computer Vision, Imaging and Computer Graphics Theory and Applications. VISIGRAPP 2015. Communications in Computer and Information Science, Springer, Cham, 2016, vol. 598 DOI 10.1007/978-3-319-29971-6_9.*
2. VENTOCILLA, E. ir RIVEIRO, M. A Comparative User Study of Visualization Techniques for Cluster Analysis of Multidimensional Data Sets. *Information Visualization, 2020, vol. 19, no. 4. pp. 318-338 ISSN 1473-8716. DOI 10.1177/1473871620922166.*
3. BERTINI, E., TATU, A. ir KEIM, D. Quality Metrics in High-Dimensional Data Visualization: An Overview and Systematization. *IEEE Transactions on Visualization and Computer Graphics, 2011, vol. 17, no. 12. pp. 2203-2212 ISSN 2160-9306. DOI 10.1109/TVCG.2011.229.*
4. ITOH, T., NAKABAYASHI, A. ir HAGITA, M. Multidimensional Data Visualization Applying a Variety-Oriented Scatterplot Selection Technique. *Journal of Visualization, 2023, vol. 26, no. 1. pp. 199-210 ISSN 1875-8975. DOI 10.1007/s12650-022-00871-6.*
5. VAN DER MAATEN, L., POSTMA, E.O. ir VAN DEN HERIK, H.J. Dimensionality Reduction: A Comparative Review. *Journal of Machine Learning Research, 2009, vol. 10, no. 66-71. pp. 13. Prieiga per: https://www.researchgate.net/profile/Eric-Postma/publication/228657549_Dimensionality_Reduction_A_Comparative_Review/links/0046353a3047fc2863000000/Dimensionality-Reduction-A-Comparative-Review.pdf.*
6. SACHA, D., et al. Visual Interaction with Dimensionality Reduction: A Structured Literature Analysis. *IEEE Transactions on Visualization and Computer Graphics, 2017, vol. 23, no. 1. pp. 241-250 ISSN 2160-9306. DOI 10.1109/TVCG.2016.2598495.*
7. ABDI, H. ir WILLIAMS, L.J. Principal Component Analysis. *Wiley Interdisciplinary Reviews: Computational Statistics, 2010, vol. 2, no. 4. pp. 433-459 DOI <https://doi.org/10.1002/wics.101>.*
8. NONATO, L.G. ir AUPETIT, M. Multidimensional Projection for Visual Analytics: Linking Techniques with Distortions, Tasks, and Layout Enrichment. *IEEE Transactions on Visualization and Computer Graphics, 2019, vol. 25, no. 8. pp. 2650-2673 ISSN 2160-9306. DOI 10.1109/TVCG.2018.2846735.*
9. VAN LONG, T. ir LINSEN, L. Visualizing High Density Clusters in Multidimensional Data using Optimized Star Coordinates. *Computational Statistics, 2011, vol. 26, no. 4. pp. 655-678 ISSN 1613-9658. DOI 10.1007/s00180-011-0271-3.*
10. KANDOGAN, E. Star Coordinates: A Multi-Dimensional Visualization Technique with Uniform Treatment of Dimensions. *Proceedings of the IEEE Information Visualization Symposium, 2001. Prieiga per: https://www.researchgate.net/publication/228605127_Star_Coordinates_A_Multi-Dimensional_Visualization_Technique_with_Uniform_Treatment_of_Dimensions.*
11. JOHANSSON, J., FORSELL, C. ir COOPER, M. On the Usability of Three-Dimensional Display in Parallel Coordinates: Evaluating the Efficiency of Identifying Two-Dimensional Relationships. *Information Visualization, 2014, vol. 13, no. 1. pp. 29-41 ISSN 1473-8716. DOI 10.1177/1473871613477091.*

12. LIND, M., JOHANSSON, J. ir COOPER, M. *Many-to-Many Relational Parallel Coordinates Displays*. 2009 13th International Conference Information Visualisation, 2009 ISBN 2375-0138. DOI 10.1109/IV.2009.43.
13. KUMASAKA, N. ir SHIBATA, R. High-Dimensional Data Visualisation: The Textile Plot. *Computational Statistics & Data Analysis*, 2008, vol. 52, no. 7. pp. 3616-3644. Prieiga per: <https://www.sciencedirect-com.ezproxy.ktu.edu/science/article/pii/S0167947307004513> ISSN 0167-9473. DOI 10.1016/j.csda.2007.11.016.
14. ETEMADPOUR, R., et al. Perception-Based Evaluation of Projection Methods for Multidimensional Data Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 2015, vol. 21, no. 1. pp. 81-94 ISSN 2160-9306. DOI 10.1109/TVCG.2014.2330617.
15. YIN, H. On Multidimensional Scaling and the Embedding of Self-Organising Maps. *Neural Networks*, 2008, vol. 21, no. 2. pp. 160-169 ISSN 0893-6080. DOI 10.1016/j.neunet.2007.12.027.
16. SAMMON, J.W. A Nonlinear Mapping for Data Structure Analysis. *IEEE Transactions on Computers*, 1969, vol. C-18, no. 5. pp. 401-409 ISSN 2326-3814. DOI 10.1109/T-C.1969.222678.
17. KOHONEN, T. The Self-Organizing Map. *Proceedings of the IEEE*, 1990, vol. 78, no. 9. pp. 1464-1480 ISSN 1558-2256. DOI 10.1109/5.58325.
18. XU, L., XU, Y. ir CHOW, T.W.S. PolSOM: A New Method for Multidimensional Data Visualization. *Pattern Recognition*, 2010, vol. 43, no. 4. pp. 1668-1675. Prieiga per: <https://www.sciencedirect.com/science/article/abs/pii/S0893608007002419> ISSN 0031-3203. DOI 10.1016/j.patcog.2009.09.025.
19. DIEHL, S., BECK, F. ir BURCH, M. Uncovering Strengths and Weaknesses of Radial Visualizations---an Empirical Approach. *IEEE Transactions on Visualization and Computer Graphics*, 2010, vol. 16, no. 6. pp. 935-942 DOI 10.1109/TVCG.2010.209.
20. RUBIO-SÁNCHEZ, M., RAYA, L., DÍAZ, F. ir SANCHEZ, A. A Comparative Study between RadViz and Star Coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 2016, vol. 22, no. 1. pp. 619-628 ISSN 2160-9306. DOI 10.1109/TVCG.2015.2467324.
21. HOFFMAN, P., et al. *DNA Visual and Analytic Data Mining*. Proceedings. Visualization '97 (Cat. No. 97CB36155), IEEE, 1997 DOI 10.1109/VISUAL.1997.663916.
22. SHARKO, J., GRINSTEIN, G. ir MARX, K.A. Vectorized Radviz and its Application to Multiple Cluster Datasets. *IEEE Transactions on Visualization and Computer Graphics*, 2008, vol. 14, no. 6. pp. 1444-1427 DOI 10.1109/TVCG.2008.173.
23. MAATEN, L. Visualizing Data using T-SNE. *Journal of Machine Learning Research*, 2008, vol. 9, no. Nov. pp. 2579. Prieiga per: <https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf?fbcl>.
24. VAN DER MAATEN, L. Accelerating T-SNE using Tree-Based Algorithms. *The Journal of Machine Learning Research*, 2014, vol. 15, no. 1. pp. 3221-3245. Prieiga per: <https://www.jmlr.org/papers/volume15/vandermaaten14a/vandermaaten14a.pdf>.

25. SPIWOK, V. ir KRÍŽ, P. Time-Lagged T-Distributed Stochastic Neighbor Embedding (T-SNE) of Molecular Simulation Trajectories. *Frontiers in Molecular Biosciences*, 2020, vol. 7. pp. 132 DOI 10.3389/fmolb.2020.00132.
26. PECORA, L.M., MONIZ, L., NICHOLS, J. ir CARROLL, T.L. A Unified Approach to Attractor Reconstruction. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 2007, vol. 17, no. 1. Prieiga per: <https://pubs.aip.org/aip/cha/article/17/1/013110/280310> DOI 10.1063/1.2430294.
27. JOHN WESLEY, R., NAYEEMULLA KHAN, A. ir SHAHINA, A. Phoneme Classification in Reconstructed Phase Space with Convolutional Neural Networks. *Pattern Recognition Letters*, 2020, vol. 135. pp. 299-306 ISSN 0167-8655. DOI 10.1016/j.patrec.2020.05.002.
28. ZHANG, J., et al. Time Series Prediction using Lyapunov Exponents in Embedding Phase Space. *Computers & Electrical Engineering*, 2004, vol. 30, no. 1. pp. 1-15. Prieiga per: <https://www-sciencedirect-com.ezproxy.ktu.edu/science/article/pii/S0045790603000156> ISSN 0045-7906. DOI 10.1016/S0045-7906(03)00015-6.
29. HAN, M., REN, W., XU, M. ir QIU, T. Nonuniform State Space Reconstruction for Multivariate Chaotic Time Series. *IEEE Transactions on Cybernetics*, 2018, vol. 49, no. 5. pp. 1885-1895.
30. LEKSCHA, J. ir DONNER, R.V. Phase Space Reconstruction for Non-Uniformly Sampled Noisy Time Series. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 2018, vol. 28, no. 8 [žiūrėta 3/27/2024]. pp. 085702. Prieiga per: <https://doi.org/10.1063/1.5023860> ISSN 1054-1500. DOI 10.1063/1.5023860.
31. MATILLA-GARCÍA, M., MORALES, I., RODRÍGUEZ, J.M. ir RUIZ MARÍN, M. Selection of Embedding Dimension and Delay Time in Phase Space Reconstruction Via Symbolic Dynamics. *Entropy*, 2021, vol. 23, no. 2. pp. 221 DOI <https://doi.org/10.3390/e23020221>.
32. YIBIN, L., YONG, S. ir CAIHONG, L. *Selection of Parameters for Phase Space Reconstruction of Chaotic Time Series*. 2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010 DOI 10.1109/BICTA.2010.5645296.
33. TAKENS, F. *Detecting Strange Attractors in Turbulence*. Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80: Springer, 2006 DOI <https://doi.org/10.1007/BFb0091924>.
34. KRÄMER, K., et al. A Unified and Automated Approach to Attractor Reconstruction. *New Journal of Physics*, 2021, vol. 23, no. 3. pp. 033017 DOI 10.1088/1367-2630/abe336.
35. TSUI, A.P., JONES, A.J. ir GUEDES DE OLIVEIRA, A. The Construction of Smooth Models using Irregular Embeddings Determined by a Gamma Test Analysis. *Neural Computing & Applications*, 2002, vol. 10. pp. 318-329 DOI <https://doi.org/10.1007/s005210200004>.
36. KRAEMER, K.H., et al. Optimal State Space Reconstruction Via Monte Carlo Decision Tree Search. *Nonlinear Dynamics*, 2022, vol. 108, no. 2. pp. 1525-1545 ISSN 1573-269X. DOI 10.1007/s11071-022-07280-2.
37. CHEN, L. ir BUJA, A. Local Multidimensional Scaling for Nonlinear Dimension Reduction, Graph Drawing, and Proximity Analysis. *Journal of the American Statistical Association*, 2009, vol. 104, no. 485. pp. 209-219 DOI <https://doi.org/10.1198/jasa.2009.0111>.

38. VIALE, L., DAGA, A.P., FASANA, A. ir GARIBALDI, L. Least Squares Smoothed K-Nearest Neighbors Online Prediction of the Remaining Useful Life of a NASA Turbofan. *Mechanical Systems and Signal Processing*, 2023, vol. 190. pp. 110154. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S0888327023000614> ISSN 0888-3270. DOI 10.1016/j.ymssp.2023.110154.
39. LIU, J., et al. Prediction of Remaining Useful Life of Multi-Stage Aero-Engine Based on Clustering and LSTM Fusion. *Reliability Engineering & System Safety*, 2021, vol. 214. pp. 107807. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S0951832021003306> ISSN 0951-8320. DOI 10.1016/j.ress.2021.107807.
40. WANG, Y., ZHAO, Y. ir ADDEPALLI, S. Remaining Useful Life Prediction using Deep Learning Approaches: A Review. *Procedia Manufacturing*, 2020, vol. 49. pp. 81-88. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S2351978920316528> ISSN 2351-9789. DOI 10.1016/j.promfg.2020.06.015.
41. ORDÓÑEZ, C., SÁNCHEZ LASHERAS, F., ROCA-PARDIÑAS, J. ir JUEZ, F.J.d.C. A Hybrid ARIMA–SVM Model for the Study of the Remaining Useful Life of Aircraft Engines. *Journal of Computational and Applied Mathematics*, 2019, vol. 346. pp. 184-191. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S037704271830428X> ISSN 0377-0427. DOI 10.1016/j.cam.2018.07.008.
42. WANG, T., GUO, D. ir SUN, X. Remaining Useful Life Predictions for Turbofan Engine Degradation Based on Concurrent Semi-Supervised Model. *Neural Computing and Applications*, 2022, vol. 34, no. 7. pp. 5151-5160 ISSN 1433-3058. DOI 10.1007/s00521-021-06089-1.
43. GAJJAR, S. ir PALAZOGLU, A. A Data-Driven Multidimensional Visualization Technique for Process Fault Detection and Diagnosis. *Chemometrics and Intelligent Laboratory Systems*, 2016, vol. 154. pp. 122-136. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S0169743916300703> ISSN 0169-7439. DOI 10.1016/j.chemolab.2016.03.027.
44. CHEN, J., WANG, C., WANG, B. ir ZHOU, Z. A Visualized Classification Method Via T-Distributed Stochastic Neighbor Embedding and various Diagnostic Parameters for Planetary Gearbox Fault Identification from Raw Mechanical Data. *Sensors and Actuators A: Physical*, 2018, vol. 284. pp. 52-65. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S0924424718307465> ISSN 0924-4247. DOI 10.1016/j.sna.2018.10.021.
45. HAN, T., et al. An Adaptive Spatiotemporal Feature Learning Approach for Fault Diagnosis in Complex Systems. *Mechanical Systems and Signal Processing*, 2019, vol. 117. pp. 170-187. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S0888327018304503> ISSN 0888-3270. DOI 10.1016/j.ymssp.2018.07.048.
46. XI, W., LI, Z., TIAN, Z. ir DUAN, Z. A Feature Extraction and Visualization Method for Fault Detection of Marine Diesel Engines. *Measurement*, 2018, vol. 116. pp. 429-437. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S0263224117307431> ISSN 0263-2241. DOI 10.1016/j.measurement.2017.11.035.
47. CARDENAS CABADA, E., LECLERE, Q., ANTONI, J. ir HAMZAOU, N. Fault Detection in Rotating Machines with Beamforming: Spatial Visualization of Diagnosis Features. *Mechanical Systems and Signal Processing*, 2017, vol. 97. pp. 33-43. Prieiga per:

<https://www.sciencedirect.com/science/article/pii/S0888327017302030> ISSN 0888-3270. DOI 10.1016/j.ymssp.2017.04.018.

48. ZHAO, M., JIN, X., ZHANG, Z. ir LI, B. Fault Diagnosis of Rolling Element Bearings Via Discriminative Subspace Learning: Visualization and Classification. *Expert Systems with Applications*, 2014, vol. 41, no. 7. pp. 3391-3401. Prieiga per:

<https://www.sciencedirect.com/science/article/pii/S0957417413009469> ISSN 0957-4174. DOI 10.1016/j.eswa.2013.11.026.

49. SAXENA, A. ir GOEBEL, K. Turbofan Engine Degradation Simulation Data Set. *NASA Ames Prognostics Data Repository*, 2008, vol. 18. Prieiga per: <https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/>.

50. SAXENA, A., GOEBEL, K., SIMON, D. ir EKLUND, N. *Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation*. 2008 International Conference on Prognostics and Health Management: IEEE, 2008 DOI 10.1109/PHM.2008.4711414.

51. LUO, P., et al. *A Moving Average Filter Based Method of Performance Improvement for Ultraviolet Communication System*. 2012 8th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP): IEEE, 2012 DOI 10.1109/CSNDSP.2012.6292672.

52. KONAK, A., COIT, D.W. ir SMITH, A.E. Multi-Objective Optimization using Genetic Algorithms: A Tutorial. *Reliability Engineering & System Safety*, 2006, vol. 91, no. 9. pp. 992-1007. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S0951832005002012> ISSN 0951-8320. DOI 10.1016/j.ress.2005.11.018.

53. RAGULSKIS, M. ir LUKOSEVICIUTE, K. Non-Uniform Attractor Embedding for Time Series Forecasting by Fuzzy Inference Systems. *Neurocomputing*, 2009, vol. 72, no. 10. pp. 2618-2626. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S0925231208004980> ISSN 0925-2312. DOI 10.1016/j.neucom.2008.10.010.

54. WANG, Z. ir BOVIK, A.C. Mean Squared Error: Love it Or Leave it? A New Look at Signal Fidelity Measures. *IEEE Signal Processing Magazine*, 2009, vol. 26, no. 1. pp. 98-117 DOI 10.1109/MSP.2008.930649.