



**Kauno technologijos universitetas**

Elektros ir elektronikos fakultetas

**Neinvazinio metabolitų kaupimosi metodo  
vystymas ir tyrimas**

Baigiamasis magistro projektas

---

**Deividas Masaitis**

Projekto autorius

**prof. dr. Renaldas Urniežius**

Vadovas

---

**Kaunas, 2023**



**Kauno technologijos universitetas**

Elektros ir elektronikos fakultetas

**Neinvazinio metabolitų kaupimosi metodo  
vystymas ir tyrimas**

Baigiamasis magistro projektas

Valdymo technologijos (6211EX014)

---

**Deividas Masaitis**

Projekto autorius

**prof. dr. Renaldas Urniežius**

Vadovas

**lekt. dr. Jolanta Repšytė**

Recenzentė

---

**Kaunas, 2023**



**Kauno technologijos universitetas**

Elektros ir elektronikos fakultetas

Deividas Masaitis

## **Neinvazinio metabolitų kaupimosi metodo vystymas ir tyrimas**

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.

Deividas Masaitis

*Patvirtinta elektroniniu būdu*

Masaitis Deividas. Neinvazinio metabolitų kaupimosi metodo vystymas ir tyrimas. Magistro baigiamasis projektas / vadovas prof. dr. Renaldas Urniežius; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): elektronikos inžinerija, inžinerijos mokslai.

Reikšminiai žodžiai: *biotechnologijos, metabolitai, acetatai, neuroniniai tinklai, optimizavimas*.

Kaunas, 2023. 47 p.

### **Santrauka**

Biotechnologiniai procesai siekia maksimizuoti tikslinio produkto gamybą ir bioproceso efektyvumą. Tačiau bioproceso metu susidarantys šalutiniai produktai, vadinami metabolitais, gali neigiamai paveikti proceso efektyvumą. Matuoti šių metabolitų koncentracijas yra brangu ir laiko reikalaujantis procesas, todėl naudojami programiniai jutikliai, paremti matematiniais modeliais, kurie naudoja lengvai surenkamą informaciją.

Šiame tyrime buvo palyginti 2 skirtingi dirbtinio neuroninio tinklo modeliai kartu su 2 duomenų apdirbimo metodais, skirti įvertinti *E. Coli* ląstelių gaminamų acetatų koncentraciją periodinių su pamaitinimu kultivavimo proceso metu. Pateikta šių modelių parametrų rinkiniai bei mokymo struktūra. Modelių parametrai optimizuoti siekiant surasti geriausius parametrų rinkinius. Visi šie metodai palyginti naudojant realius eksperimentinius duomenis bei pasirinktus tikslumo kriterijus.

Masaitis Deividas. Development and study of a non-invasive method for metabolite accumulation. Master's Final Degree Project / supervisor prof. dr. Renaldas Urniežius; Faculty of Electrical and Electronics Engineering, Kaunas University of Technology.

Study field and area (study field group): Electronics engineering, Engineering science

Keywords: *biotechnology, metabolites, acetates, neural networks, optimization.*

Kaunas, 2023. 47p.

### **Summary**

The goal of biotechnological processes is to maximize the output of target products and the efficiency of the bioprocess. However, the by-products of the bioprocess, known as metabolites, might harm process efficiency. Measuring the amounts of these metabolites is an expensive and time-consuming task. As a result, researchers employ software sensors based on mathematical models that rely on easily gathered data. In this study, we compared two artificial neural network models with two data processing methods for estimating the concentration of acetates produced by *E. coli* cells during periodic feeding cultivation. We supplied the parameter sets and training structures for these models. We improved the model parameters to discover the optimal parameter settings. We compared these approaches using experimental data and set accuracy criteria.

## Turinys

<b>Lentelių sąrašas .....</b>	<b>7</b>
<b>Paveikslų sąrašas .....</b>	<b>8</b>
<b>Įvadas.....</b>	<b>10</b>
<b>1. Literatūros analizė.....</b>	<b>11</b>
1.1. Ląstelių augimo procesas .....	11
1.2. Metabolitų susidarymas bei jų įtaka .....	13
1.3. Programiniai jutikliai.....	14
1.3.1. Mechanistiniai modeliai .....	14
1.3.2. Empiriniai modeliai .....	17
1.3.3. Hibridiniai modeliai.....	20
<b>2. Metodinė dalis.....</b>	<b>22</b>
2.1. Eksperimentų sąlygos ir duomenys modeliui.....	22
2.2. Pasikliautinumo intervalai .....	23
2.3. Eksperimento duomenų apdirbimas .....	24
2.3.1. Atskirosios reikšmės skaidymas.....	24
2.4. Principinės komponentės analizė .....	27
2.5. Dirbtiniai neuroniniai tinklai .....	31
2.5.1. Aktyvavimo funkcijos .....	32
2.5.2. Mokymo algoritmas.....	33
2.5.3. Tinklo struktūros parinkimas.....	34
<b>3. Eksperimentinio tyrimo rezultatai.....</b>	<b>36</b>
3.1. Nemodifikuoti duomenys .....	36
3.1.1. Atskirosios reikšmės skaidymas.....	37
3.1.2. Principinės komponentės analizė .....	38
3.1.3. Ilgalaikis trumpalaikės atminties neuroninis tinklas .....	40
3.2. Bajeso optimizavimas.....	41
<b>Rezultatai ir išvados .....</b>	<b>44</b>
<b>Literatūros sąrašas .....</b>	<b>45</b>

## Lentelių sąrašas

<b>1 lentelė.</b> Eksperimento metu palaikytos sąlygos .....	22
<b>2 lentelė.</b> Acetatų įvertinimui naudoti kultivavimo parametrai .....	23
<b>3 lentelė.</b> Acetatų koncentracijų grupės, pasikliautinumo intervalai ir vidurkiai .....	23
<b>4 lentelė.</b> Kokybės kriterijų priklausomybė nuo SVD dimensijų kiekio .....	37
<b>5 lentelė.</b> Kokybės kriterijų priklausomybė nuo PCA dimensijų kiekio.....	38
<b>6 lentelė.</b> Bajeso optimizavimo metu rasti modelių parametrai .....	42
<b>7 lentelė.</b> Metodų kokybės kriterijų vertės po Bajeso optimizavimo (BO).....	42

## Paveikslų sąrašas

<b>1.1.1 pav.</b> Kultivavimo proceso sudedamosios dalys .....	11
<b>1.1.2 pav.</b> Supaprastinta bioreaktoriaus sistema .....	12
<b>1.1.3 pav.</b> Skirtingos ląstelių augimo fazės .....	13
<b>1.3.1 pav.</b> Estimuotos biomasės palyginimas tarp tradicinio, neuroniniais tinklais grįsto ir hibridinio modelio. ....	14
<b>1.3.2 pav.</b> Rekursinio būsenos estimavimo modelio struktūra .....	16
<b>1.3.3 pav.</b> Acetatų įvertinimas naudojant ištirpusio deguonies koncentracijos matavimus. Juodai – įvertintos vertės, mėlynai – realios vertės, žaliai – 99% pasikliautinas intervalas (angl. <i>confidence interval</i> ) .....	16
<b>1.3.4 pav.</b> Biomasės estimavimo paklaidos .....	17
<b>1.3.5 pav.</b> (A) FANN struktūra, (B) RBPN struktūra .....	18
<b>1.3.6 pav.</b> (A) FANN įvertintos ir išmatuotos parametrų vertės, (B) RBPN įvertintos ir išmatuotos parametrų vertės .....	18
<b>1.3.7 pav.</b> <i>Fuzzy</i> logikos regulatoriaus atsako paviršius sudarytas iš 6 taisyklių.....	19
<b>1.3.8 pav.</b> Ištirtų ir įvertintų parametrų grafikai. BC-1, BC-2, BC-3 skirtingi eksperimentai. A – biomasė, B – specifinis augimo greitis, C – gliukozės koncentracija. ....	20
<b>1.3.9 pav.</b> Hibridinio modelio struktūra .....	20
<b>1.3.10 pav.</b> Modelių naudotų biomasės įvertinimui palyginimas .....	21
<b>2.2.1 pav.</b> Pasikliautinumo intervalo grafikas .....	24
<b>2.3.1 pav.</b> Atskirosios reikšmės skaidymas.....	25
<b>2.3.2 pav.</b> Atskirosios reikšmės skaidymo dalis Excel programinėje įrangoje.....	25
<b>2.3.3 pav.</b> SVD kumuliacinė paaiškinamoji variacija .....	26
<b>2.3.4 pav.</b> Atskirosios reikšmės skaidymo skaičiavimo algoritmo schema .....	26
<b>2.3.5 pav.</b> SVD skaičiavimo ir rekonstrukcijos <i>Python</i> kodas.....	27
<b>2.4.1 pav.</b> Kovariacijų matrica .....	28
<b>2.4.2 pav.</b> Principinės komponentės analizės dalis Excel programinėje įrangoje.....	28
<b>2.4.3 pav.</b> PCA kumuliacinė paaiškinamoji variacija .....	29
<b>2.4.4 pav.</b> Koreliacijų matrica tarp originalių duomenų ir PCA komponentių .....	29
<b>2.4.5 pav.</b> Principinės komponentės analizės skaičiavimo algoritmo schema.....	30
<b>2.4.6 pav.</b> PCA skaičiavimo ir rekonstrukcijos <i>Python</i> kodas 1 .....	30
<b>2.4.7 pav.</b> PCA skaičiavimo ir rekonstrukcijos <i>Python</i> kodas 2 .....	31
<b>2.5.1 pav.</b> Daugiasluoksnių tinklo pavyzdys, sudarytas iš keturių įėjimo neuronų, penkių paslėpto sluoksnių neuronų ir dvejų išėjimo neuronų.....	31
<b>2.5.2 pav.</b> Struktūrinė LSTM schema [38].....	32
<b>2.5.3 pav.</b> Hiperbolinio tangento ir sigmoidinė funkcija .....	33
<b>2.5.4 pav.</b> ReLU aktyvavimo funkcija .....	33
<b>2.5.5 pav.</b> Mokymo algoritmo principinė schema .....	35
<b>2.5.6 pav.</b> Tyrimo metu naudoto kodo dalis .....	35
<b>3.1.1 pav.</b> Patikrinimo rezultatai naudojant duomenis be PCA ar SVD modifikacijos .....	36
<b>3.1.2 pav.</b> MAE priklausomybė nuo SVD dimensijų kiekio .....	37
<b>3.1.3 pav.</b> Prognozuotų duomenų palyginimas su išmatuotais naudojant SVD metodą.....	38
<b>3.1.4 pav.</b> MAE priklausomybė nuo PCA dimensijų kiekio.....	39
<b>3.1.5 pav.</b> Prognozuotų duomenų palyginimas su realiais naudojant PCA metodą.....	39
<b>3.1.6 pav.</b> Prognozuotų duomenų palyginimas su realiais naudojant LSTM metodą.....	40



<b>3.2.1 pav.</b> MFFN modelio kodas su keičiamais parametrais .....	41
<b>3.2.2 pav.</b> Prognozuotų duomenų palyginimas su realiais naudojant MFFN metodą.....	43
<b>3.2.3 pav.</b> MFFN metodo mokymo funkcijos priklausomybė nuo epochų skaičiaus .....	43

## Įvadas

Bioprocesai susideda iš sudėtingų matematinių modelių, triukšmingų duomenų bei komplikuo­tų valdymo sistemų [1,2]. Informacijos surinkimas yra laiką užimantis ir brangus uždavinys. Pagrindiniams ląstelių būsenos kintamiesiems, tokiems kaip biomasės, tikslinio produkto koncentracijoms, tirti vis dar trūksta realiu laiku veikiančių jutiklių. Šiam tikslui pradėti naudoti netiesioginiai būsenos kintamųjų matavimo būdai. Šie matavimai gaunami naudojant matematinius modelius aprašančius ryšius tarp realiu laiku matuojamų fizikinių dydžių, tokių kaip temperatūra, ištirpęs deguonis, terpės rūgštingumas ar informacija surenkama iš išmetamųjų dujų. Šiems modeliams sudaryti naudojami tiek tradicinės balanso lygtys tiek juodosios dėžės modeliai tokie kaip dirbtiniai neuroniniai tinklai.

Šiame tyrime bus siekiama sukurti programinį jutiklį, kuris leistų stebėti *E. coli* metabolitų, acetatų, koncentracijas bioproceso metu. Per didelės šių cheminių junginių koncentracijos kenkia ląstelių augimui bei tikslinio produkto gamybai.

**Darbo tikslas.** Įvertinti acetatų koncentracijos pokyčius *E. Coli* kultivavimo metu, pasitelkiant skirtingus duomenų apdorojimo ir juodos dėžės modelius, o taip pat palyginti jų tikslumą naudojant eksperimentinius duomenis.

### Darbo uždaviniai:

1. Realizuoti atskirosios reikšmės skaidymo ir principinės komponentės analizės metodus duomenų reikšmingumui tirti;
2. Išvystyti matematinę modeliavimo metodiką, grindžiamą dirbtiniais neuroniniais tinklais, parenkant optimalius įvesties kintamuosius ir tinklo struktūros parametrus, ir patikrinti modelių adekvatumą naudojant surinktus eksperimentinius duomenis
3. Palyginti skirtingus matematinius modelius, naudojant atrinktus vertinimo kriterijus, siekiant įvertinti jų tinkamumą acetatų koncentracijos prognozavimui *E. Coli* kultivavimo metu.

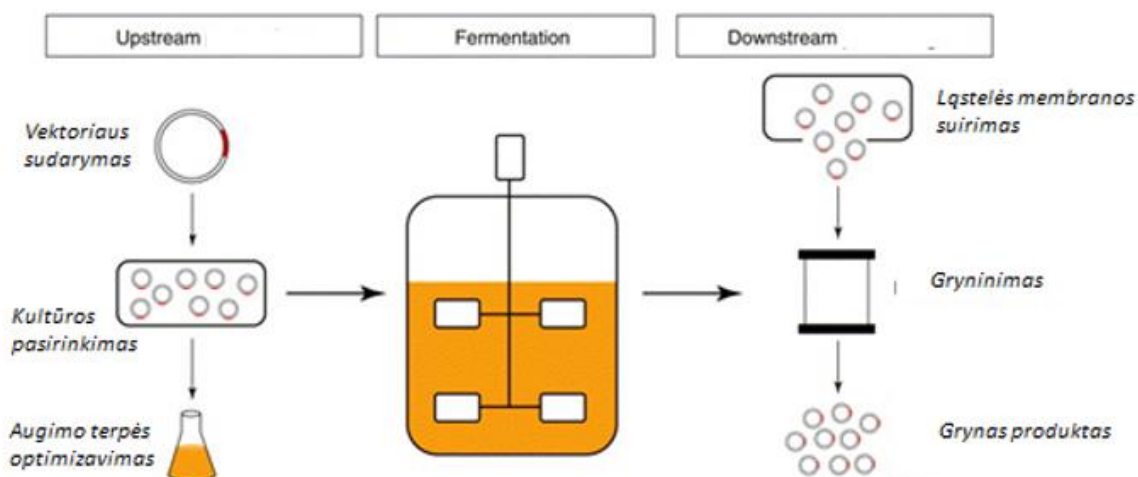
Pirmajame skyriuje aprašomas tiriamas bioprocesas ir apžvelgiami skirtingi matematiniai modeliai acetatų koncentracijai įvertinti. Antrajame skyriuje pristatoma tyrimo metodika, naudoti įrankiai, duomenys bei metodai. Trečiajame skyriuje pateikti gauti modelio rezultatai su eksperimentiniais duomenimis, pasirinkta geriausia struktūra ir jo parametrai.

## 1. Literatūros analizė

Mikroorganizmai yra neatskiriama mūsų gyvenimo dalis ir naudojami maisto ir gėrimų gamyboje. Jau senovės Egipte buvo naudojami mikroorganizmai duonos ir alaus gamybai. Biotechnologijos, kokios yra žinomos šiuo metu, yra jau nuo 19 a., kai - pradėta plačiau tyrinėti ir domėtis bioprocerais, jų komponentais bei savybėmis. Pirmoji didelio masto fermentacija, kurios tikslas buvo sintetinti citrinų rūgštį ir peniciliną, yra žinoma jau seniai [3]. Antibiotikų gamyba tuo metu atrodė panašiai kaip vyno gamyba: visas procesas buvo atliekamas rankomis ir tik temperatūra buvo valdoma automatinio būdu. Duomenys buvo renkami ir užrašomi ranka, o tokie svarbūs parametrai kaip terpės rūgštingumas ar ištirpęs deguonis nebuvo matuojami. Pastaraisiais dešimtmečiais biotechnologijų procesas smarkiai pažengė. Sukurti nauji jutikliai ir kompiuterizuotos sistemos leido ne tik stebėti procesą, bet ir atlikti jo valdymą, siekiant kuo daugiau sumažinti gamybos kaštus ir padidinti proceso išeią [4].

### 1.1. Ląstelių augimo procesas

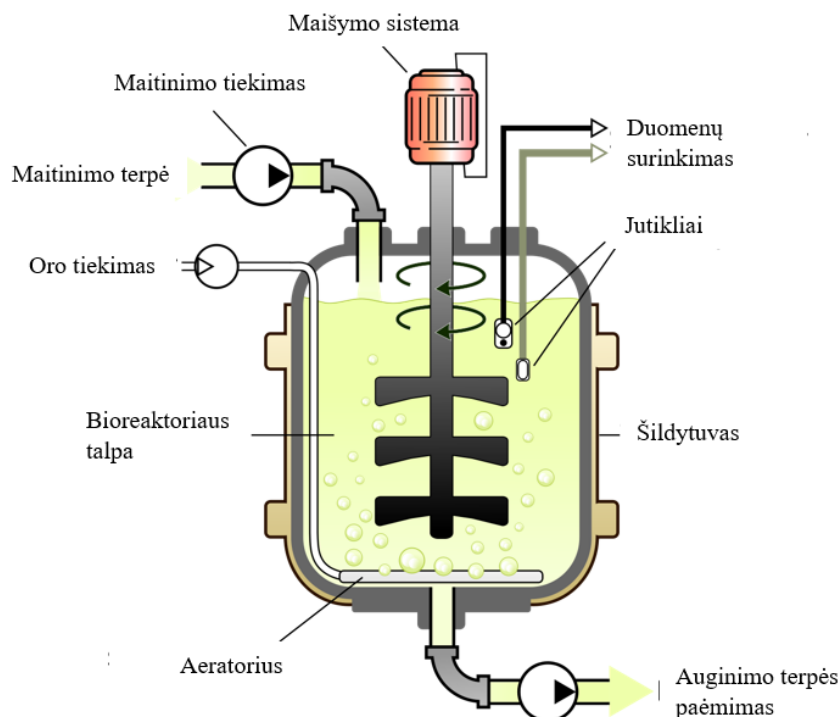
Ląstelių auginimas, dar vadinamas kultivavimu arba fermentavimu, vyksta bioreaktoriuose, kurių dydis gali svyruoti nuo kelių litrų iki kelių tonų. Fermentacijos procesas susideda iš ląstelių paruošimo ir auginimo (angl. *Upstream bioprocess*) bei produkto gryninimo proceso (angl. *Downstream bioprocess*) [5] (žr. 1.1.1 pav.).



1.1.1 pav. Kultivavimo proceso sudedamosios dalys

Ląstelių auginimo procese, pasirenkama ląstelių kultūra priklausomai nuo pageidaujamo produkto. Vėliau yra sudaroma ir optimizuojama atitinkama auginimo terpė, kurios sudėtis priklauso nuo reikiamų mineralų ir jų kiekio, kurie yra reikalingi stabiliam ir greitam ląstelių dauginimui. Pirminis kultūros auginimas atliekamas mažais kiekiais, o užtikrinti palankias aplinkos sąlygas yra naudojami inkubatoriai. Šie prietaisai palaiko tinkamą temperatūrą, drėgmę ir kartais reikalingą aplinkos dujų mišinį. Pradinis kultūros auginimas skirtas ląstelėms "pažadinti", t. y. kai kultūra, pasiekdama optimalias augimo sąlygas, pradeda sparčiai daugintis. Vėliau ši kultūra perkeliama į bioreaktorių / fermentatorių. Šio prietaiso paskirtis auginti kultūros biomasę dideliais kiekiais, o homogeniškam medžiagų ir ląstelių paskirstymui palaikyti naudojamas maišytuvas. Tinkamai terpės temperatūrai užtikrinti naudojama bioreaktoriaus šildymo ar šaldymo sistema. Temperatūros, pH, slėgio, ištirpusio deguonies matavimo prietaisai, skirti ne tik užtikrinti optimalų augimą, bet ir stebėti bei vertinti

procesą. Priklausomai nuo auginimo pobūdžio yra naudojama pamaitinimo pompa, skirta maitinimo substrato tiekimui į bioreaktorių (žr. 1.1.2 pav.).



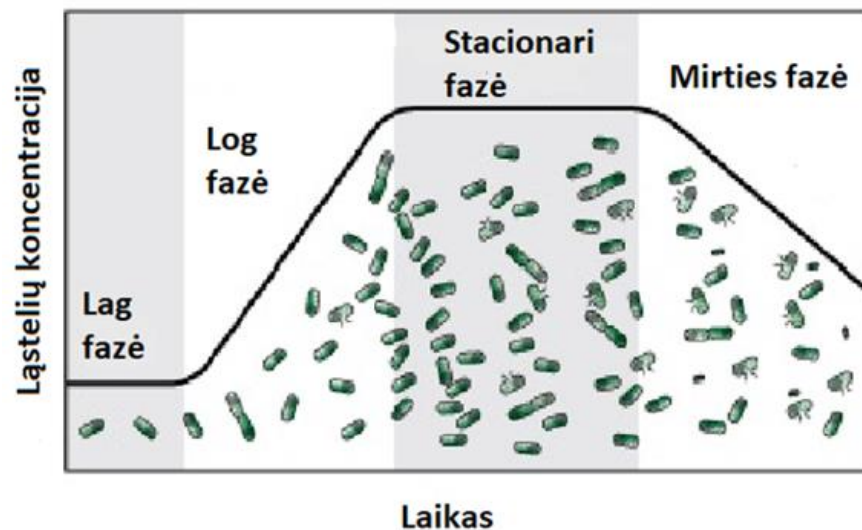
1.1.2 pav. Supaprastinta bioreaktoriaus sistema

Proceso metu naudojami ne tik realaus laiko jutikliai, bet ir imami mėginiai. Šie mėginiai yra naudojami matuoti ląstelių optinį tankį, biomasę ar substrato koncentraciją. Šie parametrai padeda nustatyti proceso kokybę, numatyti jo eigą bei ateityje atlikti tyrimus, padedančius geriau suprasti arba efektyviau vykdyti produkto gamybą.

Pramonėje yra naudojami skirtingi fermentacijos būdai. Jie pasirenkami priklausomai nuo poreikių ir turimų galimybių:

- **Pakartotina fermentacija.** Šis kultivavimo metodas yra dažniausiai naudojamas mažo masto procesams. Bioreaktorių yra užpildomas tam tikru ribotu mineralų ir maistinių medžiagų kiekiu [6]. Po kultūros inokuliacijos, tai yra po kultūros įleidimo į fermentatorių, procesas tęsiasi iki numatyto laiko arba iki maistinių medžiagų išnaudojimo. Užaugusi biomasė yra išimama ir naudojama produktui atskirti ir gryninti. Bioreaktorių yra išvalomas ir paruošiamas naujam fermentacijos ciklui.
- **Pakartotina fermentacija su pamaitinimu.** Priešingai nei pakartotina fermentacija be pamaitinimo, šios fermentacijos metu maistinių medžiagų kiekis nėra ribojamas pradiniu užpildymu. Maistinės medžiagos yra tiekiamos iš išorės, užtikrinant pakankamą jų kiekį. Šis būdas leidžia realiu laiku keisti substrato koncentraciją auginimo terpėje, o substratas gali būti tiekiamas atsižvelgiant į iš anksto sudarytus dozavimo greičio ar kiekio profilius, arba priklausomai nuo grįžtamojo ryšio parametrų: ištirpusio deguonies, pH ar deguonies suvartojimo greičio [7]. Tai leidžia užauginti didesnius kiekius ląstelių nei pakartotina fermentacija be pamaitinimo.

— **Nepertraukiama fermentacija.** Pakartotinos fermentacijos metu ląstelių augimas yra ribotas ir nusakomas augimo kreive (žr. 3.1.3 pav.). Ląstelių augimas prasideda latentinėje (angl. *lag*) fazėje, kurioje augimas ir dauginimasis yra labai lėtas. Šią fazę pakeičia eksponentinio augimo (angl. *log*) fazė. Šiuo metu ląstelių kiekis auga eksponentiškai ir yra užauginama didžioji dalis visos kultivavimo biomasės. Baigus dauginimuisi, pasiekama stacionari fazė, o vėliau prasideda mirties arba eksponentinio mažėjimo fazė, kai trūksta maistinių medžiagų. Siekiant didžiausios produkto išeigos, reikia didelio biomasės kiekio, kuris pasiekiamas nepertraukiamos fermentacijos metu. Tam tikras kiekis maistinių medžiagų yra tiekiamas į bioreaktorių, o auginimo terpė yra šalinama kartu su išskirtu produktu tokiu pat greičiu taip išlaikant vienodą terpės tūrį bioreaktoriuje[8]. Ši sistema leidžia palaikyti fermentaciją eksponentinio augimo fazėje.



1.1.3 pav. Skirtingos ląstelių augimo fazės

Po auginimo proceso ląstelės yra perduodamos produkto išskyrimui. Šio proceso metu jų membranų sienelės yra suskaldomos, o vėliau, naudojant cheminius ar mechaninius procesus, produktas yra išgryninamas.

## 1.2. Metabolitų susidarymas bei jų įtaka

Pramonės tikslas užauginti kuo didesnius tikslinio produkto kiekius. Tačiau augant ląstelių kultūrai susidaro ne tik reikalinga medžiaga bet ir pašaliniai produktai, vadinami metabolitais. *E. coli* ląstelių kultūroje dažniausiai susidaro acetatai, kurie trukdo produkto sintezei ir sukelia inhibiciją. Ši problema ypač pastebima vėlesnėse augimo fazėse. Inhibicija pailgina latentinę fazę ir taip sumažina biomasės augimo greitį ir kiekį [9]. Didėjantis acetatų kiekis rūgština fermentacijos terpę bei pažeidžia vidinius ląstelės procesus [10,11].

Aerobinio proceso metu metabolitų susidarymą gali paveikti du veiksniai. Pirmasis - nepakankamas deguonies kiekis reaktoriuje, kuris gali kilti dėl oro tiekimo arba homogeniškumo trūkumo terpėje. Antrasis veiksnys yra netinkamas maistinių medžiagų, dažniausiai gliukozės, tiekimas. Tiekiamo substrato kiekis yra priklausomas nuo gliukozės suvartojimo greičio ir jos konvertavimo į biomasę [12]. Metabolitų susidarymas priklauso ne tik nuo netinkamo auginimo sąlygų, bet ir nuo ląstelės kultūros [13]. Žinant acetatų kiekį, galima geriau suprasti ląstelių augimo ir vystymosi procesus ir sukurti efektyvesnius bioproceso valdymo algoritmus. Acetatų ir kitų metabolitų kiekiai išmatuojami

iš augimo terpės mėginių, naudojant chemines reakcijas. Tačiau šis procesas reikalauja tam tikrų žinių, medžiagų ir įrankių. Todėl pramonėje vis dažniau naudojami programiniai jutikliai.

### 1.3. Programiniai jutikliai

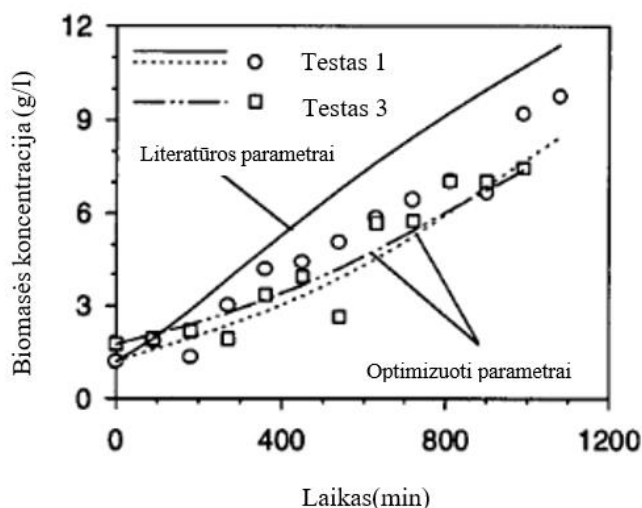
Programiniai jutikliai naudoja realiu laiku išmatuojamus bioproceso parametrus: temperatūrą, slėgį, bioreaktoriaus svorį, suvartojamo deguonies kiekį. Šie išmatuoti parametrai kartu su įvairiais matematiniais modeliais sudaro programinius jutiklius. Acetatų matavimui yra naudojama daug skirtingų matematinių modelių, nuo tradicinių mechanistinių modelių iki duomenimis paremtų empirinių modelių, tokių kaip dirbtiniai neuroniniai tinklai (ANN)[14].

#### 1.3.1. Mechanistiniai modeliai

Tradiciniai matematiniai modeliai yra paremti fundamentaliais principais: energijos tvermės dėsniumi, balanso lygtimis. Masės ir energijos tvermės dėsniai yra vieni iš dažniausiai taikomų sudarant modelius. Mechanistinių modelių pagrindą sudaro ne tik fundamentalieji principai, bet ir žinios atsinešamos iš chemijos bei biologijos. Tradiciniai modeliai naudojami biotechnologijose, skirstomi į dvi šakas: struktūrizuotus ir nestruktūrizuotus modelius [15].

##### 1.3.1.1. Struktūrizuotas modelis

Norint sudaryti struktūrizuotus modelius svarbu turėti galias žinias apie fizinius, cheminius ir biocheminius procesus, vykstančius bioproceso metu [16]. Šių modelių esmė – tikslus sintezės metu vykstančių procesų matematinis aprašymas. Tačiau jų kūrimas yra sudėtingas ir laiko reikalaujantis procesas. Šie modeliai yra kuriami vienam ląstelių tipui ar rūšiai, o jų pritaikymas kitų ląstelių parametrams įvertinti gali būti netikslus. Šis specializavimasis neleidžia panaudoti sukurtų modelių lanksčiai ir dinamiškai, tai yra jų tikslumą gali sugadinti modelyje neaprašyta sintezės proceso dalis ar pasikeitimas. *Feyo 'as de Azevedo 'as* publikacijoje palygino tradicinį, neuroniniais tinklais grįstą ir hibridinį modelį. Visi šie modeliai buvo naudojami mielių biomasės estimavimui.



1.3.1 pav. Estimuotos biomasės palyginimas tarp tradicinio, neuroniniais tinklais grįsto ir hibridinio modelio.

Kaip matyti iš paveikslo (žr. 1.3.1 pav.) tradicinis modelis prognozavo biomasės koncentraciją labai tolygiai ir be papildomo triukšmo. Tačiau abu tiek empirinis, tiek hibridinis modeliai estimavo biomasės koncentraciją daug tiksliau.

### 1.3.1.2. Nestruktūrizuotas modelis

Nestruktūrizuoti bioproceso modeliai nesistengia aprašyti kiekvieno ląstelės proceso. Nestruktūrizuoti modeliai apibūdina, sintezės proceso parametrų, tarpusavio ryšius. Dažniausiai šie modeliai paremi masės ir energijos balanso lygtimis ir kinetinėmis išraiškomis, tokiomis kaip *Monod* lygtis [17]. *Monod* išraiška aprašo vieną svarbiausių bioproceso parametrų – specifinį augimo greitį:

$$\mu = \mu_{\max} \frac{[S]}{K_s + [S]} \quad (1)$$

, kur  $\mu$  - specifinis mikroorganizmų kultūros augimo greitis,  $\mu_{\max}$  – maksimalus kultūros augimo greitis,  $K_s$  – eksperimentiniu būdu nustatoma konstanta,  $[S]$  – kultūrai tiekiamo substrato koncentracija. Ši lygtis nėra vienintelė kinetinė išraiška naudojama specifiniam augimo greičiui gauti. Mpho Muliowa savo darbe palygino 14 skirtingų kinetinių išraiškų [18]. Palyginimą sudarė *Monod*, *Blackman*, *Haldane* ir kitos išraiškos. Kiekviena iš jų yra tinkama šiam darbui, bei turi savus panaudojimo niuansus. Specifinis acetatų gaminimo ir suvartojimo greitis taip pat gali būti išreikštas *Monod* lygtimis [19].

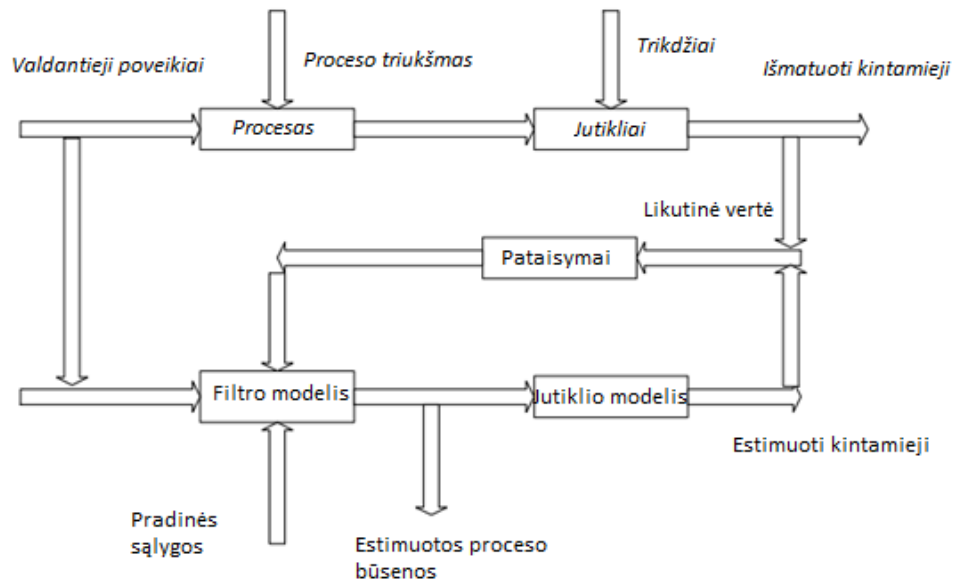
$$q_{Acp} = q_{Acp\max} \frac{S}{S + K_{sAc}}, \quad (2)$$

kur  $q_{Acp}$  – specifinis acetatų gaminimo greitis,  $q_{Acp\max}$  – maksimalus acetatų gaminimo greitis,  $K_{sAc}$  – eksperimentiniu būdu nustatoma konstanta,  $S$  – kultūrai tiekiamo substrato koncentracija.

$$q_{Acc} = q_{Acc\max} \frac{Ac}{Ac + K_{sAc}} \frac{K_{iS}}{K_{iS} + S}, \quad (3)$$

kur  $q_{Acc}$  – specifinis acetatų suvartojimo greitis,  $q_{Acc\max}$  – maksimalus acetatų suvartojimo greitis,  $K_{sAc}$  – eksperimentiniu būdu nustatoma konstanta,  $S$  – kultūrai tiekiamo substrato koncentracija,  $Ac$  – acetatų kiekis,  $K_{iS}$  – eksperimentiniu būdu nustatoma konstanta.

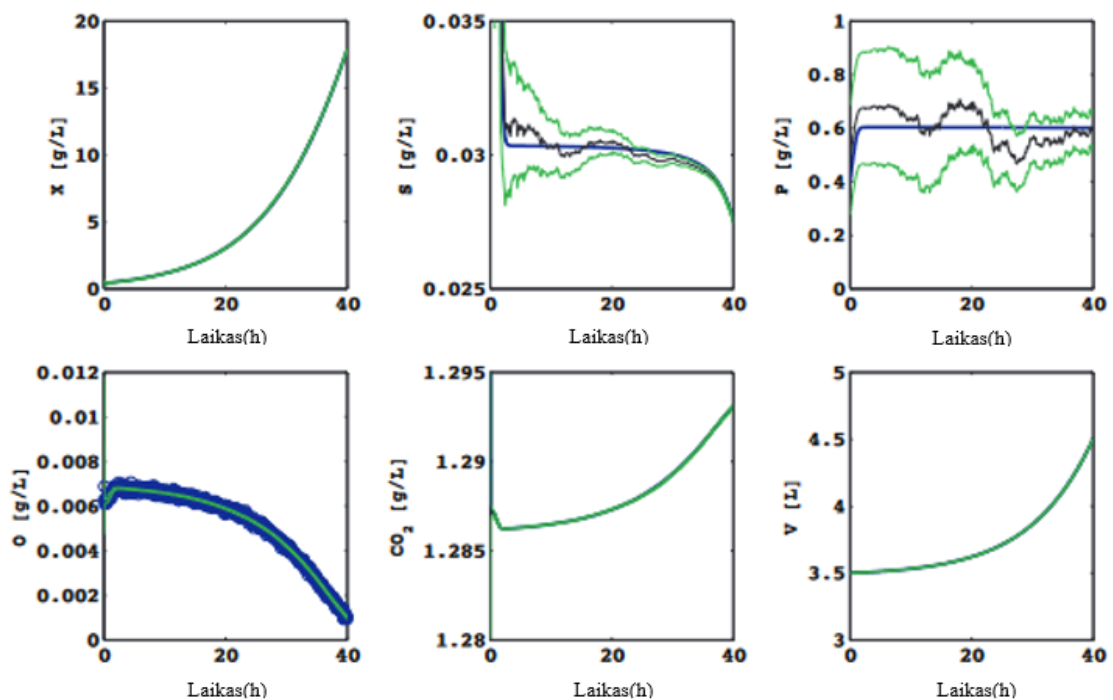
Be kinetinių išraiškų praktikoje labai populiarus išplėstas Kalmano filtras ( angl. *Extended Kalman Filter, EKF*). Paprastas Kalmano filtras yra skirtas būsenų įvertinimui (angl. *estimation*) tiesinėse sistemose, tačiau dėl bioprocesų netiesiškumo, šis metodas šioms sistemoms nėra tinkamas. Netiesiškumams įvertinti šis metodas buvo adaptuotas - išplėstas. EKF yra rekursyvus modelis, kurio schemą pateikta 1.3.2 paveiksle.



1.3.2 pav. Rekursinio būsenos estimavimo modelio struktūra

EKF efektyvumas buvo patvirtintas *L. Dewasme* tyrime. Šio tyrimo metu eksperimentiškai pagrįstas (angl. *validated*) išplėstas Kalmano filtras naudojamas *E. Coli* kultūros acetatų įvertinimui [20]. 1.3.3 paveiksle pateiktas acetatų įvertinimas naudojant tik ištirpusio deguonies koncentraciją. EKF modelis, su informacija apie biomasės ir ištirpusio deguonies koncentraciją, yra tinkamas metodus realiu laiku įvertinti acetatų bei gliukozės koncentraciją bioreaktoriaus terpėje.

Nestruktūrizuoti modeliai yra daug paprastesni nei struktūrizuoti. Šie modeliai yra tinkami ilgalaikiams procesams bei būsenoms nustatyti, tačiau jie netinka bioproceso dinamikai ar staigių pakeitimų stebėjimui.



1.3.3 pav. Acetatų įvertinimas naudojant ištirpusio deguonies koncentracijos matavimus. Juodai – įvertintos vertės, mėlynai – realios vertės, žaliai – 99% pasikliautinis intervalas (angl. *confidence interval*)



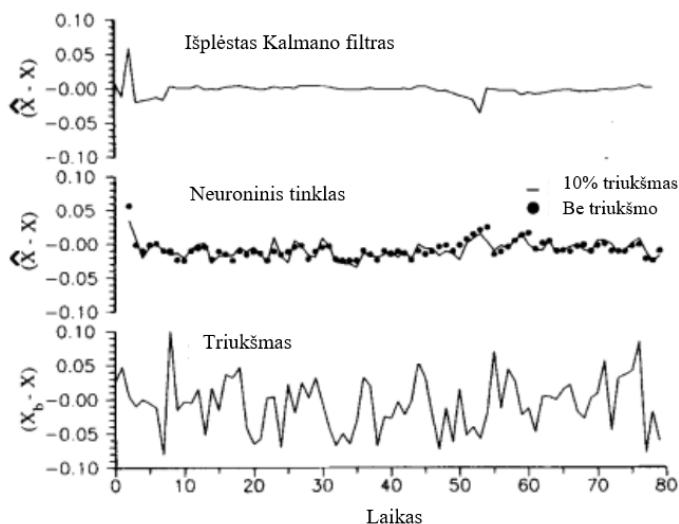
### 1.3.2. Empiriniai modeliai

Mechanistinių modelių yra sukurta daug ir įvairių, tačiau šių modelių pritaikymas dinamiškiems, daugiamačiams procesams nesuteikia norimų rezultatų ir tikslumo. Dėl to kaip alternatyva naudojami empiriniai modeliai. Šiems modeliams sudaryti nereikia tikslaus proceso aprašymo, tačiau reikia didelio kiekio duomenų. Dažniausiai naudojami empiriniai modeliai yra dirbtiniai neuroniniai tinklai (toliau – ANN, angl. *artificial neural networks*), neraiškiųjų aibių logikos (angl. *Fuzzy logic*) bei kita.

#### 1.3.2.1. Dirbtiniais neuroniniais tinklais grįsti programiniai jutikliai

ANN panaudojimas kompleksišku uždavinių sprendimui, nėra nauja idėja. Paskutinius 40-50 metų mokslininkai siekė simuliuoti smegenų struktūrą, norint sukurti algoritmą gebantį save apmokyti [21]. To siekiant buvo sukurti paprastesni būdai, naudojami duomenų analizėje ar parametru įvertinime. Literatūroje randama daugiau nei 50 skirtingų tipų neuroninių tinklų architektūrų [22].

1990-aisiais, pasiūlyti pirmieji metodai, prognozuoti biotechnologinio proceso kintamuosius naudojant neuroninius tinklus. Šiems tinklams apmokyti naudojama sukaupta fermentacijų duomenų bazė [23]. Šios publikacijos autoriai lygina, biomasės ir substrato koncentracijos įvertinimą tarp tradicinių modelių ir neuroninių tinklų. Žemiau esančiame paveiksle (žr. 1.3.4 pav.) pavaizduotos biomasės įvertinimo paklaidos esant 10% triukšmui.



1.3.4 pav. Biomasės estimavimo paklaidos

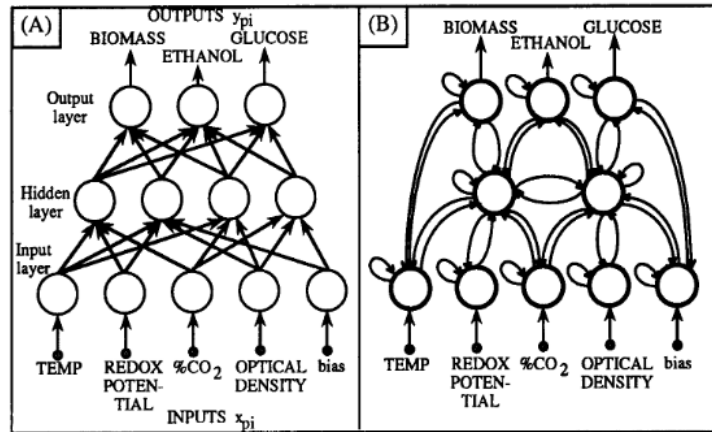
Iš tyrimo rezultatų matyti jog neuroninis tinklas sugebėjo net ir esant 10% triukšmui gana tiksliai įvertinti norimus parametrus. Autorių nuomone didžiausias neuroninių tinklų privalumas - sugebėjimas sudaryti proceso modelį neturint išankstinių žinių apie procesą.

Bėgant metams, bioprocusuose buvo išbandyta daug skirtingų neuroninių tinklų architektūrų. Dažniausiai sutinkamos:

- Tiesinio sklidimo neuroninis tinklas (angl. *Feedforward Artificial Neural Network, FANN*);
- Rekurentinis atgalinio sklidimo neuroninis tinklas (angl. *Recurrent Backpropagation Neural Networks RBPN*);

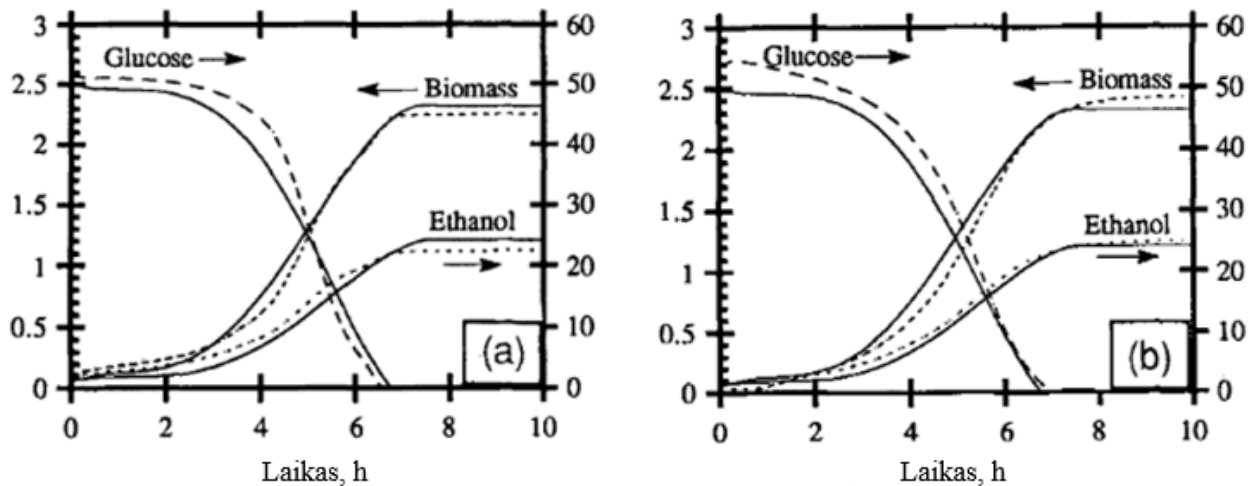
- Treniruojamas atgalinio sklaidimo rekurentinis neuroninis tinklas (angl. *Recurrent Trainable Neural Networks RTNN*);
- Afininiai tiesioginio sklaidimo valdymo neuroniniai tinklai (angl. *Control Affine Feedforward Neural Networks (CAFNN)*) [24].

M.N. Karim ir S.L. Rivera savo tyrime palygino FANN ir RBPN architektūras - biomasės, gliukozės ir etanolio koncentracijų prognozavimui [25].



1.3.5 pav. (A) FANN struktūra, (B) RBPN struktūra

Parametrų tikslumo vertinimui buvo naudotas kvadratinų paklaidų sumos (SSE) kriterijus. Abu šie metodai pasiekė panašius rezultatus: FANN architektūros (empirinio modelio) paklaidos kriterijus buvo lygus 0,03, o naudojant RBPN (mechanistinį modelį) – 0,04 (žr. 1.3.6 pav.).



1.3.6 pav. (A) FANN įvertintos ir išmatuotos parametrų vertės, (B) RBPN įvertintos ir išmatuotos parametrų vertės

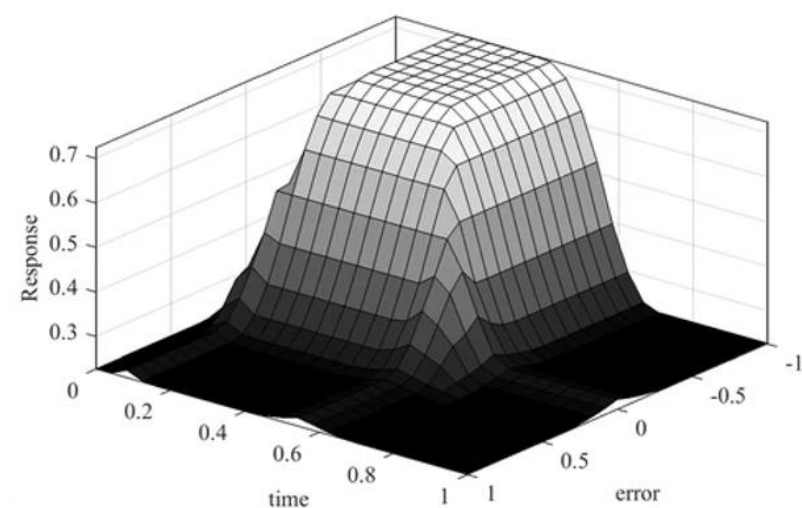
Abiejų architektūrų rezultatai rodo, kad metodai yra tinkami bioprocesų parametrų įvertinimo uždaviniams spręsti.

ANN architektūros pasirinkimas priklauso nuo jo taikymo ir duomenų pateikimo. Pernelyg sudėtingi modeliai yra linkę į problemas mokymosi metu, o optimalus neuronų ir sluoksnių skaičius priklauso nuo užduoties pobūdžio ir sudėtingumo lygio. Šie parametrai parenkami bandymų būdu arba

naudojantis tam tikromis taisyklėmis. ANN struktūros pasirinkimas, suderinimas bei mokymas reikalauja atlikti daug eksperimentų. Naudojant ANN pasitaiko ir atvejų, kada įvertintos vertės neatitinka fundamentalių apribojimų, ypačingai tada, kai realiu laiku susiduriama su duomenimis, kurie nebuvo naudojami mokymo metu.

### 1.3.2.2. Neraiškiųjų aibių logika grįsti programiniai jutikliai

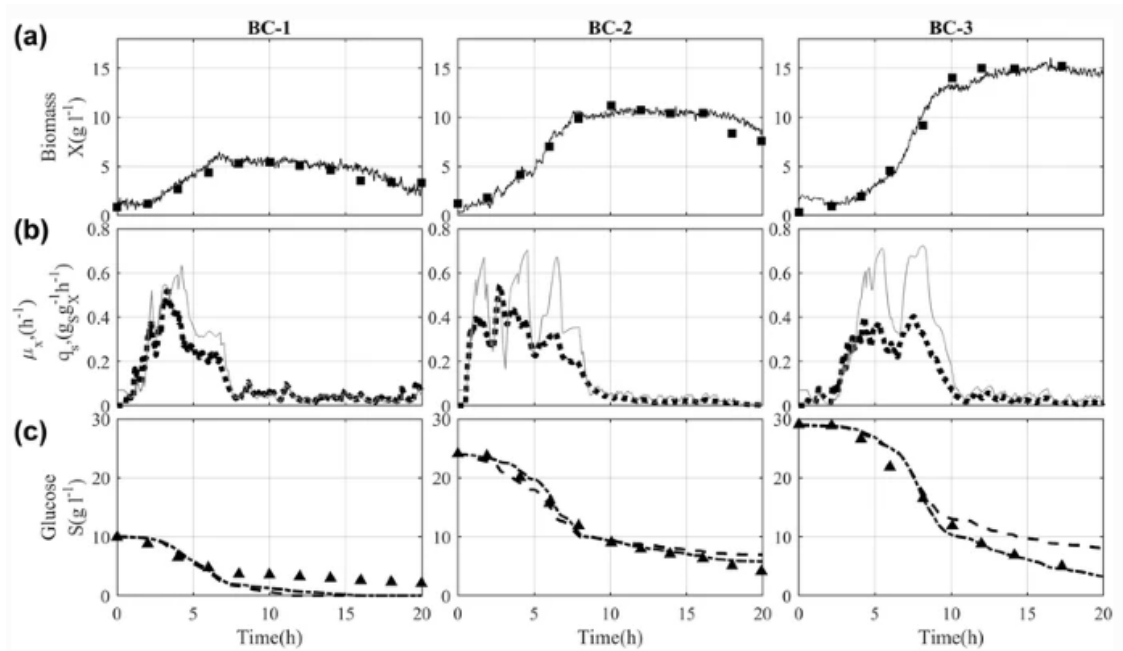
Zadeh'as 1965 metais pirmasis pristatė neraiškiųjų aibių logiką (angl. *Fuzzy logic*) [26]. Ši logika naudojama procesų valdyje ir modeliavime. Neraiškiųjų aibių logika naudojama sistemose, kuriose nereikia tikslios informacijos apie procesą. Šis metodas integruoja žmogaus patirtį bei nuovoką valdant ar modeliuojant netiesines sistemas. Pramonėje šis metodas dažniausiai naudojamas ne kaip būsenos kintamųjų įvertintojas, bet kaip reguliatorius valdantis sudėtingus netiesinius procesus. *Abdi Escalante-Sanchez* 'as su savo kolegomis panaudojo šią logiką gliukozės koncentracijos palaikymui bioreaktoriuje [27]. Žemiau esančiame paveiksle (žr. 1.3.7 pav.) pavaizduotas reguliatoriaus atsako paviršius sudarytas iš 6 taisyklių. Šios taisyklės apibūdina peristaltinės pompos veikimo būsenos priklausomybę nuo paklaidos tarp esamos koncentracijos ir nuostatos.



1.3.7 pav. *Fuzzy* logikos reguliatoriaus atsako paviršius sudarytas iš 6 taisyklių

Gliukozės supumpavimo nuostata buvo gaunama ir atnaujinama priklausomai nuo sistemoje veikiančio gliukozės koncentracijos įvertinimo. Gauti rezultatai stipriai koreliuoja tarp įvertintos bei bioproceso metu išmatuotos gliukozės koncentracijos (žr. 1.3.8 pav.). Grafikuose kvadratais pažymėtos vertės - laboratoriniai matavimai, nenutrūkstama linija žymimos vertės – įvertinti parametrai.

Neraiškiųjų aibių logika yra pranašesnė už neuroninius tinklus dėl savo paprastesnio dizaino bei galimybės lengvai modifikuoti modelį, kylant sistemos sudėtingumui [28].

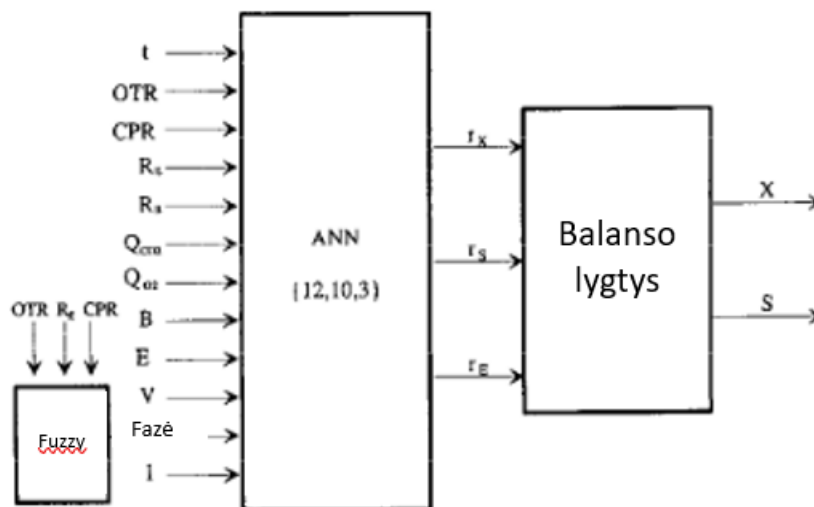


1.3.8 pav. Ištyrėtų ir įvertintų parametų grafikai. BC-1, BC-2, BC-3 skirtingi eksperimentai. A – biomasė, B – specifinis augimo greitis, C – gliukozės koncentracija.

### 1.3.3. Hibridiniai modeliai

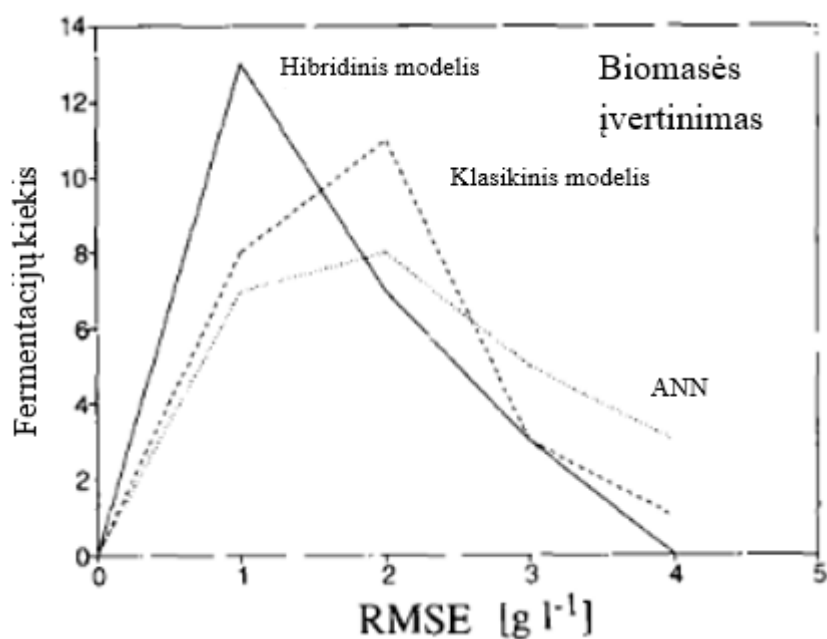
Tiek tradiciniai mechanistiniai, tiek empiriniai modeliai dažnai naudojami pramonėje, tačiau abi modelių grupės turi savų trūkumų. Juos jungiant ir naudojant kartu, atsirado nauja modelių rūšis vadinama hibridiniais modeliais. Žinomi ryšiai tarp parametų yra aprašomi klasikinių modelių, o mažiau žinomi ar sunkiai modeliuojami ryšiai yra atpažįstami empirinio modelio iš proceso duomenų bazės. Pagrindinis hibridinių modelių trūkumas - ilgesnis skaičiavimo laikas mokymo ir apskaičiavimo metu.

*Jorg'as Schubert'as*, Rimvydas Simutis ir kiti savo publikacijoje pristatė hibridinį modelį susidedantį iš dinaminių diferencialinių lygčių, dirbtinio neuroninio tinklo ir neraiškių aibių logikos [29]. Šie trys skirtingi metodai buvo naudojami būsenos kintamųjų įvertinimui, proceso valdymui bei optimizavimui.



1.3.9 pav. Hibridinio modelio struktūra

Hibridinio modelio struktūra (žr. 1.3.9 pav.) vaizduoja šių modelių parametrus bei tarpusavio ryšius. Dirbtinis neuroninis tinklas naudoja 12 įėjimo kintamųjų būsenos kintamiesiems įvertinti. Vienas iš šių parametru yra gaunamas ne tiesiai iš bioproceso, bet generuojamas mažos neraiškių aibių logikos sistemos. Dirbtinio neuroninio tinklo generuojami būsenos kintamieji yra naudojami balanso lygtyse. Šios lygtys aprašo priklausomybę tarp būsenos kintamųjų ir biomasės bei gliukozės koncentracijos. Šioje sistemoje *Fuzzy* logikos modelis yra skirtas bioproceso būsenai įvertinti. Šis papildomas reguliatorius suteikia informacijos apie dinamikos pasikeitimą procese. Informacija apie ląstelių augimo stadijos pasikeitimą leidžia ANN tiksliau įvertinti vykstančius procesus. Biomasės įvertinimo vidutinio kvadratinio nuokrypio priklausomybė nuo naudoto modelio rūšies atvaizduota 1.3.10 paveiksle.



1.3.10 pav. Modelių naudotų biomasės įvertinimui palyginimas

Hibridiniai yra pranašiausi modeliai bioproceso įvertinimui. Jie sujungia gerąsias mechanistinių ir empirinių modelių savybes. Jie naudoja informaciją apie proceso struktūrą, ryšius tarp parametru. Empirinė modelio dalis įvertina dar nežinomus, sudėtingus ar mažai ištirtus ryšius.

## 2. Metodinė dalis

Tyrimo metu visi skaičiavimai bei modeliai buvo aprašyti naudojant *Python* programavimo kalbą ir naudojantis *Visual Studio 2019* programavimo įrangos paketu. Neuroniniai tinklai sudaryti naudojantis *Tensorflow*, *Keras* bibliotekomis.

### 2.1. Eksperimentų sąlygos ir duomenys modeliui

Tiriamąjį darbo metu buvo dirbama su rekombinantinėmis *E.coli BL21(DE3) pET21-IFN-alfa-5* ląstelėmis. Duomenims surinkti atlikti 17 eksperimentų. Eksperimentų metu buvo naudojamas periodinis su pamaitinimu procesas. Dalyje eksperimentų buvo palaikomas ištirpusios gliukozės kiekis artimas 0 tai yra buvo atliekami limituoti eksperimentai. Šių eksperimentų auginimo terpė buvo vienoda ir ją sudarė 46.55 g KH<sub>2</sub>PO<sub>4</sub>, 14 g (NH<sub>4</sub>)<sub>2</sub>HPO<sub>4</sub>, 5.6 g C<sub>6</sub>H<sub>8</sub>O<sub>7</sub>.H<sub>2</sub>O, 3 mL koncentruota medžiaga skirta putojimo mažinimui, 35 g H<sub>14</sub>MgO<sub>11</sub>S, ir 105 gD (+) gliukozės monohidrato. Šių ląstelių kultivavimo metu buvo panaudotas *IPTG* preparatas skirtas tikslinio produkto sintezės aktyvavimui. Žemiau pateiktos eksperimentų metu bioreaktoriuje palaikytos sąlygos:

1 lentelė. Eksperimento metu palaikytos sąlygos

Sąlygos	Kultūra	Komentaras
Bioreaktoriaus tūris	7 L	-
Proceso tipas	Periodinis su pamaitinimu	-
pH nuostatas	6.8	Nuostatas palaikytas PID reguliatoriumi tiekiant NaOH (natrio šarmas)
Tiekiamo oro srautas	1,75–3,75 L/min	Papildomai tiekiamas grynas deguonis 0-7,5 l/min srautu siekiant padidinti deguonies pernešimą
Temperatūra	37 °C	Matuota PT100 temperatūros jutikliu
Maišyklės sūčiai	800-1200 RPM	-
Ištirpusio deguonies nuostatas	20 %	Matuota <i>Ingold</i> jutikliu ( <i>Mettler Toledo</i> )
Išėjimo dujų matavimai	Deguonies, anglies dioksido koncentracija	<i>BlueSens</i> dujų analizatoriumi ( <i>BCpreFerm, BlueSens, Herten, Germany</i> )

Eksperimentų metu buvo paimti 287 mėginiai. Šie mėginiai buvo naudojami acetatų, biomasės bei ištirpusios gliukozės koncentracijoms išmatuoti. Biomasė ir gliukozė suteikia modeliui informacijos apie ląstelių vystymąsi bei augimą. Realiu laiku buvo renkama informacija apie deguonies suvartojimo greitį (angl. *Oxygen uptake rate* OUR), anglies išsiskyrimo greitį (angl. *Carbon production rate* CPR), bioreaktoriaus tūrį, sumaitintą gliukozės kiekį bei indukcijos laiką. OUR ir CPR teikia svarbią informaciją apie ląstelės augimo būseną bei greitį [30,31]. Supumpuota gliukozė lemia biomasės augimą, o ištirpusi gliukozė ir tūris nusako apie bioreaktoriaus terpės skiedimą, kuris atsiranda dėl maitinimo. Indukcija nusako, kada buvo panaudotas *IPTG* preparatas, kuris turi didelę

įtaką proceso dinamikai dėl pasikeitusių biocheminių procesų [32,33]. Šių matavimų vienetai parodyti 2 lentelėje.

**2 lentelė.** Acetatų įvertinimui naudoti kultivavimo parametrai

Parametras	Matavimo vienetai
Laikas (nuo proceso pradžios)	h
Biomasė	g/L
Tūris	kg
Anglies išsiskyrimo greitis (angl. <i>Carbon production rate (CPR)</i> )	g/(kg*h)
Ištirpusi gliukozė	g/L
Deguonies suvartojimo greitis (angl. <i>Oxygen uptake rate (OUR)</i> )	g/(kg*h)
Sumaitinamos gliukozės kiekis	g/L
Indukcija	Konstanta (0 arba 1)

## 2.2. Pasikliautinumo intervalai

Prieš matematinių modelių sudarymą, yra sudaromi ir apskaičiuojami acetatų koncentracijų pasikliautinumo intervalai. Pasikliautinumo intervalas yra statistinė riba, kuri įvertina tikėtiną reikšmių intervalą, remiantis duomenų imtimi. Tai svarbus įrankis, padedantis įvertinti acetatų koncentracijų neapibrėžtumą. Pasikliautinumo intervalai dažniausiai skaičiuojami naudojant Stjudento arba normalųjį (Gauso) skirstinį. Gauso skirstinys yra naudojamas tuo atveju, kai yra tiriami dideli duomenų kiekiai. Šiuo atveju yra naudojamas Stjudento dar vadinamas t-skirstinys.

$$PI = \mu \pm \frac{t(n,P)*s}{\sqrt{n}}, \quad (4)$$

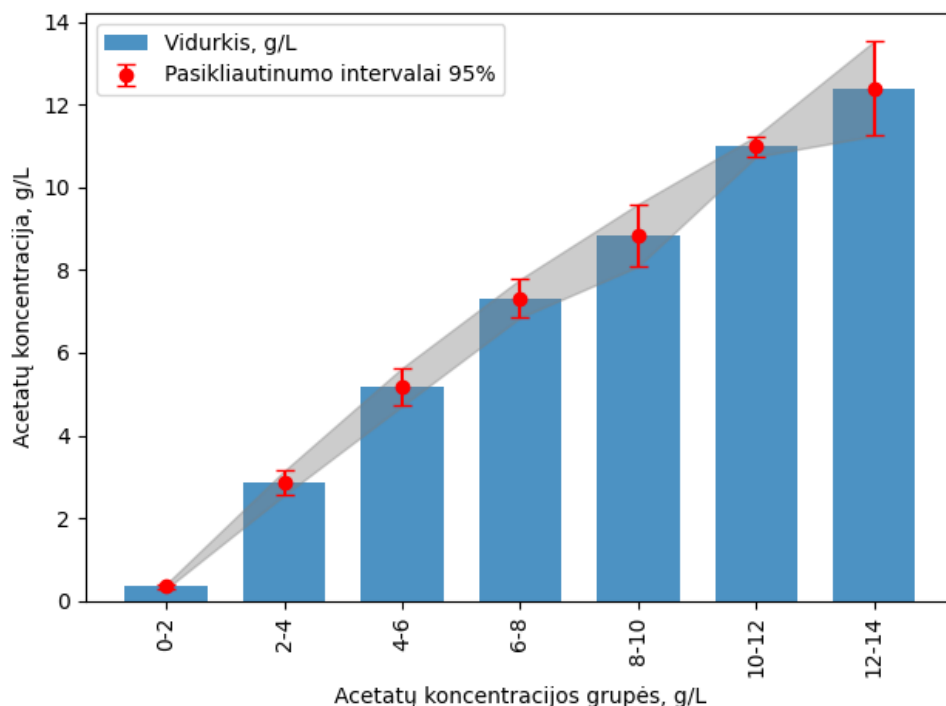
čia  $\mu$  - grupės vidurkis,  $t$  – Stjudento vertė priklausanti nuo  $n$  – verčių kiekio, ir  $P$  pasikliautinumo lygis,  $s$  – standartinis nuokrypis.

Šis skirstinys leidžia apibrėžti pasikliautinumo intervalą, atsižvelgiant į pasikliautinumo lygį ir laisvųjų laipsnių skaičių tai yra duomenų kiekį, kuris yra naudojamas pasikliautinumui suskaičiuoti. Prieš pasikliautinumo intervalų skaičiavimą, acetatų koncentracijos yra sugrupuojamos į grupes po 2 g/L. Pasikliautinumo intervalai yra skaičiuojami šioms grupėms kartu su vidutine grupių verte. Grupės, intervalai ir vidurkiai pateikiami 3 lentelėje bei 2.2.1 paveiksle.

**3 lentelė.** Acetatų koncentracijų grupės, pasikliautinumo intervalai ir vidurkiai

	0-2 g/L	2-4 g/L	4-6 g/L	6-8 g/L	8-10 g/L	10-12 g/L	12-14 g/L
<b>Apatinė riba</b>	0,2956	2,5559	4,7159	6,8548	8,0894	10,7473	11,2498
<b>Viršutinė riba</b>	0,4121	3,1458	5,6377	7,7832	9,6027	11,2471	13,54
<b>Skirtumas</b>	0,1165	0,5899	0,9218	0,9284	1,5133	0,4998	2,2902
<b>Vidurkis</b>	0,3524	2,8513	5,1764	7,3188	8,846	10,9972	12,3949

Šioje lentelėje bei grafike pateikti pasikliautinumo intervalai yra sudaryti naudojant 0,95 pasikliautinumo lygį tai yra maždaug 95% visų atsitiktinai paimtų acetatų koncentracijų vidurkis atitinkamose grupėse pateka į apskaičiuotus režius. Šis pasikliautinumo intervalas padeda geriau suprasti tiek naudojamų duomenų pasiskirstymą bei triukšmą, tiek vėliau spręsti apie matematinio modelio efektyvumą.



2.2.1 pav. Pasikliautinumo intervalo grafikas

### 2.3. Eksperimento duomenų apdirbimas

Tyrimo metu surinkti duomenys buvo apdoroti naudojant dvi matematinės technikas skirtas duomenų analizei: atskirosios reikšmės skaidymą ir principinę komponentų analizę. Šios technikos pirmiausiai buvo apskaičiuotos Excel programinėje įrangoje. Tai buvo atlikta dėl metodų principinio veikimo išsiaiškinimo. Vėliau metodai aprašyti naudojant *Python* programavimo kalbą. Šių metodų apskaičiavimui yra naudojama matricų daugyba.

#### 2.3.1. Atskirosios reikšmės skaidymas

Atskirosios reikšmės skaidymas (angl. *Singular value decomposition* SVD) yra matematinė technika, plačiai naudojama duomenų analizėje ir kitose srityse. Ši technika dažnai naudojama tokioms užduotims kaip duomenų glaudinimas, matricų matmenų mažinimas ir didelių bei sudėtingų duomenų rinkinių analizė. SVD tai matricos skaidymas į daugiklius (angl. *factorization*) pasitelkiant tiesinės transformacijos metodus [34]. Originali matrica  $C$  yra suskaidoma į 3 dalis: dešinįjį vektorių ( $U$ ), kairįjį vektorių ( $V^T$ ) ir įstrižinę matrica  $\Sigma$  :



$$C_{m \times n} = U_{m \times r} \times \sum_{r \times r} \times V^1_{r \times n}$$

2.3.1 pav. Atskirosios reikšmės skaidymas

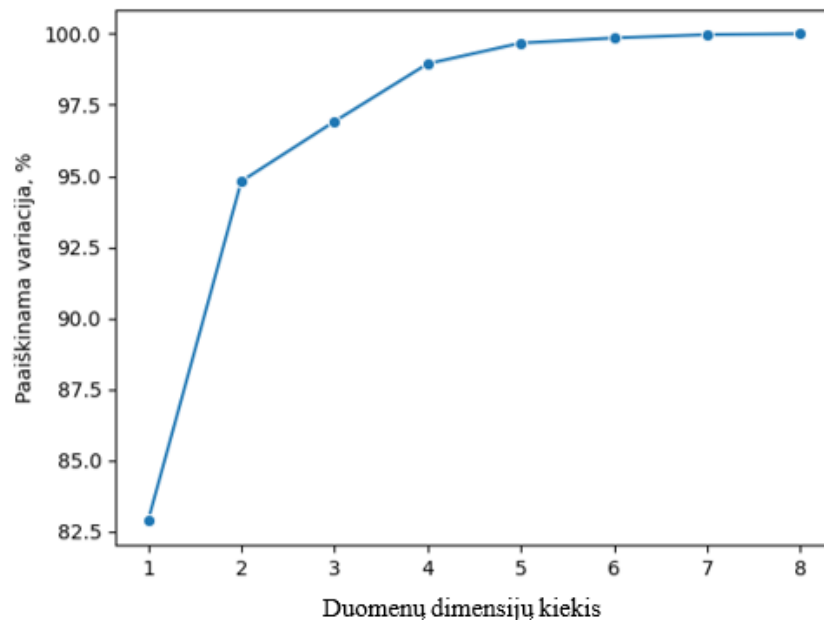
$\Sigma$  matricą sudaro vertės gautos ištraukus šaknį iš matricos skaidymo metu gautų tikrinių / latentinių reikšmių (angl. *eigenvalues*). Sudauginus šias naujas matricas gaunama originali matricą. Šis skaidymas naudojamas duomenų modeliams ir ryšiams identifikuoti, duomenų sudėtingumui sumažinti ir svarbioms ypatybėms bei modeliams išgauti, atliekant tokias užduotis kaip klasifikavimas, įvertinimas. Atskirosios reikšmės skaidymo skaičiavimo Excel dalis pateiktas 2.3.2 paveiksle.

Originalūs duomenys (A)							Transponuoti originalūs duomenys (AT)							
Time	Biomass	Protein	Glucose	BioWeight	OUR	Feed	Co2							
0	0.1324	0	4.9002	3.746	0.027249	0	0.148613	0.1324	0.118	0.146933	0.2126	0.32664	0.446867	0.505543
1	0.118	0	2.36905	1.9645	0.071065	0	0.118022	0	0	0	0	0	0	0
2	0.146933	0	1.3752	1.302333	0.154578	0	0.138998	4.9002	2.36905	1.3752	0.833225	0.33148	0.016667	0.014286
3	0.2126	0	0.833225	0.972	0.312685	0	0.213031	3.746	1.9645	1.302333	0.972	0.7726	0.639833	0.548571
4	0.32664	0	0.33148	0.7726	0.572414	0	0.31141	0.027249	0.071065	0.154578	0.312685	0.572414	0.838053	0.852865
5	0.446867	0	0.016667	0.639833	0.838053	0.116629	0.385353	0	0	0	0	0	0.116629	1.71126
6	0.505543	0	0.014286	0.548571	0.852865	1.71126	0.327793	0.148613	0.118022	0.138998	0.213031	0.31141	0.385353	0.327793
7	0.6089	0	0.0125	0.47975	1.213896	3.980918	0.409611	Originaliųjų ir transponuotųjų duomenų matricų daugyba (AT*A)						
8	0.804	0	0.011111	0.428	1.77241	6.722267	0.43067	2156.292	627.2912	392.1568	230.8092	6314.42	40734.65	430.5
9	0.96184	0	0.007692	0.307231	6.721324	28.13154	0.664651	627.2912	916.6226	28.64276	49.26912	1711.883	11999.95	109.8805
10	1.308618	0	0.009091	0.344364	3.554339	14.60617	0.425696	392.1568	28.64276	5707.161	1291.153	994.5899	5496.588	161.5121
11	1.6697	0	0.008333	0.319083	4.740009	20.42361	0.553936	230.8092	49.26912	1291.153	388.551	637.7139	3409.911	77.11322
12	2.053631	0	0.007692	0.307231	6.721324	28.13154	0.664651	6314.42	1711.883	994.5899	637.7139	19620.42	114095.3	1320.21
13	2.620114	1.040714	0.007143	0.308643	9.911721	38.81299	0.810102	40734.65	11999.95	5496.588	3409.911	114095.3	840564	7352.482
14	3.330133	0.862	0.006667	0.291667	12.11964	52.13415	0.836381	430.5	109.8805	161.5121	77.11322	1320.21	7352.482	100.1282
15	3.39675	1.013125	0.00625	0.286	9.673854	60.27063	0.679119	Latentinės vertės AT*A						
16	3.327529	1.129412	0.005882	0.276941	8.930481	65.42635	0.645155	858398.3	6012.7	4177.499	743.715	88.47	28.955	3.538
0	0.1644	0	4.8102	3.751	0.069796	0	0.212329	Matrica E						
1	0.1298	0	2.315	1.85	0.091907	0	0.113572	926.4978	0	0	0	0	0	0
2	0.156	0	1.2851	1.224667	0.181876	0	0.130684	0	77.5416	0	0	0	0	0
3	0.2276	0	0.792675	0.9125	0.354199	0	0.203646	0	0	64.63357	0	0	0	0
4	0.34344	0	0.29186	0.7254	0.616944	0	0.296008	0	0	0	27.27114	0	0	0
5	0.483733	0	0.016667	0.602	0.907773	0.228476	0.370713	0	0	0	0	9.405849	0	0
6	0.556514	0	0.014286	0.515714	1.013865	2.16886	0.345854	0	0	0	0	0	5.380985	0
7	0.75	0	0.0125	0.452125	1.524459	4.807031	0.430351	0	0	0	0	0	0	1.880957
8	0.944267	0	0.011111	0.403	5.672137	8.105341	0.313279	0	0	0	0	0	0	0
9	1.25504	0	0.01	0.3601	3.283994	12.38421	0.374359	Latentinis vektorius AT*A						
10	1.653636	0	0.009091	0.330727	4.750272	18.06937	0.51302	5.52	1.502	2.471	-14.562	1.37	2.802	-0.385
11	2.2583	0	0.008333	0.315917	7.07587	25.73648	0.645654	1.625	-0.233	0.358	-397.392	-0.14	-0.107	0.018
12	2.838985	0	0.007692	0.303692	10.47931	36.15613	0.775655	0.754	33.473	-2.351	-2.757	-1.253	0.149	0.008
13	3.336857	0.000714	0.007143	0.2965	13.54592	49.9609	0.870036	0.466	7.724	-0.124	-6.186	5.379	-0.922	-0.086
14	3.582853	0.004667	0.006667	0.294333	10.50792	60.44838	0.695471	15.498	5.466	13.345	13.399	-0.498	-0.582	-0.002
15	3.68125	0.00625	0.00625	0.283188	9.877235	66.29379	0.637486	113.584	-1.078	-1.939	4.599	-0.019	-0.061	0.01
0	0.148	0	20.5378	3.813	0.080325	0	0.254594	1	1	1	1	1	1	1
1	0.1292	0	9.9086	1.878	0.09881	0	0.12391							
2	0.155733	0	6.425567	1.242	0.186913	0	0.128682							
3	0.2189	0	4.774125	0.925	0.356935	0	0.204859	64.63357	27.27114	9.405849	5.380985	1.880957		
4	0.34464	0	3.60312	0.7354	0.674371	0	0.318749							
5	0.570267	0	2.49215	0.6095	1.328639	0	0.471289							
6	1.0056	0	1.366614	0.520429	2.630654	0	0.421889							
7	1.6563	0	1.918663	0.469125	4.963189	11.55979	0.673738							

2.3.2 pav. Atskirosios reikšmės skaidymo dalis Excel programinėje įrangoje

Duomenų matricos dimensijų mažinimas yra duomenų rinkinio kintamųjų skaičiaus mažinimo procesas, išsaugant kuo daugiau informacijos. Gautas latentinės vertės yra išrikiuojamos nuo didžiausios iki mažiausios. Dimensijos esančios rikiuotės pabaigoje nustatomos į 0, o reikiamas kiekis paliekamas. Naudojant gautas naujas matricas yra atliekama originalios matricos

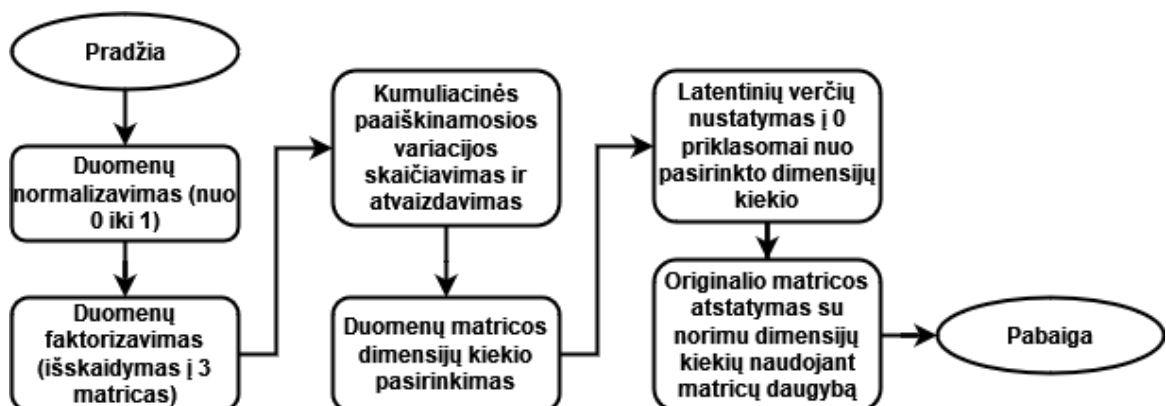
rekonstrukcija t. y. matricos yra sudauginamos. Dimensijų kiekio pasirinkimas priklauso nuo norimos išsaugoti informacijos kiekio. Šis kiekis yra pasirenkamas iš patirties ar naudojantis paaiškinamąją variaciją (žr. 2.3.3 pav.). Paaiškinamoji variacija, tai visos duomenų variacijos dalis išreikšta procentais. Paaiškinamoji variacija yra apskaičiuojama naudojantis latentines vertes [35].



2.3.3 pav. SVD kumuliacinė paaiškinamoji variacija

Prieš atliekant SVD skaičiavimus duomenys normalizuoti t. y. transformuoti į vertes nuo 0 iki 1. Gauta paaiškinamoji variacija parodo, kad naudojant 4 matricos dimensijas yra gaunamas didesnis nei 95% paaiškinamumas. 6 matricos dimensijos nusako beveik 100% duomenų variacijos. Esant stipriai koreliacijai tarp duomenų latentinės vertės bus mažesnės, ir variacijai paaiškinti prireiks mažiau dimensijų. Praktikoje reikia atlikti keletą papildomų testų bei įvertinimų, norint tiksliai nustatyti reikiamą kiekį dimensijų.

Duomenų dimensijų kiekis pasirenkamas pagal latentines vertes. Nenaudojamos latentinės vertės yra nustatomos į 0. Po šio nustatymo nauja duomenų matrica yra apskaičiuojama sudauginus dešiniąjį, kairinį vektorius su modifikuota įstrižine (latentinių verčių) matrica. Taip gaunama matrica su nustatytu kiekiu dimensijų. Atskirosios reikšmės skaidymo skaičiavimo algoritmo schema pateikta 2.3.4 paveiksle, o dalis *Python* kodo 2.3.5 pav. Duomenų faktorizavimui naudojama *Numpy* biblioteka.



2.3.4 pav. Atskirosios reikšmės skaidymo skaičiavimo algoritmo schema

```

matrix = np.array(matrix)
U, s, VT = np.linalg.svd(matrix)

# Compute the sum of the squares of all singular values
sum_squares_all = np.sum(s**2)

# Compute the proportion of variance retained for each value of k
k_values = range(1, len(s)+1)
proportions_retained = [(np.sum(s[:k]**2) / sum_squares_all)*100 for k in k_values]
array = np.transpose(np.array(proportions_retained))

def SVD_reconstruction(U, Sigma, VT, n_elements):
    U1 = np.array(U)
    Sigma1 = np.array(Sigma)
    VT1 = np.array(VT)

    #Sigma1 = Sigma1[:, :n_elements]
    Sigma1 = np.diag(Sigma1[:n_elements])
    VT1 = VT1[:n_elements, :]
    U1 = U1[:, :n_elements]
    reduced_matrix = U1.dot(Sigma1.dot(VT1))

    return reduced_matrix[:, :n_elements]

```

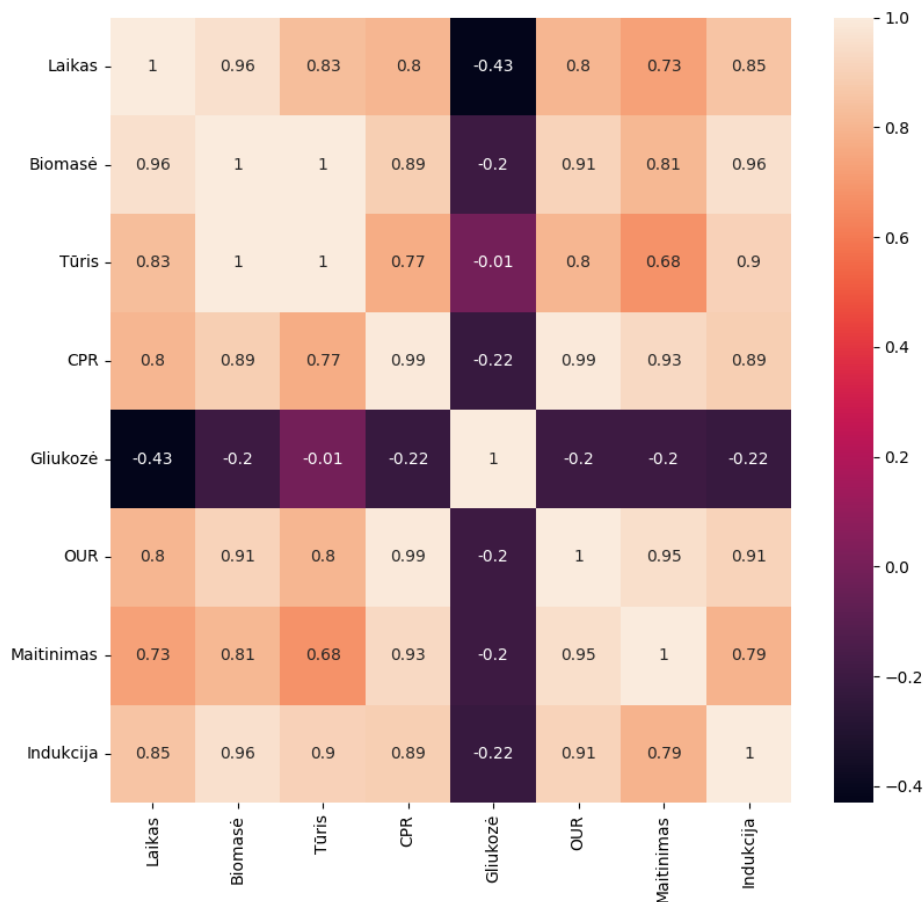
2.3.5 pav. SVD skaičiavimo ir rekonstrukcijos Python kodas

## 2.4. Principinės komponentės analizė

Principinės komponentės analizė (angl. *Principal component analysis* PCA) yra SVD tęsinys [34]. Atskirosios reikšmės skaidymas yra atliekamas naudojant originalią duomenų matricą. Principinės komponentės analizė tai kovariacijos matricos dekompozicija [36]. SVD ištraukia duomenis į ta pusę, kuri turi didžiausią variaciją. PCA duomenis transformuoja taip, kad pirmoji komponentė (matricos dimensija) nusakytų didžiausią variacijos kiekį. Prieš panaudojant PCA duomenis turi būti standartizuoti [37].

$$Z = \frac{X - \mu}{\sigma}, \quad (5)$$

čia  $Z$  – standartizuota vertė,  $X$  – pradinė vertė,  $\mu$  - stulpelio vidurkis,  $\sigma$  – standartinis nuokrypis. Atlikus duomenų standartizavimą yra atliekamas kovariacijos skaičiavimas (žr. 2.4.1 pav.).



2.4.1 pav. Kovariacijų matrica

Kovariacija tai dydis tarp dviejų verčių, kuris parodo priklausomą kintamumą. Jei vertės didėja arba mažėja kartu, kovariacija - teigiama, jei viena didėja, kita mažėja - neigiama. Jei kovariacija yra lygi 0 duomenys yra statistiškai nepriklausomi. Pavyzdžiui iš kovariacijų matricos galime teigti, kad biomasė stipriai koreliuoja su laiku, tačiau turi neigiamą koreliaciją su gliukozės kiekiu - bėgant laikui gliukozės kiekis mažėja. Principinės komponentės analizės Excel skaičiavimo dalis pateikta 2.4.2 paveiksle. PCA skaičiavimo metu yra gaunamos latentinės vertės ir tikriniai vektoriai (angl. Eigenvector). Latentinės vertės kaip ir SVD atveju yra naudojamos kumuliatyviniam paaiškinamosios variacijos skaičiavimui (žr. 2.4.3 pav.).

Kovariacija								
	Time	Biomass	Protein	Glucose	BioWeight	OUR	Feed	Co2
Time	1	0.865962768	0.381121466	-0.584848218	-0.702014486	0.784150919	0.83185579	0.722488635
Biomass	0.865962768	1	0.349238025	-0.402983935	-0.519319588	0.937838446	0.93910445	0.847648193
Protein	0.381121466	0.349238025	1	-0.151461011	-0.172569982	0.282438899	0.327339029	0.215330696
Glucose	-0.584848218	-0.402983935	-0.151461011	1	0.846860848	-0.397283988	-0.318214988	-0.455874487
BioWeight	-0.702014486	-0.519319588	-0.172569982	0.846860848	1	-0.507440149	-0.425880753	-0.568107959
OUR	0.784150919	0.937838446	0.282438899	-0.397283988	-0.507440149	1	0.80388276	0.926197378
Feed	0.83185579	0.93910445	0.327339029	-0.318214988	-0.425880753	0.80388276	1	0.655991082
Co2	0.722488635	0.847648193	0.215330696	-0.455874487	-0.568107959	0.926197378	0.655991082	1

Latentinės vertės								
	5.203	1.216	0.882	0.393	0.155	0.097	0.042	0.11
1.065	-0.283	-0.297	-0.56	5.026	6.401	-0.099	0.097	
1.084	3.202	0.426	-0.16	-1.691	-1.47	0.106	-4.734	
0.445	3.262	-3.719	0.35	-0.173	-0.571	0.032	0.172	
-0.719	8.567	0.508	0.008	9.19	-1.304	-0.029	-0.018	
-0.836	7.397	0.28	0.032	-7.745	4.229	0.103	-0.009	
1.048	2.765	0.82	0.511	-1.479	-0.445	-1.314	1.239	
0.983	3.889	0.255	-0.9	-2.814	-2.956	0.424	2.678	
1	1	1	1	1	1	1	1	1

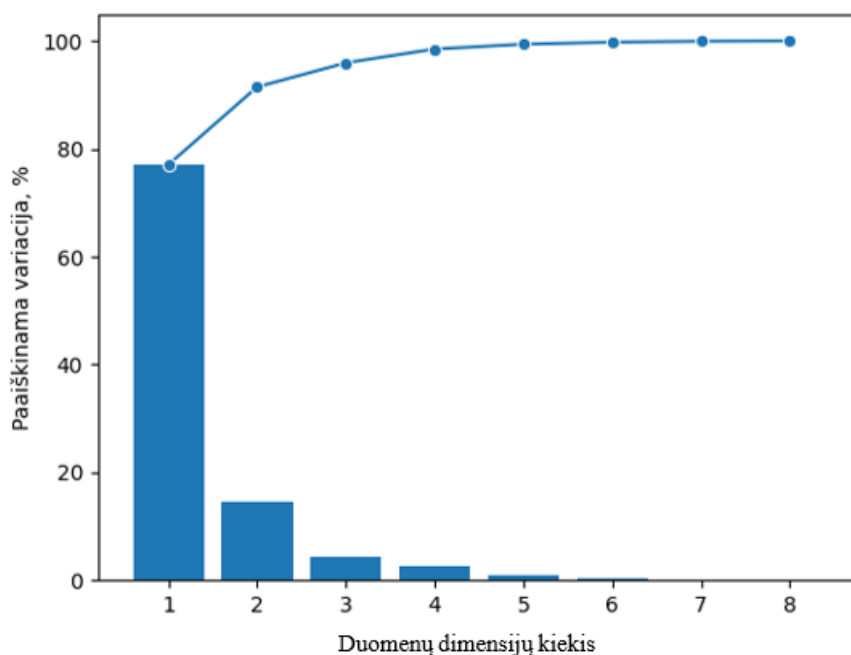
  

PCA procentas							
PCA1	PCA2	PCA3	PCA4	PCA5	PCA6	PCA7	PCA8
64%	15%	11%	5%	2%	1%	1%	1%
64%	79%	90%	95%	97%	98%	99%	100%

Transponuoti latentiniai vektoriai							
	1.065	1.084	0.445	-0.72	-0.836	1.048	0.983
1	1.065	1.084	0.445	-0.72	-0.836	1.048	0.983
-0.283	-0.283	3.202	3.262	8.567	7.397	2.765	3.889
-0.297	-0.297	0.426	-3.719	0.508	0.28	0.82	0.255
-0.561	-0.561	-0.163	0.35	0.008	0.032	0.511	-0.899
5.026	5.026	-1.691	-0.173	9.19	-7.745	-1.479	-2.814
6.401	6.401	-1.47	-0.571	-1.3	4.229	-0.445	-2.956
-0.099	-0.099	0.106	0.032	-0.03	0.103	-1.314	0.424
0.097	0.097	-4.734	0.172	-0.02	-0.009	1.239	2.678

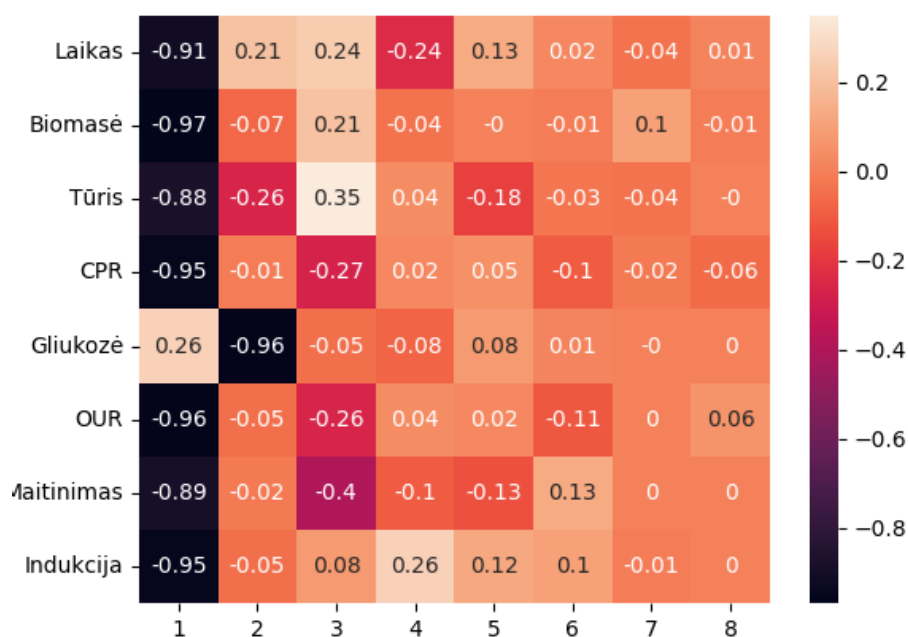
2.4.2 pav. Principinės komponentės analizės dalis Excel programinėje įrangoje



2.4.3 pav. PCA kumuliacinė paaiškinamoji variacija

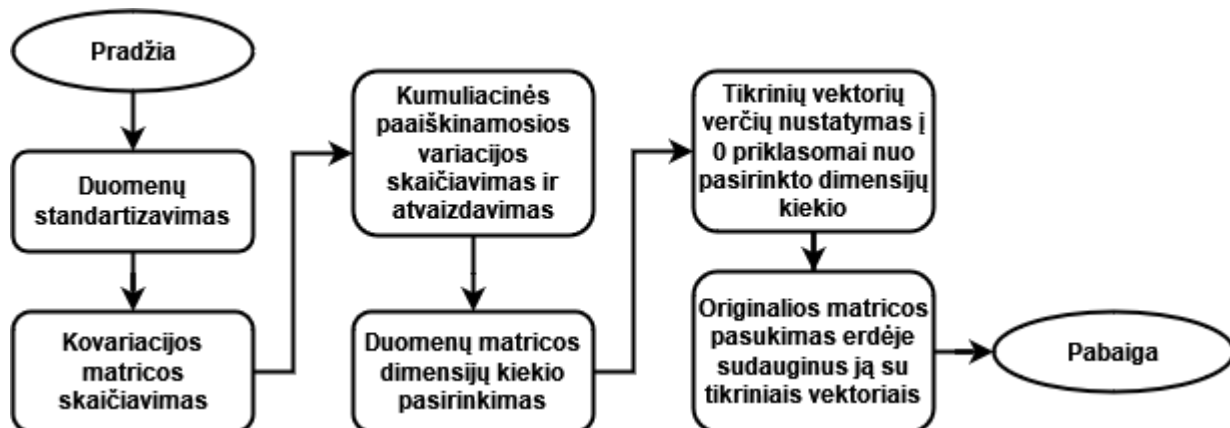
4 PCA matricos dimensijos apibūdina daugiau nei 95% visos duomenų variacijos. PCA metodu transformuoti duomenys saugo kitokią informaciją nei SVD, tai galima spręsti iš kumuliatyvinės variacijos grafikų. Transformuotus duomenis skaityti yra sudėtinga, nes jie nebeišlaiko originalių duomenų tendencijų, tačiau apskaičiavus koreliaciją tarp originalių duomenų ir PCA komponentių yra matomos priklausomybės.

Iš koreliacijų matricos (žr. 2.4.4 pav.) matyti, kad visi komponentai, išskyrus gliukozę turi didelę įtaką pirmajai komponentei. Tai parodo, kad visi juodai pažymėti duomenys yra stipriai koreliuoti tarpusavyje. Didžiausia priklausomybė yra tarp antrosios komponentės ir gliukozės. Tai parodo, kad 2 komponentė teikia informaciją apie gliukozės kiekį. Ši matrica taip pat patvirtina variacijos grafiką ir parodo, kad pirmosios 3 komponentės teikia apie 92% visos informacijos.



2.4.4 pav. Koreliacijų matrica tarp originalių duomenų ir PCA komponentių

Principinės komponentės matricos dimensijų kiekis yra nustatomas nereikalingų tikrinių vektorių vertes nustačius į 0. Šiuo atveju nėra daroma originalios matricos rekonstrukcija kaip su SVD metodu, o yra daromas duomenų matricos pasukimas. Tai atliekama sudauginus originalią duomenų ir tikrinių verčių matricas. Ši daugyba pasuka duomenis geometriškai ir pateikia juos iš kitos perspektyvos. Principinės komponentės analizės skaičiavimo algoritmo schema pateikta 2.4.5 paveiksle.



2.4.5 pav. Principinės komponentės analizės skaičiavimo algoritmo schema

2.4.6 paveiksle pavaizduota pagrindinė dalis *Python* kodo naudoto PCA skaičiavimui ir originalių duomenų pasukimui. Kovariacijų matrica ir latentinės vertės su tiksliniais vektoriais apskaičiuoti naudojantis *Numpy* biblioteka. Po kovariacijų matricos išskaidymo latentinės vertės ir tikriniai vektoriai yra išrikiuojami latentinių verčių mažėjimo tvarka.

```

def PCA(arr):
    array = np.copy(arr)
    covariance_matrix = np.cov(array.T)

    eigen_values, eigen_vectors = np.linalg.eig(covariance_matrix)

    idx = eigen_values.argsort()[::-1]
    eigen_values = eigen_values[idx]
    eigen_vectors = eigen_vectors[:,idx]

    variance_explained = []
    for i in eigen_values:
        variance_explained.append((i/sum(eigen_values))*100)
    cumulative_variance_explained = np.cumsum(variance_explained)

    cumulative_variance_explained = np.transpose(cumulative_variance_explained)
    array = np.transpose(np.array(variance_explained))
    k_values = range(1, len(cumulative_variance_explained)+1)
  
```

2.4.6 pav. PCA skaičiavimo ir rekonstrukcijos *Python* kodas 1

```

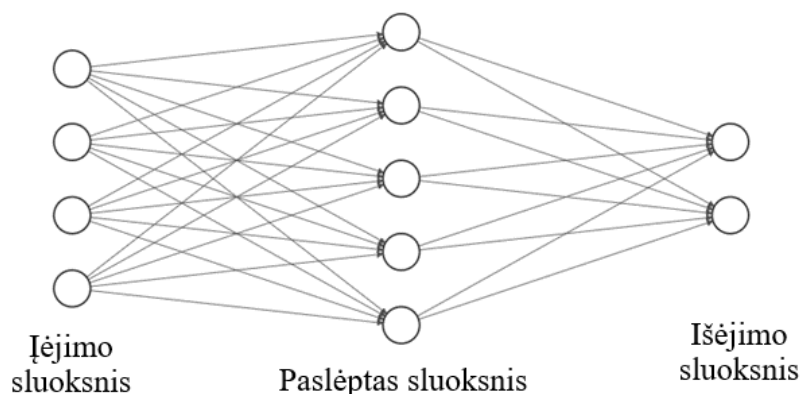
def PCA_reconstruction(data, eigenvector, dim):
    eigenvectorq = np.copy(eigenvector)
    if dim<=8 and dim>=1:
        eigenvectorq = eigenvector[:, :dim]
    return np.dot(data,eigenvectorq)[:,:dim]

```

2.4.7 pav. PCA skaičiavimo ir rekonstrukcijos Python kodas 2

## 2.5. Dirbtiniai neuroniniai tinklai

Neinvaziniam metabolitų kaupimuisi įvertinti pritaikant SVD bei PCA metodus buvo pasirinktas empirinis modelis, paremtas grįžtamoju dirbtiniu neuroniniu tinklu. Neuroninis tinklas susideda iš neuronų, kurie gauta informacija transformuoja naudojant tam tikrą matematinę funkciją bei svorinius koeficientus. ANN sluoksniai susideda iš  $n$  skaičiaus neuronų. Paprasčiausio neuroninio tinklo struktūra susideda iš vieno neuronų sluoksnio. Praktikoje labiau paplitusi daugiasluoksnė sistema susidedanti iš įėjimo, išėjimo bei paslėptų sluoksnų. Daugiasluoksnė struktūra leidžia tinklą pritaikyti klasifikavimo, regresijos ar prognozavimo uždaviniams spręsti.

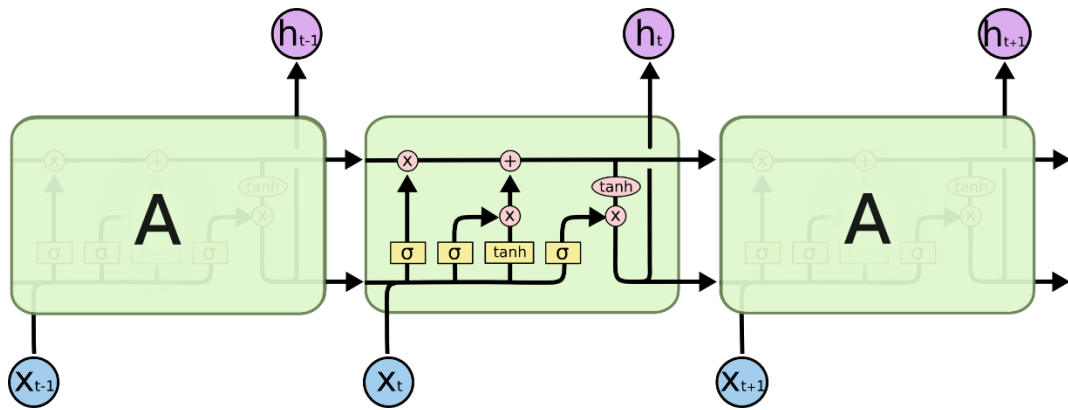


2.5.1 pav. Daugiasluoksnio tinklo pavyzdys, sudarytas iš keturių įėjimo neuronų, penkių paslėpto sluoksnio neuronų ir dvejų išėjimo neuronų.

Acetatų estimavimui pasirinktas tiesinio sklidimo daugiasluoksnis tinklas (angl. *Multilayer Feed-forward Neural Network*, MFFN), kurio visi signalai yra susieti tiesiškai ir viena kryptimi. Ši struktūra dažnai pasirenkama modeliuoti netiesiniams ryšiams. Tačiau, be MFFN tinklo, buvo išbandytas ir rekurentinis ilgalaikis trumpalaikės atminties neuroninis tinklas (angl. *Long short-term neural network*, LSTM).

Skirtingai nei MFFN, LSTM sugeba atsiminti ankstesnes reikšmes, kurios yra būtinos numatant sekantį laiko momentą. LSTM tinklas turi tris pagrindinius modulius: pamiršimo modulį, įvesties modulį ir išvesties modulį. Pirmasis modulis skirtas pamiršti informaciją iš praeities, kuri jau nėra aktuali dabartiniam uždaviniui. Įvesties modulis skirtas priimti naują informaciją ir pridėti ją prie atminties, o išvesties modulis skirtas generuoti prognozes naudojant informaciją iš atminties. Vienas iš pagrindinių LSTM tinklo privalumų yra jo gebėjimas modeliuoti ilgalaikę priklausomybę tarp duomenų, kas yra svarbu, kai reikia analizuoti laiko eilutes. Struktūrinė LSTM schema pateikta 2.5.2 pav.





2.5.2 pav. Struktūrinė LSTM schema [38]

Čia nauja informacija pateikiama neuroniniam tinklui ( $X_t$ ), o prognozė ( $h_t$ ). Dėl rekurentinės (angl. *recurrent*) šio modelio struktūros į sluoksnį yra pateikiama ne tik nauja informacija, bet ir informacija iš prieš tai buvusių sluoksnių.

### 2.5.1. Aktyvavimo funkcijos

Aktyvavimo funkcijos tai matematinės funkcijos, kurios transformuoja duomenis neurono viduje [39,40]. Šios funkcijos būna 3 rūšių:

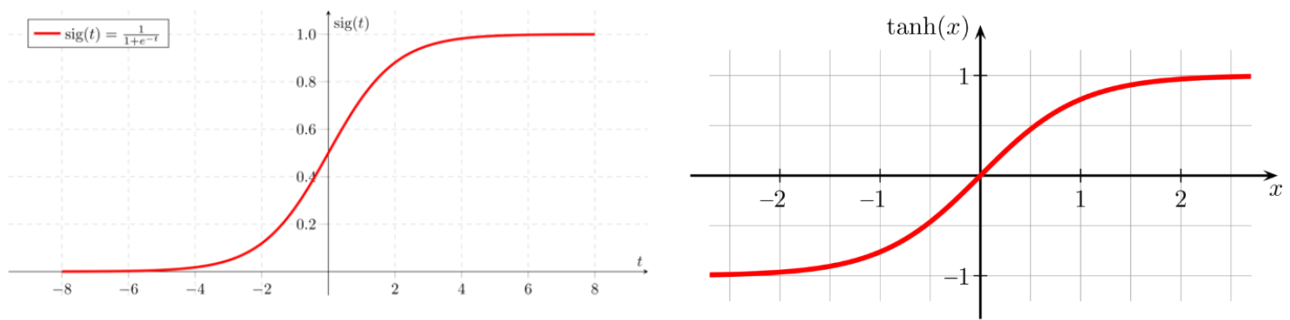
- Teisinės – neurono išėjimo rezultatas tiesiškai priklauso nuo įėjimo svorių sumos;
- Slenkstinės – išėjimo rezultatas priklauso nuo to, ar įėjimo svorių suma viršija slenkstinę vertę;
- Netiesinės – išėjimo rezultatas yra netiesiškai proporcingas įėjimui.

Dirbtiniuose neuroniniuose tinkluose dažniausiai naudojamos netiesinės aktyvavimo (perdavimo) funkcijos. Jos leidžia tinklui sudaryti sudėtingesnes duomenų struktūras. Praktikoje dažniausia sutinkamos 3 netiesinės funkcijos: sigmoidinė (angl. *sigmoid*), hiperbolinio tangento (angl. *tanh*) ir ištaisyta linijinė (angl. *rectified linear activation*, arba *ReLU*).

Sigmoidinė funkcija, dar vadinama logistine funkcija, padeda normalizuoti visus įėjimus ir gražina vertę nuo 0 iki 1 (žr. 2.5.3 pav.). Ši funkcija padeda įėjimo signalus, kurie gali varijuoti nuo  $-\infty$  iki  $+\infty$ , apriboti. Šis apribojimas ypač naudingas klasifikavimo uždaviniuose, tačiau ši savybė taip pat turi nykstančio gradiento problemą. Kuo didesnis / mažesnis įėjimo signalas tuo didesnė tikimybė, kad išėjimo signalas bus nykstančiai arti 0 arba 1. Ši problema riboja funkcijos panaudojimą.

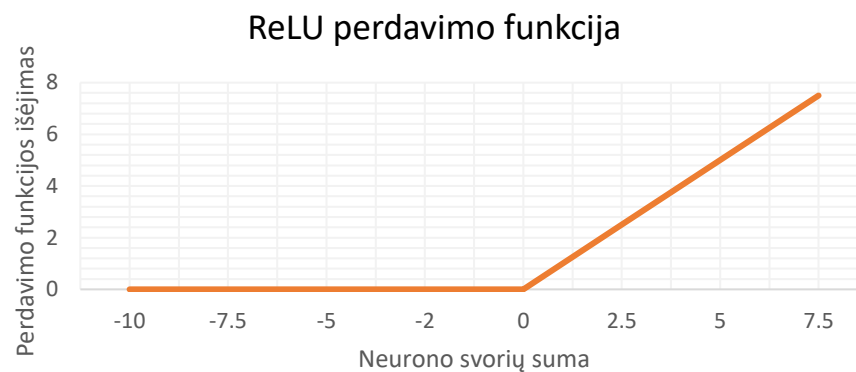
Hiperbolinio tangento funkcija savo savybėmis labai panaši į sigmoidinę (žr. 2.5.3 pav.). Šios funkcijos pranašumas, kad jos išėjimo signalas turi didesnę verčių rėžį, nuo -1 iki 1. Kaip ir sigmoidinė funkcija ji turi nykstanto gradiento problemą. Ši problema ne tik sumažina naudingų įėjimo kintamųjų rėžius, bet ir labai sulėtina mokymosi procesą.





2.5.3 pav. Hiperbolinio tangento ir sigmoidinė funkcija

Šią lėto mokymosi problemą sutvarko ištaisyta linijinė funkcija – ReLu (žr. 2.5.4). Ši funkcija kitaip nei tangentinė ar sigmoidinė pasižymi tiesinės funkcijos savybėmis, kurios pagreitina mokymosi spartą. ReLu geba gražinti tikrojo 0 vertę, o ne tik jo aproksimaciją. Neigiama šios funkcijos savybė – esant neigiamoms įėjimo vertėms gražinti 0. Tai dar vadinama „mirštančios“ ReLU atveju. Dėl savo privalumų ištaisyta linijinė funkcija yra viena iš plačiausiai naudojamų aktyvavimo funkcijų.



2.5.4 pav. ReLU aktyvavimo funkcija

## 2.5.2. Mokymo algoritmas

Dirbtinis neuroninis tinklas yra apmokomas koreguojant tinklo svorius – siekiama sumažinti paklaidą tarp įvertintų ir realių mokymo duomenų. Vienas populiariausių mokymo algoritmų - klaidos atgalinis skleidimas (angl. *error back-propagation*). Šis algoritmas, skaičiuoja, neuronų svorių ir įėjimų slenkstinių verčių, gradientus pradėdamas nuo neuroninio tinklo išėjimo ir užbaigdamas įėjimu. Pasirinktas paklaidos kriterijus yra šio algoritmo tikslo funkcija priklausanti nuo tinklų apibrėžiančių parametrų. Mokymo metu yra naudojami įvairūs optimizavimo algoritmai padedantys greičiau minimizuoti tikslo funkciją keisdami jo svorius ir slenkstines vertes:

- Stochastinis gradientinis nusileidimo metodas (angl. *Stochastic gradient descent*) – šis metodas įvertina tikslo funkcijos pasikeitimus ir generuoja naują parametro vertę ta kryptimi, kurioje tikslo funkcija mažiausia. Vienas iš šito metodo minusų tai savybė „užstrigti“ lokaliame funkcijos minimume. Taip pat šis metodas yra ganėtinai lėtas. Vienas iš šio metodo pagreitinimų tai modelio mokymas naudojant ne visą informaciją iš karto, bet dalimis [41].
- Gradientinis nusileidimo metodas su momentu (angl. *Stochastic gradient descent with momentum*) – yra dar vienas iš būdų kaip pagreitinėti minėtą metodą. Mažo gradiento metu

mokymosi žingsnis labai sulėtėja. Įvedus papildomą momento skaičiavimą šis sulėtėjimas yra panaikinamas. Momento savybės trūkumas – funkcijos minimumo „pralėkimas“, dėl per didelės pagreičio vertės.

- Adaptyvaus pagreičio įvertinimo algoritmas *Adam* (angl. *adaptive moment estimation*) – ištaiso šį momento trūkumą. Šis algoritmas plokštumoje (mažo gradiento metu) stipriai padidina pagreičio vertes, tačiau padidėjus gradientui momentas yra sumažinamas. Adam algoritmas dinamiškai keičia mokymo greitį kiekvienam tinklo parametrai, adaptuodamas jį pagal gradientų pokyčio vidurkį. Šis algoritmas eksponentiškai mažina mokymosi greitį, taip padidindamas tikimybę rasti globalų funkcijos minimumą [42]. Šis optimizavimo algoritmas ir bus naudojamas šio neuroninio tinklo mokymo metu.
- Adaptyvaus pagreičio įvertinimo algoritmas su svorio mažinimu *AdamW* (angl. *adaptive moment estimation with weight decay*) – šis metodas yra Adam metodo patobulinimas siekiant sumažinti modelio perspaudimą. Perspaudimas tai modelio savybė „atsiminti“ mokymo duomenis, o ne juos generalizuoti. Prie Adam yra pridedama papildoma reguliarizavimo dalis, kuri palaiko svorinius koeficientus kuo mažesnius [43].

### 2.5.3. Tinklo struktūros parinkimas

Neuroninis tinklas naudotas metodų efektyvumo patikrinimui sudarytas iš dviejų paslėptųjų sluoksnių. Kiekvieną iš tų sluoksnių sudarė 10 neuronų. Sluoksniu aktyvavimui buvo naudota *ReLU* funkcija. Po kiekvieno paslėpto sluoksniu buvo pridėtas parametru numetimo sluoksniu (angl. *Dropout layer*) su 20% atmetimu. Šis sluoksniu naudojamas modelio perspaudimo sumažinimui. LSTM metodo naudojimo metu pirmasis sluoksniu yra pakeičiamas į LSTM sluoksniu su 18 neuronų. Mokymo metu buvo pasirinkta naudoti 1000 epochų, tačiau nustoti mokytis tuomet kai mokymo kriterijaus vertė nebekrenta 50 epochų. Taip epochų skaičius kinta dinamiškai priklausomai nuo modelio mokymosi greičio ir rezultatų.

Tinklo mokymo metu kokybės įvertinimui pasirinkta modifikuota vidutinė kvadratinė paklaida (angl. *Mean Square Error* MSE). MSE modifikuojamas dėl paklaidų mažesnių už 1 nykstamai mažėjančio vertinimo. Ši modifikacija padeda išlaikyti tiek mažesnių tiek didesnių už 1 paklaidų svarbą.

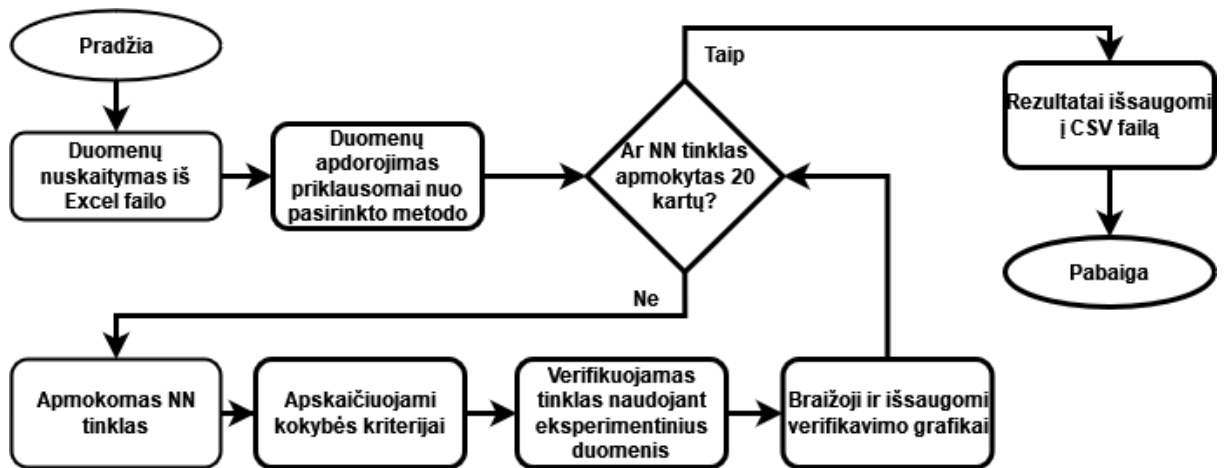
$$MMSE = \frac{\sum_{i=1}^n (|A_i^* - A_i| + 1)^2}{n}, \quad (6)$$

Po mokymo modeliui įvertinti yra naudojami visiškai nauji duomenys. Šiuo atveju modelio įvertinimui yra naudojama likutinė kvadratų suma (angl. *Residual sum of squares* RSS):

$$RSS = \sum_{i=1}^n (A_i^* - A_i)^2, \quad (7)$$

čia  $A_i^*$  - reali išmatuota vertė,  $A_i$  – neuroninio tinklo apskaičiuota vertė,  $n$  – duomenų kiekis.

Principinė algoritmo schema pateikta 2.5.5 paveiksle. Mokymo metu kiekvieno neuroninio tinklo atsikartojamumas testuojamas 20 kartų. Mokymo pabaigoje visi grafikai išsaugomi, o kokybės kriterijų vertės įrašomos į CSV formato failą. Kodo pavyzdys pateiktas 2.5.6 paveiksle.



2.5.5 pav. Mokymo algoritmo principinė schema

```

for gain in range(1, 20):
    train_x_mod = train_x
    valid_x_mod = valid_x

    n_features = len(read_columns) - 1

    if(mode == 'LSTM'):
        lstm_model = build_and_compile_model_LSTM(n_steps_for_LSTM, n_features)

    elif(mode == 'HyperLSTM'): ...
    elif(mode == 'Normal'):
        lstm_model = build_and_compile_model(train_x_mod)
    elif(mode == 'HyperNormal'):
        tuner = keras_tuner.BayesianOptimization(
            hypermodel=build_and_compile_model_Hyper_Fit,
            objective="val_loss",
            max_trials=500,
            executions_per_trial=2,
            overwrite=True,
            directory="HyperParameter_tuner",
            project_name="BayesianOptimization_500_PCA")

    if mode == 'HyperLSTM' or mode == 'HyperNormal':
        tuner.search_space_summary()
    else:
        lstm_model.summary()

    mae_ave = 0.0
    rss_ave = 0.0

    start = time.time()
    if mode == 'HyperLSTM' or mode == 'HyperNormal': ...
    else: ...
    print(f"Time: {(time.time() - start)} s")

    if mode != 'HyperLSTM' or mode != 'HyperNormal': ...

end_full = time.time()-start_full
print(f'{end_full}s')
if mode != 'HyperLSTM' or mode != 'HyperNormal':
    metrics.to_csv(f"{folder}/metrics_{end_full}s.csv", index = False)
  
```

2.5.6 pav. Tyrimo metu naudoto kodo dalis

### 3. Eksperimentinio tyrimo rezultatai

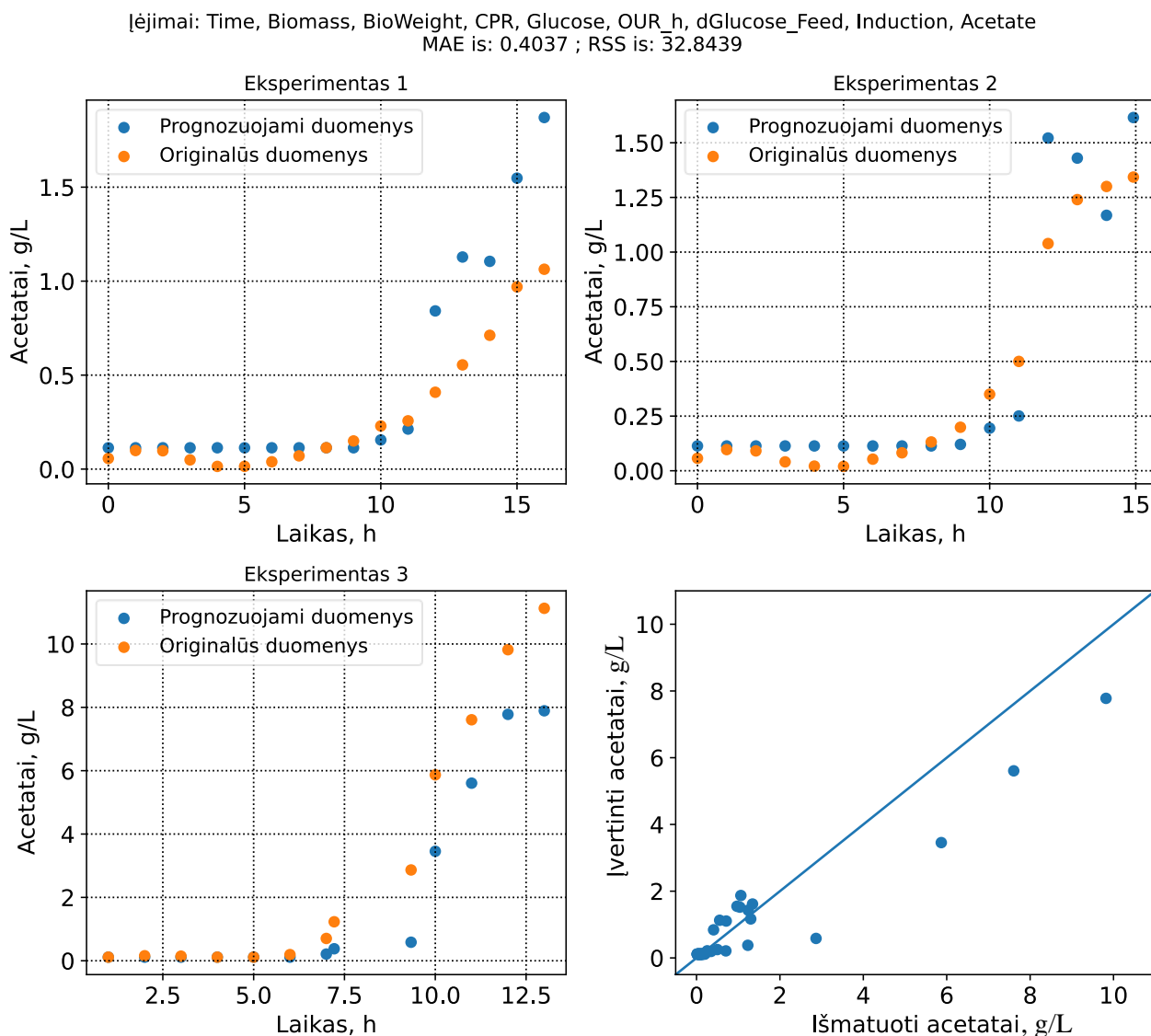
Tyrimo metu modeliams įvertinti buvo skaičiuojama likutinė kvadratų suma, tačiau papildomai naudota ir vidutinė absoliutinė paklaida (angl. *Mean absolute error* MAE).

$$MAE = \frac{\sum_{i=1}^n |A_i^* - A_i|}{n}, \quad (8)$$

čia  $A_i^*$  - reali išmatuota vertė,  $A_i$  - neuroninio tinklo apskaičiuota vertė,  $n$  – duomenų kiekis. Geriausiam modeliui išrinkti buvo naudojama MAE funkcija, o likutinė kvadratų suma naudota kaip papildomas kriterijus.

#### 3.1. Nemodifikuoti duomenys

Pirmasis mokymas buvo atliktas naudojant duomenis be papildomos modifikacijos (NMD). Šis metodas yra paprasčiausias ir nereikalaujantis papildomų skaičiavimų. Gauti rezultatai pasirinkti mokant neuroninį tinklą 20 kartų. Tai atliekama dėl modelio atsitiktinės prigimties.



3.1.1 pav. Patikrinimo rezultatai naudojant duomenis be PCA ar SVD modifikacijos

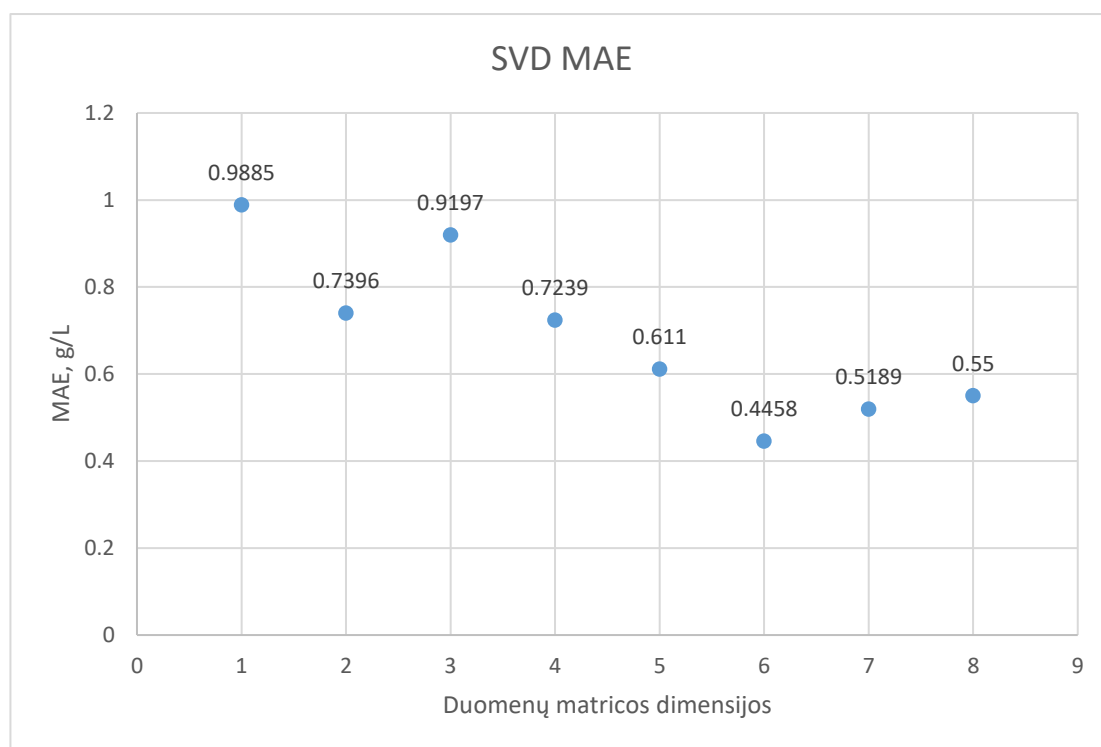
Rezultatai pavaizduoti 3.1.1 pav. Gautas MAE – 0.4037; RSS – 32.8439.

### 3.1.1. Atskirosios reikšmės skaidymas

Atskirosios reikšmės skaidymo tyrimo metu neuroninis tinklas buvo apmokomas keičiant duomenų matricos dimensių kiekį nuo 1 iki 8. Kokybės kriterijų vertės su skirtingomis matricos dimensiomis pateiktos 4 lentelėje, o pagrindinis kriterijus atvaizduotas 3.1.2 paveiksle. Lentelėje žalia spalva pavaizduotas geriausias variantas. Dimensių mažinimo metu mažiausią MAE paklaidą turėjo modelis naudotas su 7 įėjimo duomenų dimensiomis.

4 lentelė. Kokybės kriterijų priklausomybė nuo SVD dimensių kiekio

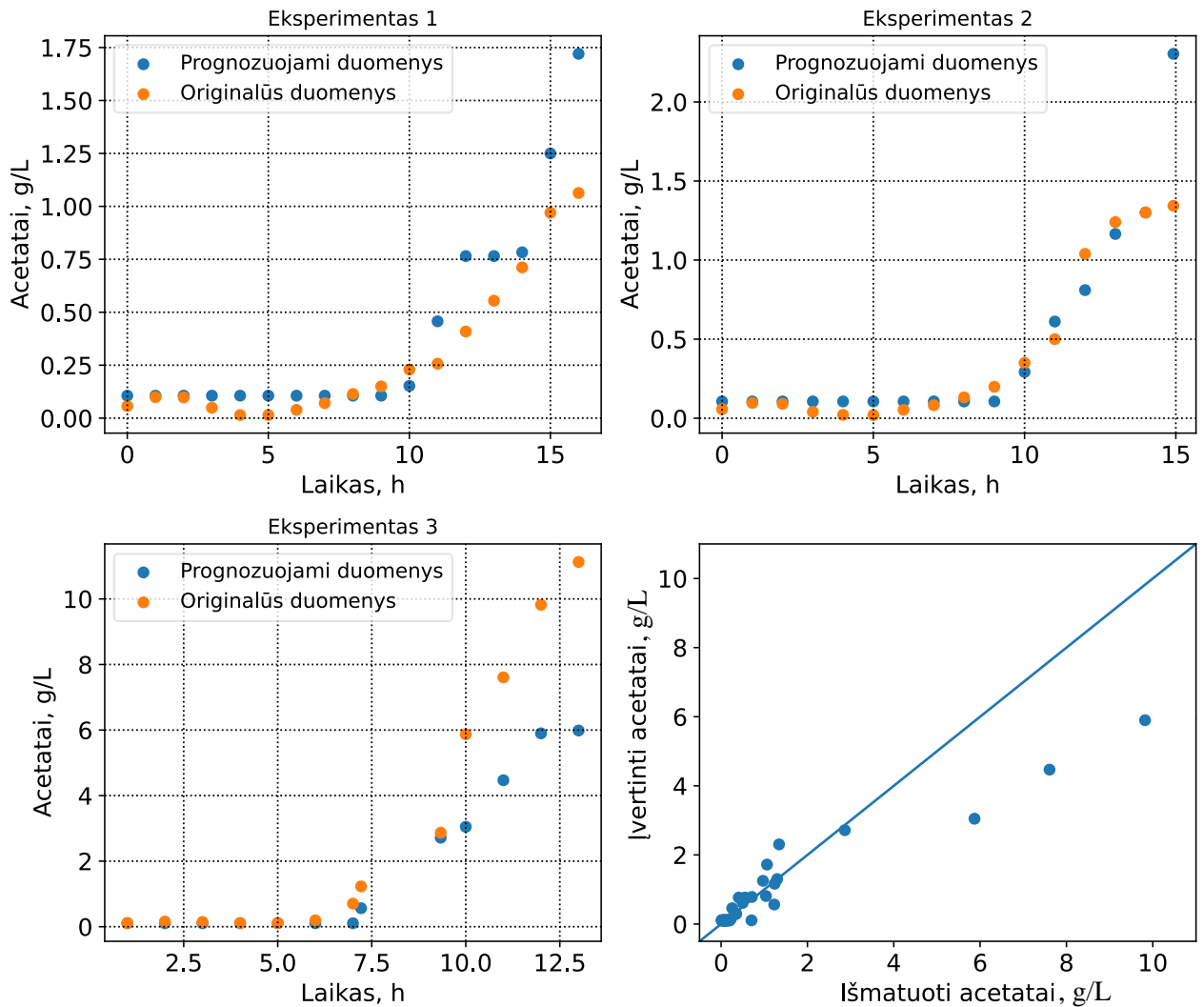
	1	2	3	4	5	6	7	8
RSS	158,5702	131,8901	207,5251	193,9359	96,3658	62,2571	83,7695	69,7023
MAE	0,9885	0,7396	0,9197	0,7239	0,611	0,4458	0,5189	0,55



3.1.2 pav. MAE priklausomybė nuo SVD dimensių kiekio

Iš grafiko matyti, kad modelio paklaidos sumažėjo beveik du kartus panaudojus 6 matricos dimensijas ir daugiau. Paklaidos pasiskirstymas naudojant nuo 5 iki 8 duomenų dimensių gali priklausyti ir nuo neuroninio tinklo atsitiktinumo faktoriaus. Geriausio modelio acetatų grafikas pavaizduotas 3.1.3 paveiksle.

Įėjimai: Time, Biomass, BioWeight, CPR, Glucose, OUR\_h, dGlucose\_Feed, Induction, Acetate  
 MAE is: 0.4458 ; RSS is: 62.2571



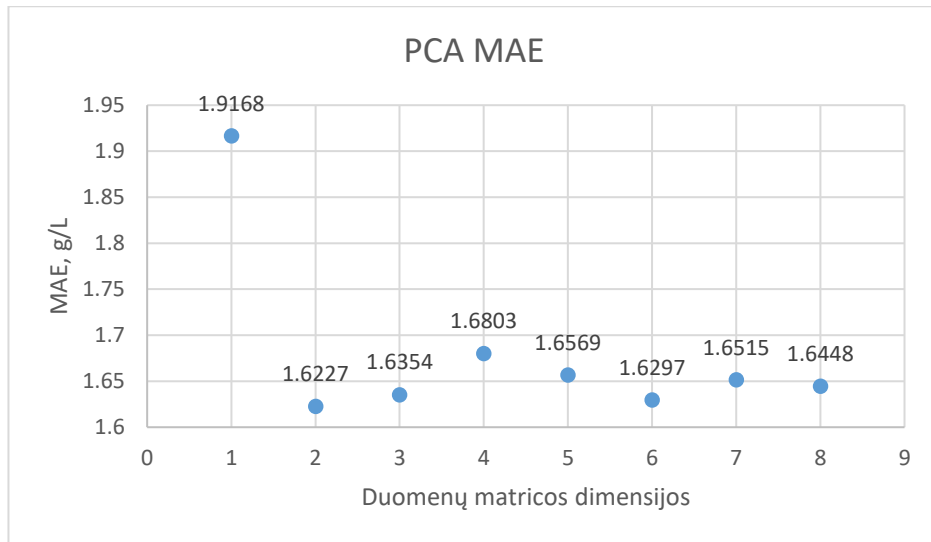
3.1.3 pav. Prognozuotų duomenų palyginimas su išmatuotais naudojant SVD metodą

### 3.1.2. Principinės komponentės analizė

Principinės komponentės analizės tyrimo metu neuroninis tinklas buvo apmokomas keičiant duomenų matricos dimensijų kiekį nuo 1 iki 8. Kokybės kriterijų vertės su skirtingomis duomenų dimensijomis pateiktos 5 lentelėje, o pagrindinis kriterijus atvaizduotas 3.1.4 paveiksle. Lentelėje žalia spalva pavaizduotas geriausias variantas. Duomenų matricos dimensijų mažinimo metu mažiausią MAE paklaidą turėjo modelis naudotas su 7 įėjimo matricos dimensijomis.

5 lentelė. Kokybės kriterijų priklausomybė nuo PCA dimensijų kiekio

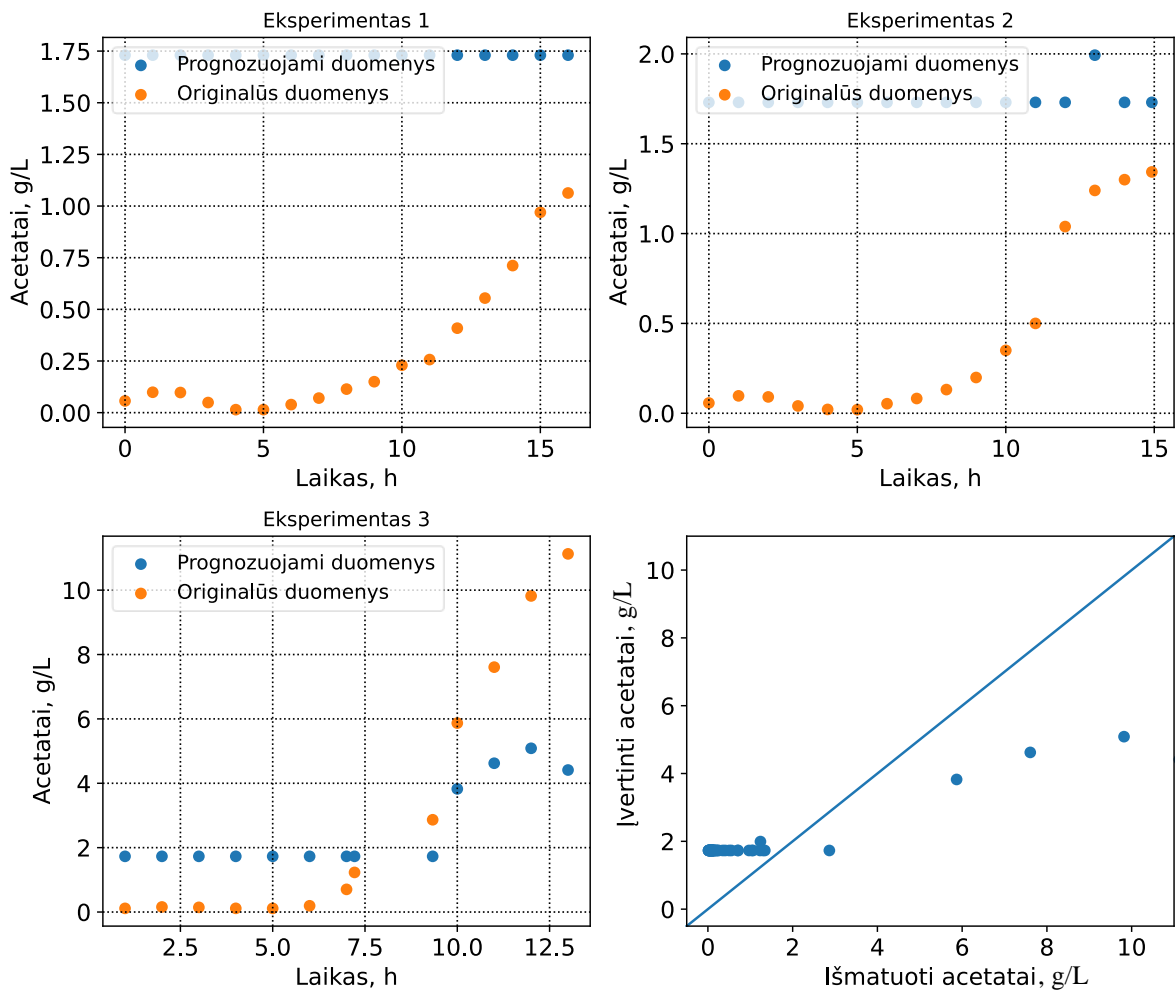
	1	2	3	4	5	6	7	8
RSS	246,8062	170,4654	178,6092	188,4638	187,0397	165,923	190,8996	166,0849
MAE	1,9168	1,6227	1,6354	1,6803	1,6569	1,6297	1,6515	1,6448



3.1.4 pav. MAE priklausomybė nuo PCA dimensijų kiekio

Iš grafiko ir lentelės matyti, kad naudojant PCA metodą geriausi rezultatai pasiekti naudojant 2 įėjimo matricos dimensijas. Šie rezultatai nėra patenkinami, paklaidos yra 2-3 kartus didesnės už paklaidas gautas naudojant kitus metodus. Geriausio modelio acetatų grafikas pavaizduotas 3.1.5 paveiksle.

Įėjimai: Time, Biomass, BioWeight, CPR, Glucose, OUR\_h, dGlucose\_Feed, Induction, Acetate  
 MAE is: 1.6227 ; RSS is: 170.4654

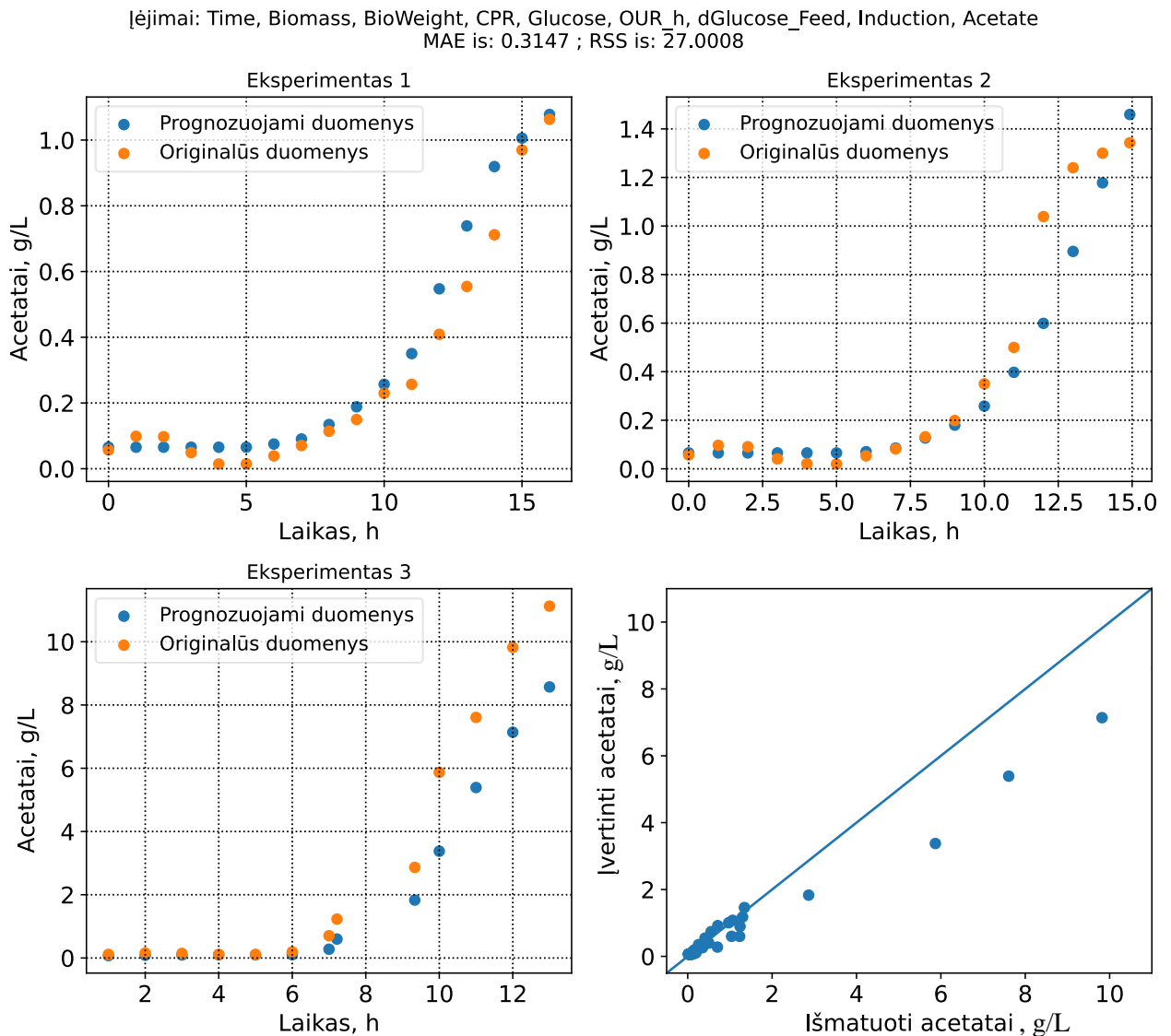


3.1.5 pav. Prognozuotų duomenų palyginimas su realiais naudojant PCA metodą

Šis metodas nėra atmetamas. Iš 3.1.5 paveikslo rezultatų galima daryti išvadą, kad modelis buvo perspaustas arba naudota netinkama neuroninio tinklo konfigūracija.

### 3.1.3. Ilgalaikis trumpalaikės atminties neuroninis tinklas

Paskutinis yra tiriamas LSTM metodas. Šiam metodui tirti yra ne tik pakeičiama neuroninio tinklo struktūra bet ir įėjimo duomenų paruošimas reikalauja daugiau resursų. Kaip ir pirminis neuroninis tinklas be SVD ir PCA metodų, šis yra apmokomas 20 kartų. Taip sumažinama atsitiktinumo galimybės įtaka. Rezultatai pavaizduoti 3.1.6 pav. Gautas MAE – 0,3147; RSS – 27.



3.1.6 pav. Prognozuotų duomenų palyginimas su realiais naudojant LSTM metodą

LSTM metodu gauti rezultatai yra geriausi iš 4 naudotų metodų. Visi šie metodai tiriami naudojant vienodą neuroninio tinklo struktūrą. Ši tyrimo metodika leido lengviau lyginti skirtingus variantus, tačiau šie rezultatai gali būti klaidingi. Šios metodikos tikslumu įsitikinti kiekvienam iš metodų buvo surasti geriausi neuroninio tinklo parametrai naudojantis Bajeso optimizavimu.



## 3.2. Bajeso optimizavimas

Bajeso optimizavimas yra statistinis metodas, skirtas efektyviai ieškoti geriausių dirbtinių neuroninių tinklų parametrų. Šis metodas yra paremtas Bajeso tikimybių teorija ir stochastiniu proceso modeliavimu. Bajeso optimizavimo tikslas yra rasti globalų stochastinės funkcijos minimumą arba maksimumą. Tai yra daug efektyvesnis būdas nei rankiniu būdu keisti parametrus ir patikrinti jų poveikį modelio tikslumui. Šis metodas padeda sumažinti laiko ir resursų išlaidas, kad būtų galima surasti optimalius ANN parametrus.

Šis optimizavimas veikia etapais, pasirenkant sekantį parametrų rinkinį, kurį reikia įvertinti, remiantis dabartiniu žinių apie modelio našumą. Kiekviename žingsnyje tikimybinis modelis pritaikomas prie pastebėtų parametrų ir atitinkamų objektyvų vertės. Šis modelis tada yra naudojamas numatyti naujo rinkinio, kuris dar nebuvo įvertintas, našumą. Modelis subalansuoja tyrinėjimo ir išnaudojimo kompromisą parametrų erdvėje, leisdamas metodui greitai susikoncentruoti į optimalius rinkinius, vengiant vertinimo nesėkmingose parametrų erdvės vietose. Šis metodas yra pritaikytas naudojant *Tensorflow-Keras* biblioteką.

Kiekvienam iš keturių prieš tai aptartų modelių buvo pritaikytas šis optimizavimas. Buvo keičiamas neuronų kiekis, numetimo procentas, aktyvavimo funkcija bei mokymo greitis. *Python* kodas kuriame aprašytas MFFN modelis su keičiamais parametrais pavaizduotas 3.2.1 paveiksle.

```
def build_and_compile_model_Hyper_Fit(hp):
    model = keras.Sequential()

    model.add(layers.Input(shape=(train_X.shape[1])))
    model.add(layers.Dense(
        units=hp.Int(f"1_units", min_value = 2, max_value = 20, step = 2),
        activation = hp.Choice(f"1_layer_activation", ["relu", "tanh", "sigmoid"])
    ))
    model.add(layers.Dropout(rate = hp.Float(f"1_dropout_rate", min_value = 0.05, max_value = 0.5, step = 0.05)))
    model.add(layers.Dense(
        units=hp.Int(f"2_units", min_value = 2, max_value = 20, step = 2),
        activation = hp.Choice(f"2_layer_activation", ["relu", "tanh", "sigmoid"])
    ))
    model.add(layers.Dropout(rate = hp.Float(f"2_dropout_rate", min_value = 0.05, max_value = 0.5, step = 0.05)))
    model.add(layers.Dense(1,
        kernel_constraint=non_neg(),
        bias_constraint=non_neg()))

    learning_rate = hp.Float("lr", min_value=1e-4, max_value=1e-2, sampling="log")
    model.compile(
        optimizer=keras.optimizers.Adam(learning_rate=learning_rate),
        loss=my_loss_fn,
    )
    return model
```

3.2.1 pav. MFFN modelio kodas su keičiamais parametrais

Programoje buvo bandoma 3 aktyvavimo funkcijos: ReLu, hiperbolinio tangento ir sigmoidine. Kitiems parametrams nurodytos pasirinkimo ribos nuo mažiausio iki didžiausio bei kitimo žingsnis. Mokymo greitis buvo keičiamas logaritmiškai. LSTM modelio parametrų optimizavimo kodas yra analogiškas, pridamas tik LSTM sluoksnio aktyvavimo pasirinkimas.

Kiekvienam modeliui buvo atlikta 500 iteracijų geriausiam parametrų rinkiniui rasti. Gauti parametrų rinkiniai pateikti 6 lentelėje.

**6 lentelė.** Bajeso optimizavimo metu rasti modelių parametrai

Metodas	Įėjimo aktyvavimas	Neuronų skaičius (1 sluoksnis)	Aktyvavimas (1 sluoksnis)	Numetimo kiekis (1 sluoksnis)	Neuronų skaičius (2 sluoksnis)	Aktyvavimas (2 sluoksnis)	Numetimo kiekis (2 sluoksnis)	Mokymo greitis
MFFN	-	20	ReLU	5%	20	Sigmoid	25%	0.01
SVD+MFFN	-	16	ReLU	5%	8	Sigmoid	5%	0.01
PCA+MFFN	-	4	Tanh	45%	14	Tanh	30%	0.0064
LSTM	ReLU	16	Tanh	45%	14	Tanh	35%	0.0043

7 lentelėje pateikti kokybės kriterijų rezultatai naudojant kiekvieną metodą.

**7 lentelė.** Metodų kokybės kriterijų vertės po Bajeso optimizavimo (BO)

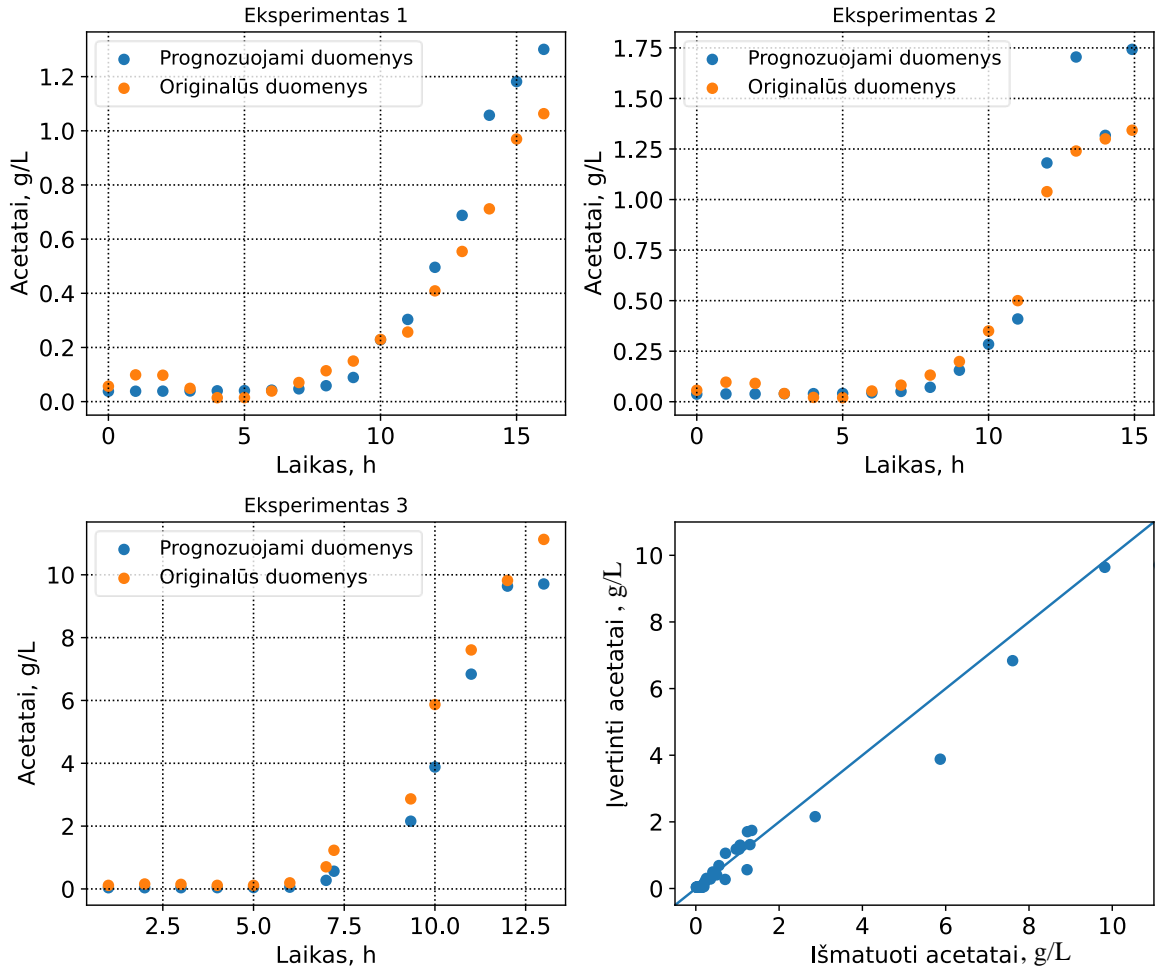
Metodas	MAE (be BO)	MAE	RSS (be BO)	RSS	Geriausios matricių dimensijos (be BO)	Geriausios matricių dimensijos
MFFN	0,4037	0,2054	32,8439	8,4755	-	-
SVD+MFFN	0,4458	0,2996	62,2571	20,0338	6	7
PCA+MFFN	1,6227	0,4562	170,4654	34,8972	2	8
LSTM	0,3147	0,2157	27,0008	7,6249	-	-

Po Bajeso optimizavimo vidutinės absoliutinės paklaidos sumažėjo 37,3 – 75,95%. MFFN metodo MAE yra mažesnis už LSTM. Skirtumas nėra didelis dėl to galima teigti, jog abu šie metodai yra adekvatūs acetatų prognozavimui. Toliau pateiktas MFFN įvertinimo grafikas bei MFFN mokymo funkcijos vertės priklausomybė nuo iteracijų kiekio (žr. 3.2.1, 3.2.2 pav.).

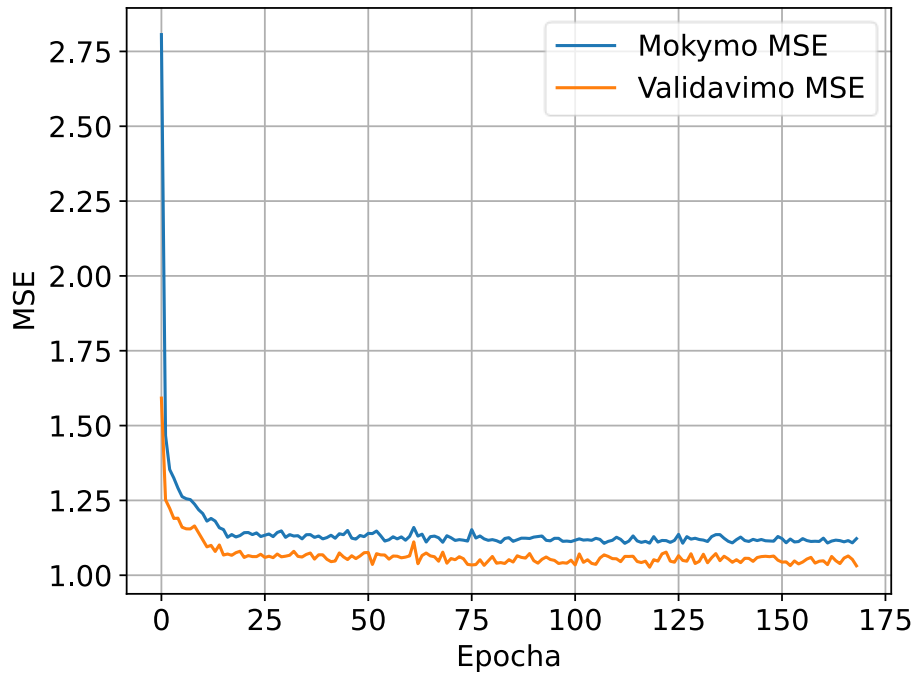
Lyginant 3.1.6 ir 3.2.2 paveikslus matyti, jog po Bajeso optimizacijos eksperimentas turinti didesnes vertes yra įvertintas geriau. Tai ypač matyti įvertintų acetatų priklausomybėje nuo išmatuotų.

Mokymo funkcijos grafike (žr. 3.2.3 pav.) matoma, jog mokymas nustotas po 170 epochų. Panašus epochų skaičius yra ir kitose mokymuose. Epochų skaičius kito nuo 120 iki 500 epochų. Maksimalus epochų skaičius pasiektas nebuvo. Lyginant mokymo ir validavimo kokybės funkcijos kreives matyti, jog mokymo metu šis modelis pasiekė prastesnius rezultatus. Įprastai validavimo, o ne mokymo metu yra pasiekiami prastesni rezultatai. Šiuo atveju šie rezultatai gauti dėl papildomų numetimo sluoksnių panaudojimo. Numetimo sluoksniai yra įjungiami tik mokymo metu, validavimo metu jie nėra naudojami.

Įėjimai: Time, Biomass, BioWeight, CPR, Glucose, OUR\_h, dGlucose\_Feed, Induction, Acetate  
 MAE is: 0.2054 ; RSS is: 8.4755



3.2.2 pav. Prognozuotų duomenų palyginimas su realiais naudojant MFFN metodą



3.2.3 pav. MFFN metodo mokymo funkcijos priklausomybė nuo epochų skaičiaus

## Rezultatai ir išvados

1. Realizuoti atskirosios reikšmės skaidymo (SVD) ir principinės komponentės analizės (PCA) metodai bei dvi neuroninio tinklo struktūros: daugiasluoksnio tiesinio sklidimo ir ilgalaikio trumpalaikės atminties modelio.
2. Naudojant duomenų apdorojimo metodus ir neuroninius tinklus įvertinta acetatų koncentracija. Gauta geriausia vidutinė absoliutinė paklaida – 0,3147 g/L t. y. apie 3% ir likutinė kvadratų suma – 27 (g/L)<sup>2</sup>. Šie rezultatai gauti naudojant ilgalaikį trumpalaikės atminties neuroninį tinklą.
3. Realizavus Bajeso optimizavimą, geriausiems modelių parametrams surasti gauta geriausia vidutinė absoliutinė paklaida – 0,2054 g/L t. y. apie 2% ir likutinė kvadratų suma – 8,4755 (g/L)<sup>2</sup>. Šie rezultatai gauti naudojant atnaujintą daugiasluoksnio tiesinio sklidimo neuroninį tinklą.

## Literatūros sąrašas

1. J. Randek and C.-F. Mandenius, "Online soft sensing in upstream bioprocessing," *Critical Reviews in Biotechnology*, vol. 38, no. 1, pp. 106–121, Jan. 2018, doi: [10.1080/07388551.2017.1312271](https://doi.org/10.1080/07388551.2017.1312271).
2. S. Rathore, S. Mishra, S. Nikita, and P. Priyanka, "Bioprocess Control: Current Progress and Future Perspectives," *Life*, vol. 11, no. 6, p. 557, Jun. 2021, doi: [10.3390/life11060557](https://doi.org/10.3390/life11060557)
3. M. Gavrilescu, 'Sustainability', in *Comprehensive Biotechnology*, Elsevier, 2011, pp. 905–923. doi: [10.1016/B978-0-08-088504-9.00147-1](https://doi.org/10.1016/B978-0-08-088504-9.00147-1).
4. J. S. Alford, 'Bioprocess control: Advances and challenges', *Computers & Chemical Engineering*, vol. 30, no. 10–12, pp. 1464–1475, Sep. 2006, doi: [10.1016/j.compchemeng.2006.05.039](https://doi.org/10.1016/j.compchemeng.2006.05.039).
5. Khan, Firdos Alam. (2020) 2020. *Biotechnology Fundamentals Third Edition*. 1st ed. CRC Press.
6. A. Cinar, S. J. Parulekar, C. Undey, and G. Birol, *Batch Fermentation*, 0 ed. CRC Press, 2003. doi: [10.1201/9780203911358](https://doi.org/10.1201/9780203911358).
7. B. J. Minihane and D. E. Brown, 'Fed-batch culture technology', *Biotechnology Advances*, vol. 4, no. 2, pp. 207–218, 1986, doi: [10.1016/0734-9750\(86\)90309-5](https://doi.org/10.1016/0734-9750(86)90309-5).
8. W. D. Maxon, 'Continuous Fermentation: A Discussion of Its Principles and Applications', *Appl Microbiol*, vol. 3, no. 2, pp. 110–122, Mar. 1955, doi: [10.1128/am.3.2.110-122.1955](https://doi.org/10.1128/am.3.2.110-122.1955).
9. H. G. Lawford and J. D. Rousseau, "Improving fermentation performance of recombinant zymomonas in acetic acid-containing media," *Appl Biochem Biotechnol*, vol. 70–72, no. 1, pp. 161–172, Mar. 1998, doi: [10.1007/BF02920133](https://doi.org/10.1007/BF02920133).
10. M. E. Pampulha and M. C. Loureiro-Dias, "Activity of glycolytic enzymes of *Saccharomyces cerevisiae* in the presence of acetic acid," *Appl Microbiol Biotechnol*, vol. 34, no. 3, Dec. 1990, doi: [10.1007/BF00170063](https://doi.org/10.1007/BF00170063).
11. C. M. Takahashi, D. F. Takahashi, M. L. C. Carvalhal, and F. Alterthum, "Effects of Acetate on the Growth and Fermentation Performance of *Escherichia coli* KO11," *ABAB*, vol. 81, no. 3, pp. 193–204, 1999, doi: [10.1385/ABAB:81:3:193](https://doi.org/10.1385/ABAB:81:3:193).
12. M. De Mey, S. De Maeseneire, W. Soetaert, and E. Vandamme, "Minimizing acetate formation in *E. coli* fermentations," *J Ind Microbiol Biotechnol*, vol. 34, no. 11, pp. 689–700, Nov. 2007, doi: [10.1007/s10295-007-0244-2](https://doi.org/10.1007/s10295-007-0244-2).
13. A. J. Wolfe, "The Acetate Switch," *Microbiol Mol Biol Rev*, vol. 69, no. 1, pp. 12–50, Mar. 2005, doi: [10.1128/MMBR.69.1.12-50.2005](https://doi.org/10.1128/MMBR.69.1.12-50.2005).
14. H. Zhang, "Software Sensors and Their Applications in Bioprocess," in *Computational Intelligence Techniques for Bioprocess Modelling, Supervision and Control*, vol. 218, M. do Carmo Nicoletti and L. C. Jain, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 25–56. doi: [10.1007/978-3-642-01888-6\\_2](https://doi.org/10.1007/978-3-642-01888-6_2).
15. V. M. Saucedo and M. N. Karim, 'Analysis and comparison of input-output models in a recombinant fed-batch fermentation', *Journal of Fermentation and Bioengineering*, vol. 83, no. 1, pp. 70–78, Jan. 1997, doi: [10.1016/S0922-338X\(97\)87330-8](https://doi.org/10.1016/S0922-338X(97)87330-8).
16. S. Foyo de Azevedo, B. Dahm, and F. R. Oliveira, 'Hybrid modelling of biochemical processes: A comparison with the conventional approach', *Computers & Chemical Engineering*, vol. 21, pp. S751–S756, May 1997, doi: [10.1016/S0098-1354\(97\)87593-X](https://doi.org/10.1016/S0098-1354(97)87593-X).

17. J. D. Owens and J. D. Legan, 'Determination of the Monod substrate saturation constant for microbial growth', *FEMS Microbiology Letters*, vol. 46, no. 4, pp. 419–432, Oct. 1987, doi: [10.1111/j.1574-6968.1987.tb02478.x](https://doi.org/10.1111/j.1574-6968.1987.tb02478.x).
18. M. Muloiwa, S. Nyende-Byakika, and M. Dinka, 'Comparison of unstructured kinetic bacterial growth models.', *South African Journal of Chemical Engineering*, vol. 33, pp. 141–150, Jul. 2020, doi: [10.1016/j.sajce.2020.07.006](https://doi.org/10.1016/j.sajce.2020.07.006).
19. D. Levisauskas, V. Galvanauskas, R. Simutis, and A. Lübbert, 'Model based calculation of substrate/inducer feed-rate profiles in fed-batch processes for recombinant protein production', *Biotechnology Techniques*, vol. 13, no. 1, pp. 37–42, 1999, doi: [10.1023/A:1008887514011](https://doi.org/10.1023/A:1008887514011).
20. L. Dewasme, G. Goffaux, A.-L. Hantson, and A. V. Wouwer, 'Experimental validation of an Extended Kalman Filter estimating acetate concentration in E. coli cultures', *Journal of Process Control*, vol. 23, no. 2, pp. 148–157, Feb. 2013, doi: [10.1016/j.jprocont.2012.09.004](https://doi.org/10.1016/j.jprocont.2012.09.004).
21. C. Di Massimo, M. J. Willis, G. A. Montague, M. T. Tham, and A. J. Morris, 'Bioprocess model building using artificial neural networks', *Bioprocess Engineering*, vol. 7, no. 1–2, pp. 77–82, Sep. 1991, doi: [10.1007/BF00383582](https://doi.org/10.1007/BF00383582).
22. G. Montague and J. Morris, 'Neural-network contributions in biotechnology', *Trends in Biotechnology*, vol. 12, no. 8, pp. 312–324, Aug. 1994, doi: [10.1016/0167-7799\(94\)90048-5](https://doi.org/10.1016/0167-7799(94)90048-5).
23. J. Thibault, V. Van Breusegem, and A. Chéruiy, 'On-line prediction of fermentation variables using neural networks: PREDICTION OF FERMENTATION VARIABLES', *Biotechnol. Bioeng.*, vol. 36, no. 10, pp. 1041–1048, Dec. 1990, doi: [10.1002/bit.260361009](https://doi.org/10.1002/bit.260361009).
24. H. Zhang, 'Software Sensors and Their Applications in Bioprocess', in *Computational Intelligence Techniques for Bioprocess Modelling, Supervision and Control*, vol. 218, M. do Carmo Nicoletti and L. C. Jain, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 25–56. doi: [10.1007/978-3-642-01888-6\\_2](https://doi.org/10.1007/978-3-642-01888-6_2).
25. M. N. Karim and S. L. Rivera, 'Comparison of feed-forward and recurrent neural networks for bioprocess state estimation', *Computers & Chemical Engineering*, vol. 16, pp. S369–S377, May 1992, doi: [10.1016/S0098-1354\(09\)80044-6](https://doi.org/10.1016/S0098-1354(09)80044-6).
26. L. A. Zadeh, 'Fuzzy sets', *Information and Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965, doi: [10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)
27. Escalante-Sánchez, J. Barrera-Cortés, H. M. Poggi-Varaldo, T. Ponce-Noyola, and I. S. Baruch, 'A soft sensor based on online biomass measurements for the glucose estimation and control of fed-batch cultures of Bacillus thuringiensis', *Bioprocess Biosyst Eng*, vol. 41, no. 10, pp. 1471–1484, Oct. 2018, doi: [10.1007/s00449-018-1975-3](https://doi.org/10.1007/s00449-018-1975-3).
28. P. R. Patnaik, 'Artificial intelligence as a tool for automatic state estimation and control of bioreactors', *Laboratory Robotics and Automation*, vol. 9, no. 6, pp. 297–304, 1997, doi: [10.1002/\(SICI\)1098-2728\(1997\)9:6<297::AID-LRA3>3.0.CO;2-W](https://doi.org/10.1002/(SICI)1098-2728(1997)9:6<297::AID-LRA3>3.0.CO;2-W).
29. J. Schubert, R. Simutis, M. Dors, I. Havlik, and A. Lübbert, 'Bioprocess optimization and control: Application of hybrid modelling', *Journal of Biotechnology*, vol. 35, no. 1, pp. 51–68, Jun. 1994, doi: [10.1016/0168-1656\(94\)90189-9](https://doi.org/10.1016/0168-1656(94)90189-9).
30. A. Survyla, D. Levisauskas, R. Urniezius, and R. Simutis, 'An oxygen-uptake-rate-based estimator of the specific growth rate in Escherichia coli BL21 strains cultivation processes',

- Computational and Structural Biotechnology Journal*, vol. 19, pp. 5856–5863, 2021, doi: [10.1016/j.csbj.2021.10.015](https://doi.org/10.1016/j.csbj.2021.10.015).
31. A. Survyla, R. Urniezius, and R. Simutis, ‘Viable cell estimation of mammalian cells using off-gas-based oxygen uptake rate and aging-specific functional’, *Talanta*, vol. 254, p. 124121, Mar. 2023, doi: [10.1016/j.talanta.2022.124121](https://doi.org/10.1016/j.talanta.2022.124121).
  32. R. Urniezius, A. Survyla, D. Paulauskas, V. A. Bumelis, and V. Galvanauskas, ‘Generic estimator of biomass concentration for *Escherichia coli* and *Saccharomyces cerevisiae* fed-batch cultures based on cumulative oxygen consumption rate’, *Microb Cell Fact*, vol. 18, no. 1, p. 190, Dec. 2019, doi: [10.1186/s12934-019-1241-7](https://doi.org/10.1186/s12934-019-1241-7).
  33. R. Urniezius, B. Kemesi, and R. Simutis, ‘Bridging Offline Functional Model Carrying Aging-Specific Growth Rate Information and Recombinant Protein Expression: Entropic Extension of Akaike Information Criterion’, *Entropy*, vol. 23, no. 8, p. 1057, Aug. 2021, doi: [10.3390/e23081057](https://doi.org/10.3390/e23081057).
  34. G. W. Stewart, “On the Early History of the Singular Value Decomposition,” *SIAM Rev.*, vol. 35, no. 4, pp. 551–566, Dec. 1993, doi: [10.1137/1035134](https://doi.org/10.1137/1035134).
  35. L. Zhang, J. S. Marron, H. Shen, and Z. Zhu, “Singular Value Decomposition and Its Visualization,” *Journal of Computational and Graphical Statistics*, vol. 16, no. 4, pp. 833–854, Dec. 2007, doi: [10.1198/106186007X256080](https://doi.org/10.1198/106186007X256080).
  36. S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1–3, pp. 37–52, Aug. 1987, doi: [10.1016/0169-7439\(87\)80084-9](https://doi.org/10.1016/0169-7439(87)80084-9).
  37. G. D’Agostini, “On the use of the covariance matrix to fit correlated data,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 346, no. 1–2, pp. 306–311, Jul. 1994, doi: [10.1016/0168-9002\(94\)90719-6](https://doi.org/10.1016/0168-9002(94)90719-6).
  38. „Understanding LSTM Networks“ prieiga per internetą: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> ; Žiūrėta: 2023-05-28
  39. M. Shanker, M. Y. Hu, and M. S. Hung, “Effect of data standardization on neural network training,” *Omega*, vol. 24, no. 4, pp. 385–397, Aug. 1996, doi: [10.1016/0305-0483\(96\)00010-2](https://doi.org/10.1016/0305-0483(96)00010-2).
  40. C. Bircanoglu and N. Arica, “A comparison of activation functions in artificial neural networks,” in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, Izmir, May 2018, pp. 1–4. doi: [10.1109/SIU.2018.8404724](https://doi.org/10.1109/SIU.2018.8404724).
  41. Bottou, Léon. "Stochastic gradient learning in neural networks." *Proceedings of Neuro-Nimes* 91.8 (1991): 12.
  42. D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization.” arXiv, Jan. 29, 2017. Accessed: Jan. 16, 2023. [Online]. Available: <http://arxiv.org/abs/1412.6980>
  43. I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization.” arXiv, Jan. 04, 2019. Accessed: Jan. 16, 2023. [Online]. Available: <http://arxiv.org/abs/1711.05101>