



**Kauno technologijos universitetas**

Elektros ir elektronikos fakultetas

**Biotechnologinio proceso būsenos kintamųjų įvertinimas,  
naudojant autoasociatyvinius neuroninius tinklus**

Baigiamasis magistro projektas

---

**Matas Ragauskas**

Projekto autorius

**Prof. dr. Vytautas Galvanauskas**

Vadovas

---

**Kaunas, 2026**



**Kauno technologijos universitetas**

Elektros ir elektronikos fakultetas

**Biotechnologinio proceso būsenos kintamųjų įvertinimas,  
naudojant autoasociatyvinius neuroninius tinklus**

Baigiamasis magistro projektas

Valdymo technologijos (6211EX014)

---

**Matas Ragauskas**

Projekto autorius

**Prof. dr. Vytautas Galvanauskas**

Vadovas

**Prof. dr. Renaldas Urniežius**

Recenzentas

---

**Kaunas, 2026**



**Kauno technologijos universitetas**

Elektros ir elektronikos fakultetas

Matas Ragauskas

## **Biotechnologinio proceso būsenos kintamųjų įvertinimas, naudojant autoasociatyvinius neuroninius tinklus**

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.

Matas Ragauskas

*Patvirtinta elektroniniu būdu*

Ragauskas Matas. Biotechnologinio proceso būsenos kintamųjų įvertinimas, naudojant autoasociatyvinius neuroninius tinklus. Magistro baigiamasis projektas / vadovas prof. dr. Vytautas Galvanauskas; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas.

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

Reikšminiai žodžiai: autoasociatyviniai neuroniniai tinklai, netiesioginis būsenos įvertinimas, biotechnologinis procesas.

Kaunas, 2026. 42 p.

### **Santrauka**

Biotechnologiniai procesai yra sudėtingos netiesinės sistemos, kurių efektyvus valdymas priklauso nuo tikslaus proceso būsenos kintamųjų įvertinimo. Praktikoje dalis svarbių kintamųjų, tokių kaip biomasės koncentracija, negali būti tiesiogiai matuojami realiuoju laiku, nes jų nustatymas reikalauja laboratorinių tyrimų. Dėl šios priežasties aktualu taikyti programinius jutiklius, paremtus duomenimis grįstais metodais, kurie leidžia netiesiogiai įvertinti sunkiai išmatuojamus proceso parametrus pagal kitus realiuoju laiku registruojamus kintamuosius.

Šiame darbe nagrinėjamas biotechnologinio GFP baltymo sintezės proceso būsenos kintamųjų įvertinimas taikant neuroninius tinklus. Tyrimui naudoti eksperimentiniai kultivavimo proceso duomenys, kurie prieš neuroninių tinklų apmokymą buvo apdoroti: sujungti pagal laboratorinių matavimų laikus, papildyti išvestiniais kintamaisiais, normalizuoti, taip pat atmesti klaidingi ir netipiniai eksperimentai. Iš viso neuroninių tinklų apmokymui naudoti 78 eksperimentų duomenys.

Darbe sudaryti ir ištirti skirtingų struktūrų autoasociatyviniai neuroniniai tinklai, skirti biomasės kintamajam atkurti. Tinklų apmokymui taikytas Levenbergo–Marquardto algoritmas, o modelių tikslumas vertintas pagal vidutinę absoliutinių nuokrypių sumą, vidutinį absoliutinį nuokrypį ir absoliutinę procentinę paklaidą. Geriausi autoasociatyvinio neuroninio tinklo rezultatai gauti naudojant 4:2:2:2:4 struktūrą: šio modelio vidutinė absoliutinių nuokrypių suma siekė 4485,5 g, vidutinis absoliutinis nuokrypis buvo 10,86 g, o absoliutinė procentinė paklaida – 9,05 %.

Siekiant įvertinti gautų rezultatų efektyvumą, autoasociatyvinis neuroninis tinklas buvo palygintas su tiesioginės sklaidos neuroniniu tinklu. Geriausi tiesioginės sklaidos neuroninio tinklo rezultatai gauti naudojant 3:2:1 struktūrą: vidutinė absoliutinių nuokrypių suma siekė 3097,0 g, vidutinis absoliutinis nuokrypis buvo 7,49 g, o absoliutinė procentinė paklaida – 6,26 %. Palyginus abu metodus nustatyta, kad tiesioginės sklaidos neuroninis tinklas buvo tikslesnis konkrečiai biomasės prognozavimo užduočiai, tačiau autoasociatyvinis neuroninis tinklas taip pat gali būti taikomas biotechnologinio proceso būsenos kintamųjų atkūrimui.

Ragauskas Matas. Estimation of biotechnological process state variables using autoassociative neural networks. Master's Final Degree Project / supervisor prof. dr. Vytautas Galvanauskas; Faculty of Electrical and Electronics Engineering, Kaunas University of Technology.

Study field and area (study field group): electronics engineering (engineering science).

Keywords: autoasociative neural networks, state estimation, biotechnological process.

Kaunas, 2026. 42 pages.

### **Summary**

Biotechnological processes are complex nonlinear systems whose effective control depends on the accurate estimation of process state variables. In practice, some important variables, such as biomass concentration, cannot be measured directly in real time because their determination requires laboratory analysis. For this reason, it is relevant to apply soft sensors based on data-driven methods, which allow difficult-to-measure process parameters to be indirectly estimated using other variables recorded in real time.

This work investigates the estimation of state variables in a biotechnological GFP protein synthesis process using neural networks. The study used experimental cultivation process data, which were processed before neural network training: the data were combined according to laboratory measurement times, supplemented with derived variables, normalized, and incorrect or atypical experiments were removed. In total, data from 78 experiments were used for neural network training.

In this work, autoassociative neural networks of different structures were developed and analyzed for the reconstruction of the biomass variable. The Levenberg–Marquardt algorithm was applied for network training, while model accuracy was evaluated using the average sum of absolute deviations, the average absolute deviation, and the absolute percentage error. The best results for the autoassociative neural network were obtained using a 4:2:2:2:4 structure: the average sum of absolute deviations was 4485.5 g, the average absolute deviation was 10.86 g, and the absolute percentage error was 9.05 %.

In order to evaluate the effectiveness of the obtained results, the autoassociative neural network was compared with a feedforward neural network. The best results for the feedforward neural network were obtained using a 3:2:1 structure: the average sum of absolute deviations was 3097.0 g, the average absolute deviation was 7.49 g, and the absolute percentage error was 6.26 %. The comparison of both methods showed that the feedforward neural network was more accurate for the specific biomass prediction task, while the autoassociative neural network can also be applied for the reconstruction of biotechnological process state variables.

## Turinys

<b>Lentelių sąrašas .....</b>	<b>7</b>
<b>Paveikslų sąrašas .....</b>	<b>8</b>
<b>Įvadas.....</b>	<b>9</b>
<b>1. Literatūros analizė.....</b>	<b>10</b>
1.1. Biotechnologiniai procesai .....	10
1.2. Būsenos kintamųjų nustatymo būdai.....	10
1.2.1. Masių balansai .....	10
1.2.2. Filtrai .....	11
1.2.3. Adaptyvūs stebėtojai .....	12
1.2.4. Neuroniniai tinklai.....	12
1.2.5. Hibridiniai modeliai.....	13
1.3. Dirbtiniai neuroniniai tinklai .....	13
1.3.1. Neuroninių tinklų komponentai.....	13
1.3.2. Apmokymo algoritmai.....	15
1.4. Autoasociatyviniai neuroniniai tinklai .....	15
1.4.1. Autoasociatyvinių neuroninių tinklų architektūra .....	15
1.4.2. Konvoliuciniai neuroniniai tinklai.....	16
1.4.3. Autoasociatyvinių neuroninių tinklų apmokymas.....	17
1.4.4. Autoasociatyvinių neuroninių tinklų variacijos.....	18
1.4.5. Autoasociatyvinių neuroninių tinklų taikymas.....	18
<b>2. Metodinė dalis.....</b>	<b>19</b>
2.1. Duomenų paruošimas .....	19
2.1.1. Išvestinių dydžių skaičiavimas .....	20
2.1.2. Netipinių duomenų atmetimas.....	20
2.1.3. Duomenų sujungimas .....	21
2.1.4. Apmokymo duomenų normalizavimas.....	21
2.2. Neuroninių tinklų apmokymas .....	22
2.2.1. Neuroninio tinklo struktūra .....	22
2.2.2. Neuroninio tinklo struktūros parinkimo principai.....	23
2.2.3. Autoasociatyvinio neuroninio tinklo apmokymas taikant Levenbergo – Marquardo algoritimą.....	24
2.2.4. Vertinimo kriterijai .....	27
2.2.5. Apmokymo duomenų kintamųjų parinkimas .....	28
2.2.6. Automatinis neuroninio tinklo apmokymas .....	28
<b>3. Tyrimo rezultatai.....</b>	<b>29</b>
3.1. Autoasociatyvinis neuroninis tinklas.....	29
3.1.1. Duomenų kintamųjų parinkimas .....	29
3.1.2. Automatinis neuroninio tinklo apmokymas .....	31
3.2. Tiesioginės sklaidos neuroninis tinklas .....	34
3.2.1. Automatinis neuroninio tinklo apmokymas .....	34
3.3. Gautų rezultatų apibendrinimas.....	36
<b>Išvados .....</b>	<b>38</b>
<b>Literatūros sąrašas .....</b>	<b>39</b>

## Lentelių sąrašas

<b>1 lentelė.</b> Duomenų kintamųjų pavadinimai ir reikšmės.....	19
<b>2 lentelė.</b> Autoasociatyvinio neuroninio tinklo validacijos paklaidų priklausomybė nuo neuroninio tinklo struktūros.....	32
<b>3 lentelė.</b> Tiesioginės sklaidos neuroninio tinklo validacijos paklaidų priklausomybė nuo neuroninio tinklo struktūros.....	35

## Paveikslų sąrašas

<b>1 pav.</b> Principinė neuroninio tinklo schema.....	14
<b>2 pav.</b> Autoasociatyvinio neuroninio tinklo architektūra.....	16
<b>3 pav.</b> Duomenų palyginimas.....	21
<b>4 pav.</b> Duomenų palyginimas su normalizuotais duomenimis.....	22
<b>5 pav.</b> Biomasės įverčiai, gauti naudojant autoasociatyvinį neuroninį tinklą (apmokymui naudoti kintamieji: biomasė, OUR integralas, bazė, CPR integralas).....	30
<b>6 pav.</b> Biomasės įverčiai, gauti naudojant autoasociatyvinį neuroninį tinklą (apmokymui naudoti kintamieji: biomasė, OUR integralas, bazė, CPR integralas).....	30
<b>7 pav.</b> Biomasės įverčiai, gauti naudojant autoasociatyvinį neuroninį tinklą (apmokymui naudoti kintamieji: biomasė, OUR integralas, bazė, CPR integralas).....	31
<b>8 pav.</b> Biomasės įverčiai, gauti naudojant tiesioginės sklaidos neuroninį tinklą.....	36

## Įvadas

Biotechnologiniai procesai yra sudėtingos, netiesinės ir dažnai sunkiai modeliuojamos sistemos, kurių efektyvus valdymas tiesiogiai priklauso nuo gebėjimo tiksliai įvertinti proceso būsenos kintamuosius. Praktikoje dalis šių kintamųjų negali būti išmatuojami realiu laiku arba jų matavimas yra sudėtingas, brangus ir reikalaujantis laboratorinių tyrimų. Dėl šios priežasties vis plačiau taikomi netiesioginiai būsenos kintamųjų nustatymo metodai, paremti matematiniais modeliais ir duomenimis grįstais sprendimais.

Tradiciniai metodai, tokie kaip masių balansai ar filtravimo algoritmai, nors ir yra plačiai naudojami, turi ribotą pritaikomumą sudėtingose netiesinėse sistemose. Pastaraisiais metais didelį dėmesį pritraukė dirbtiniai neuroniniai tinklai, gebantys modeliuoti sudėtingas priklausomybes tarp proceso kintamųjų be iš anksto žinomos fizikinės struktūros. Ypatingai išsiskiria autoasociatyviniai neuroniniai tinklai, kurie gali būti naudojami trūkstamų ar neišmatuojamų duomenų atkūrimui, remiantis kitais proceso kintamaisiais.

Šiame darbe nagrinėjamas biotechnologinis GFP baltymo sintezės procesas, kurio metu siekiama nustatyti proceso būsenos kintamuosius naudojant autoasociatyvinius neuroninius tinklus. Taip pat atliekamas šių tinklų palyginimas su tiesioginės sklaidos neuroniniais tinklais, siekiant įvertinti skirtingų metodų efektyvumą sprendžiant konkrečią būsenos kintamojo nustatymo užduotį.

**Darbo tikslas:** ištirti autoasociatyvinių neuroninių tinklų taikymo galimybes biotechnologinio proceso būsenos kintamųjų nustatymui ir įvertinti jų veikimo efektyvumą.

### **Darbo uždaviniai:**

1. Išanalizuoti biotechnologinių procesų būsenos kintamųjų nustatymo metodus.
2. Paruošti eksperimentinius duomenis neuroninių tinklų apmokymui.
3. Sudaryti ir apmokyti autoasociatyvinį neuroninį tinklą.
4. Nustatyti optimalią neuroninio tinklo struktūrą.
5. Įvertinti modelio tikslumą ir stabilumą.
6. Palyginti autoasociatyvinį neuroninį tinklą su tiesioginės sklaidos neuroniniu tinklu.

## 1. Literatūros analizė

### 1.1. Biotechnologiniai procesai

Biotechnologiniai procesai apima platų technologijų ir metodų spektrą, kuriuose naudojami gyvieji organizmai, jų ląstelės ar biologinės sistemos produktų kūrimui ir praktinių sprendimų taikymui įvairiose srityse, tokiose kaip farmacija, aplinkosauga ir žemės ūkis. Nors biotechnologijų ištakos siekia senovę, kai fermentacijos procesai buvo taikomi maisto gamyboje ir augalininkystėje, šiuolaikinė biotechnologija reikšmingai pažengė genų inžinerijos, molekulinės biologijos ir kitose srityse. Pastaraisiais dešimtmečiais ši sritis sulaukė ypatingo mokslininkų bei visuomenės dėmesio dėl savo potencialo spręsti aktualias globalias problemas. Vienas ryškiausių pavyzdžių – biotechnologinių procesų svarba kuriant vakcinas nuo SARS-CoV-2 viruso [1].

Biotechnologinių procesų būsenos kintamieji yra pagrindiniai parametrai naudojami biotechnologinių sistemų modeliavime ir valdyme, nurodantys esmines charakteristikas apibrėžiančias sistemos būseną duotuoju laiko momentu. Šie kintamieji yra be galo svarbūs norint suprasti biologinių, cheminių ir fizinių sistemų dinamikas, dėl ko jie yra nepakeičiami įvairiose biotechnologijų pramonės srityse [2]. Tiksliai stebint ir koreguojant būsenos kintamuosius galima optimizuoti gamybos efektyvumą bei užtikrinti proceso valdymo pastovumą. Būsenos kintamųjų matavimą įprastai galima suskirstyti į dvi grupes: empirinių duomenų matavimą realiu laiku naudojant įvairius fizinius jutiklius ir programinius jutiklius, kurie netiesiogiai įvertina kintamuosius [3].

### 1.2. Būsenos kintamųjų nustatymo būdai

Biologinių procesų valdymui būtinas esminių parametru vertinimas realiu metu. Tačiau svarbios būsenos kintamųjų (pvz.: produkto koncentracija ar biomasės augimo greitis) vertės negali būti išmatuojamos per reikiamą laiką. Todėl šios vertės dažnai yra įvertinamos netiesiogiai naudojant programinius jutiklius.

#### 1.2.1. Masių balansai

Masių balansai yra vieni iš fundamentalių principų naudojamų biotechnologijoje. Šis principas pagrįstas masės tvermės dėsniu, kuris teigia, jog materija iš niekur neatsiranda ir niekur nedingsta, bet gali būti transformuojama [4]. Biologiniuose procesuose, kaip fermentacija, ląstelių auginimas ar fermentų reakcijos, masių balansais suskaičiuojami substratai, produktai, šalutiniai produktai bei biomasė, kad būtų galima optimizuoti išėigą [2].

Bendruoju atveju masių balansas aprašomas 1 lygtimi, pagal kurią bioreaktoriuje sukauptas medžiagos kiekis nustatomas įvertinant į sistemą patenkančią ir joje susidarantį medžiagos kiekį bei atimant suvartotą ir iš sistemos pasišalinantį kiekį [4, 5]. Atsižvelgiant į konkrečią taikymo sritį ar proceso pobūdį, ši lygtis gali būti modifikuojama, tačiau jos esmė išlieka nepakitusi – būtina sistemiškai įvertinti visus medžiagos srautus ir jų pokyčius.

$$Kaupimas = Įėjimas + Gamyba - Suvartojimas - Išėjimas \quad (1)$$

Būsenos kintamųjų vertinime masių balansai dažnai atlieką svarbią rolę bendros sistemos būsenos nustatymui. Įprastai masių balansas tiesiogiai neduoda informacijos apie ieškomus kintamuosius.

Tačiau, panaudojant kitus būsenos kintamųjų nustatymo būdus, yra svarbus kitų metodų veikimui bei jų rezultatų aktualumui [6].

### 1.2.2. Filtrai

Būsenos kintamųjų nustatymui taikomi įvairūs filtravimo metodai, leidžiantys, remiantis žinomais stochastiniais modeliais ir eksperimentinių matavimų duomenimis, prognozuoti šių kintamųjų vertes. Vienas plačiausiai taikomų metodų yra Kalmano filtras bei jo įvairios modifikacijos [7]. Paprasčiausia Kalmano filtro forma veikia atlikdama du rekursiškai pasikartojančius etapus: pirmiausia prognozuojamos būsenos kintamųjų vertės, atsižvelgiant į sistemos dinamiką, o vėliau šios prognozės koreguojamos pagal išmatuotų kintamųjų teikiamą informaciją. Sistemos dinamika aprašoma 2 lygtimi, o matavimo išėjimai – 3 lygtimi. Šiame modelyje sistemos lygtį veikia baltasis triukšmas, apibūdinamas skirstiniu  $p(\omega) \sim N(0, Q)$ , o matavimo lygtį – baltasis triukšmas, apibūdinamas skirstiniu  $p(v) \sim N(0, R)$  [8].

$$x_k = Ax_{k-1} + Bu_k + \omega_{k-1} \quad (2)$$

$$z_k = Hx_k + v_k \quad (3)$$

Kalmano filtras remiasi tuo, kad  $x_k$  būseną laiko momentu  $k$  yra gaunama iš vidurkio,  $\bar{x}_k$  ir  $\bar{P}_k$  kovariacijos. Šis filtras gauna informaciją laiko momentu  $k - 1$  kintamaisiais  $\hat{x}_{k-1}$  ir  $P_{k-1}$ . Taip pat gaunamas valdymo signalas  $u_k$  ir išmatuota jutiklių informacija  $z_k$ . Tuomet Kalmano filtro išėjimas laiko momentu  $k$  yra aprašytas  $\hat{x}_k$  ir  $P_k$  vertėmis [8].

Sistemos būseną nuspėjama naudojant 4 ir 5 lygtis. Čia  $A_k$  – nusako objekto būseną,  $B$  – nurodo valdymo signalų vertes,  $Q_k$  – nusako sistemos triukšmo kovariaciją. Šiuo momentu naudojami praeitos būsenos duomenys sistemos būsenai įvertinti [8].

$$\bar{x}_k = A_k \hat{x}_{k-1} + Bu_k \quad (4)$$

$$\bar{P}_k = A_k P_{k-1} A_k^H + Q_k \quad (5)$$

Pagal išmatuotus duomenis sistemos būseną yra patikslinama naudojant 6-8 lygtis. Čia  $H_k$  – nusakoma išmatuota aplinkos būseną,  $R_k$  – nurodo matavimų triukšmo kovariaciją,  $K_k$  – nurodo Kalmano stiprinimo koeficientą. Nuspėtos būsenos reikšmės yra palyginamos su išmatuotomis vertėmis, taip sistema pakoreguoja savo parametrus, jog kito ciklo metu spėjimai būtų atliekami tiksliau [8].

$$K_k = \bar{P}_k H_k^H (H_k P_k H_k^H + R_k)^{-1} \quad (6)$$

$$\hat{x}_k = \bar{x}_k + K_k (z_k - H_k \bar{x}_k) \quad (7)$$

$$P_k = (I - K_k H_k) \bar{P}_k \quad (8)$$

Biotechnologiniuose procesuose dažnai taikomas išplėstasis Kalmano filtras (angl. extended Kalman filter), kuris yra tinkamas nesudėtingų netiesinių sistemų būsenų vertinimui. Tuo tarpu sudėtingesnėms sistemoms, kurių netiesiškumas negali būti pakankamai tiksliai aproksimuotas pirmosios eilės Teiloro skirstiniu, taikomas neskaidytas Kalmano filtras (angl. unscented Kalman filter) [9]. Pastaraisiais metais, didėjant skaičiavimo išteklių prieinamumui, vis didesnio dėmesio

sulaukia dalelių filtras (angl. particle filter). Šis metodas yra ypač naudingas valdant sudėtingas netiesines sistemas ir apdorojant ne Gauso pobūdžio triukšmą [7].

Dėl jų stochastinės prigimties ir galimybės gauti informaciją apie būsenos kintamuosius iš įvairių tipų sistemų, filtrai tapo neatsiejama būsenos kintamųjų vertinimo dalimi. Dėl plataus filtrų tipų pasirinkimo įprastai galima surasti ir pritaikyti tinkamą filtrą kiekvienam procesui [10]. Taip pat gauti išfiltruoti duomenys toliau gali būti naudojami kitų metodų, kaip autoasociatyvinių neuroninių tinklų, išgaunant informaciją apie kitus proceso būsenos kintamuosius.

### 1.2.3. Adaptyvūs stebėtojai

Adaptyvūs stebėtojai (angl. adaptive observers) yra algoritmai skirti nustatyti tiesiogiai neišmatuojamus ar sunkiai išmatuojamus parametrus. Šie stebėtojai naudodami realius jutiklių duomenis ir tam tikrus matematinius modelius dinamiškai koreguoja savo parametrus taip pateikdami informaciją apie stebimą būsenos kintamąjį. Adaptyvus stebėtojas taip pat gali įvertinti nežinomus proceso parametrus kaip kinetiką bei pateikia spėjimo paklaidas, kuriomis remiantis galima spręsti ar duomenimis galima pasitikėti [11].

Tiesinės vieno įėjimo ir išėjimo sistemos lygtys pateiktos 9 ir 10 lygtyse. Čia  $a$  ir  $b$  reikšmės yra nežinomi sistemos parametrai, o matricų  $\bar{A}$  ir  $c'$  vertės yra žinomos [12].

$$\dot{x} = [-a|\bar{A}]x + bu \quad (9)$$

$$y = c'x = x_1 \quad (10)$$

Projektuojant adaptyvų stebėtuvą tikslas būtų stebėti sistemos būseną  $x$  bei nežinomus parametrus  $a$  ir  $b$ . Todėl pilno adaptyvaus stebėtuvo lygtis pateikta 11 lygtyje. Čia  $\hat{a}(t)$  ir  $\hat{b}(t)$  yra nuspėjamos parametrų  $a$  ir  $b$  vertės,  $\hat{x}$  – nuspėjama  $x$  vertė, o  $w^1(t)$  ir  $w^2(t)$  yra išoriniai signalai, stabilizuojantys adaptyvaus stebėtuvo būseną [12].

$$\dot{\hat{x}} = K\hat{x} + [k - \hat{a}(t)]x_1(t) + \hat{b}(t)u(t) + w^1(t) + w^2(t) \quad (11)$$

Adaptyviųjų stebėtojų generuojami duomenys taip pat gali būti naudojami kaip įvestis kitiems, neuroniniais tinklais grįstiems, modeliams. Tokiu būdu sistemose, kurioms nėra sudaryto aiškiai apibrėžto matematinio modelio, tampa įmanoma prognozuoti būsenos kintamuosius [13]. Be to, adaptyviųjų stebėtojų duomenų integravimas su kitais būsenos kintamųjų vertinimo metodais sudaro galimybę ne tik nustatyti papildomų būsenos kintamųjų vertes, bet ir padidinti jau įvertintų kintamųjų tikslumą.

### 1.2.4. Neuroniniai tinklai

Dirbtiniai neuroniniai tinklai yra itin efektyvus įrankis biologinių procesų būsenos kintamųjų nustatymui. Jie suteikia duomenimis grįstą metodą sudėtingų biocheminių sistemų stebėsenai ir valdymui. Neuroniniai tinklai geba modeliuoti netiesines priklausomybes tarp netiesioginių jutiklių signalų, tokių kaip optiniai ar fluorescenciniai matavimai, ir pagrindinių proceso kintamųjų, pavyzdžiui, biomasės, substrato bei produkto koncentracijų [14].

Be nuspėjamų kintamųjų tikslumo gerinimo, neuroniniais tinklais grįsti metodai sumažina priklausomybę nuo ne-realaus laiko parametrų matavimo. Tai ženkliai atpigina visą procesą, nes

naudojama mažiau brangių laboratorinių matavimų bei suteikia galimybę gauti duomenis realiu laiku, kur tiesioginiai matavimai būtų sudėtingi arba su ženkliu vėlinimu [15].

Stebint procesų kinetiką neuroniniai tinklai taip pat yra naudingi vertinant biomasės specifinius augimo parametrus ir aptinkant pokyčius proceso kinetiniuose modeliuose. Naudojant neuroninius tinklus galima atpažinti ir parinkti tinkamus kinetinius modelius atitinkamiems būsenos kintamiesiems. Šiuo metodu galima ne vien tik nustatyti procesui tinkantį kinetinį modelį, bet ir identifikuoti perėjimus nuo vieno kinetinio modelio prie kito [16]. Šioje vietoje galima pastebėti didelį neuroninių tinklų potencialą sudėtingų biologinių sistemų valdyme ir modeliavime.

### **1.2.5. Hibridiniai modeliai**

Hibridiniai modeliai biotechnologinių procesų būsenos kintamųjų nustatyme reprezentuoja sudėtingą metodą, kuris apjungia mechanistinį supratimą su duomenimis grįstais metodais pagerinant biologinių procesų nuspėjamumą ir efektyvumą. Integruojant parametrinius modelius, kurie naudoja žinomus mokslinius duomenis, su neparimetriniais modeliais, kurie naudoja empirinius duomenis, hibridiniai modeliai geba įvertinti sudėtingas ir netiesines dinamikas dažnai surandamas biotechnologinėse sistemose [17, 18].

Tradicinių modeliavimo technikų trūkumus stengiamasi pašalinti su hibridiniais modeliais, fokusuojantis į kitų modelių stipriąsias vietas. Tokiu būdu hibridiniai modeliai gali pagerinti ekstrapoliavimo galimybes, sumažinti duomenų reikalavimus bei pagerinti bendrą aiškumą [19]. Šie modeliai gali pranokti įprastinius metodus įvairiuose scenarijuose, ypač dinaminėse sistemose, kurių veikimo mechanizmai nėra pilnai suprantami [20].

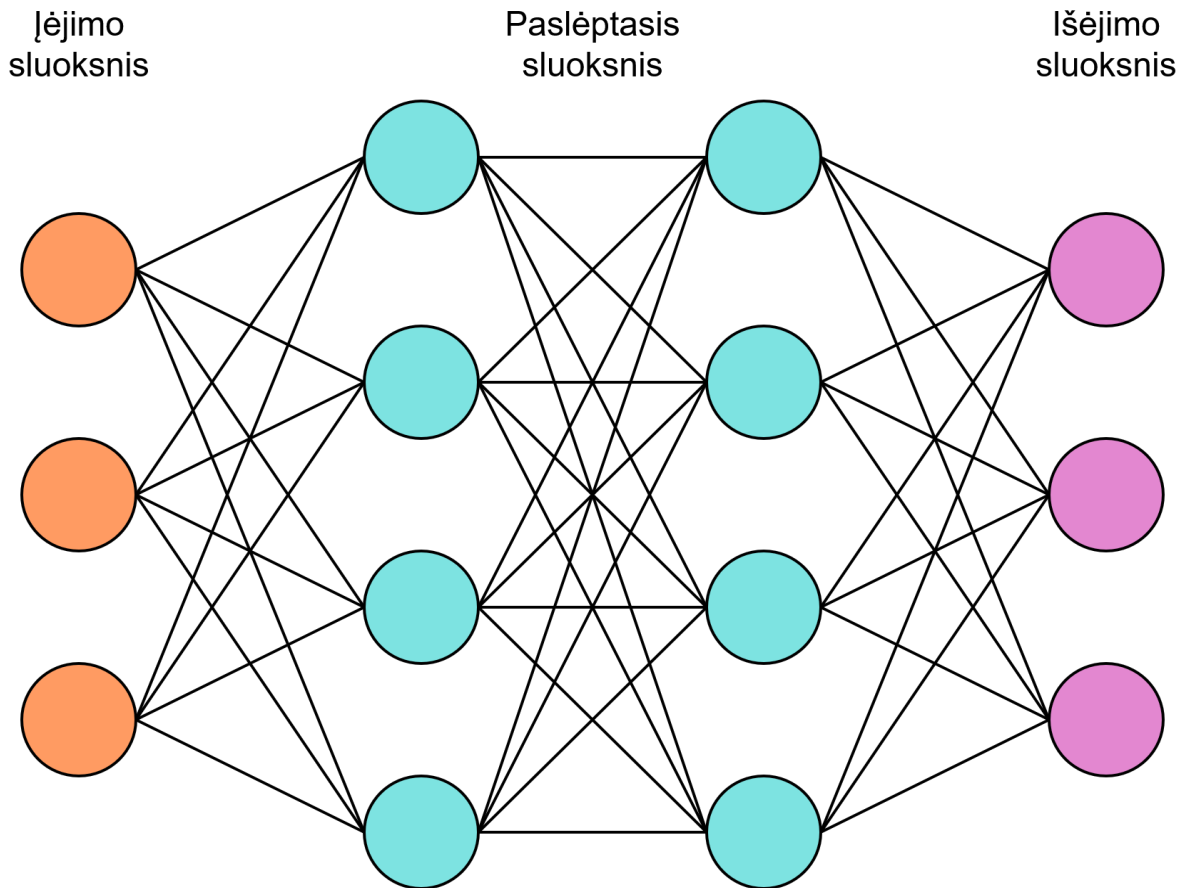
Pastaruoju metu vis didesnio dėmesio sulaukia hibridiniai būsenos kintamųjų vertinimo modeliai. Vienas dažniausiai taikomų sprendimų yra Kalmano filtrų ir įvairių tipų neuroninių tinklų integravimas į bendrą sistemą. Toks metodų derinimas leidžia pasiekti didesnę būsenos kintamųjų nustatymo tikslumą, palyginti su atvejais, kai Kalmano filtrai ar neuroniniai tinklai taikomi atskirai [9].

## **1.3. Dirbtiniai neuroniniai tinklai**

Dirbtiniai neuroniniai tinklai yra mašininio mokymosi klasė paremta žmogaus smegenų struktūra bei funkcionalumu, gebanti atpažinti šablonus bei atlikti sprendimus įvertinus duomenis. Šių tinklų architektūra susideda iš kelių pagrindinių komponentų, kurie veikdami kartu apdoroja duomenis.

### **1.3.1. Neuroninių tinklų komponentai**

Tam, kad geriau suprasti pagrindinius neuroninių tinklų elementus galima remtis 1 paveikslu, kuriame pateikta principinė neuroninio tinklo schema.



**1 pav.** Principinė neuroninio tinklo schema

Neuroninio tinklo pagrindą sudaro neuronai, kurie yra individualios dalelės, atkartojančios biologinių neuronų elgseną. Kiekvienas neuronas gauna įėjimo vertę, ją apdoroja ir išduoda išėjimo vertę. Dirbtinis neuronas veikia surinkdamas įėjimų vertes, pritaikydamas jiems matematinę funkciją ( $f(X)$ ) ir sugeneruodamas išėjimo vertę priklausomą nuo matematinių skaičiavimų rezultatų. Kiekvieno neuroso aktyvacija priklauso nuo specialių sąlygų, vadinamų aktyvacijos funkcijomis [21].

Neuronai yra suskirstyti į sluoksnius, įprastai susidedančius iš įėjimo sluoksnio, paslėptųjų sluoksnių bei išėjimo sluoksnio. Kiekviename sluoksnyje yra neuronų, kurie atlieka informacijos perdavimą per visą tinklą. Įėjimo sluoksnis gauna neapdorotus duomenis, paslėptieji sluoksniai vykdo įvairias duomenų transformacijas bei mokymo procesus, o išėjimo sluoksnis sugeneruoja galutinius rezultatus. Neuroniniai tinklai gali turėti keletą paslėptųjų sluoksnių, kurie pagerina sąsajų duomenyse aptikimą [22].

Kiekviena jungtis tarp neuronų turi jai priskirtą svorį, kuris simbolizuoja įėjimo svarbą neuroso išėjimo signalui. Papildomai, kiekvienas neuronas turi nuokrypį (angl. bias), kuris pridedamas prie įėjimų sumos. Šių svorių ir nuokrypų koregavimas apmokymo metu yra būtinas norint turėti tinkamai veikiančią neuroninį tinklą [22].

Bendruoju atveju kiekvieno neurono išėjimas gali būti nusakytas 12 lygtimi [23]. Čia  $y_i$  –  $i$  neurono išėjimo vertė,  $n$  – į neuroną ateinančių jungčių kiekis,  $x_k$  ir  $K_k$  – kiekviena ateinanti vertė ir jos svoris,  $\alpha_i$  – neurono nuokrypis.

$$y_i = \sum_{k=1}^n (x_k K_k) + \alpha_i \quad (12)$$

Aktyvavimo funkcijos neuroniniuose tinkluose turi didelę svarbą, nes į modelį įveda netiesiškumą. Jos yra taikomos neuronų įėjimų ir nuokrypos sumai, taip įgalinant tinklą išmokti sudėtingus modelius. Naudojamos įvairios aktyvavimo funkcijos priklausomai nuo norimo modelio [24].

Pastaruoju metu vis dažniau naudojamas gilių neuroninių tinklų terminas. Giliais neuroniniais tinklais vadinami tokie neuroniniai tinklai, kurie savo paslėptuosiuose sluoksniuose turi daugiau nei tris neuronų sluoksnius [25].

### 1.3.2. Apmokymo algoritmai

Grižtamoji sklaida (angl. backpropagation) yra vienas iš pagrindinių neuroninių tinklų apmokymo algoritmų skirtų sumažinti modelio paklaidą. Jis veikia grąžinant paklaidos vertes per tinklą iš galo į priekį. Taip yra parenkami optimalūs neuronų svoriai, todėl neuroninis tinklas išmokomas priskirti tam tikras įėjimo vertes prie norimų išėjimo verčių. Šitas metodas automatizuoja mokymo procesą, įgalinant modelį keisti savo parametrus pagal veiklos rodiklius ir tobulėti laikui bėgant. Grižtamosios sklaidos tikslas yra pasiekti tokią būseną, kurioje tinklas gali išduoti teisingus išėjimus visiems duotiesiems įėjimams [26].

Konstruktivaus mokymo (angl. constructive learning) algoritmai yra sukurti optimizuoti neuroninių tinklų konfigūracijas efektyviam modelių klasifikavimui. Šie algoritmai dinamiškai keičia tinklo struktūrą pagerinant duomenų panašumų aptikimo galimybes [27]. Šioje kategorijoje galima išskirti genėjimo (angl. pruning) algoritmus, kurių tikslas yra sumažinti neuroninių tinklų dydį ženkliai nekeičiant jų efektyvumo. Naujas genėjimo metodas įvertina neuroninį sudėtingumą nuspręsdamas kuriuos ryšius reikia pašalinti, leidžiant tinklui išlaikyti išmoktą elgseną ir sumažinant jo sudėtingumą [28].

### 1.4. Autoasociatyviniai neuroniniai tinklai

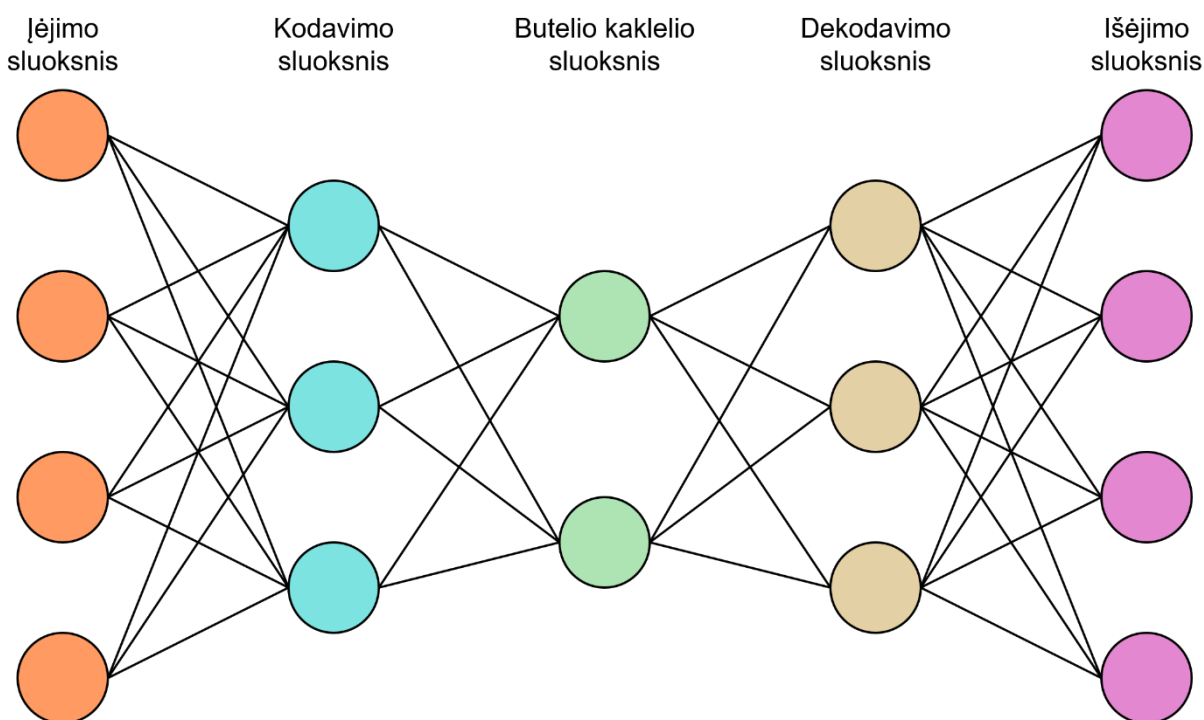
Autoasociatyviniai neuroniniai tinklai yra specializuota dirbtinių neuroninių tinklų atšaka, skirta išmokti atkartoti įėjimo duomenis ir sukurti sąsajas tarp įėjimų bei išėjimų. Tai yra galima dėl išskirtinės šių neuroninių tinklų architektūros. Autoasociatyviniai neuroniniai tinklai puikiai tinka anomalijų aptikimo, duomenų atkūrimo bei požymių išskyrimo užduotims atlikti.

Šaltiniuose autoasociatyviniai neuroniniai tinklai dar vadinami autoenkoderiais. Struktūra autoenkoderiai yra panašūs autoasociatyviniams neuroniniams tinklams, nes dažnu atveju viduriniuose sluoksniuose turi mažesnę kiekį neuronų. Tai abejiems modeliams leidžia suspausti duomenis ir taip atskirti tik svarbiausias duomenų vietas [29]. Todėl abejais terminais nusakomi panašios struktūros, tas pačias užduotis atliekantys modeliai.

#### 1.4.1. Autoasociatyvinių neuroninių tinklų architektūra

Autoasociatyvinių neuroninių tinklų architektūra skiriasi nuo tradicinės neuroninio tinklo architektūros. Tai leidžia neuroniniam tinklui efektyviau sukurti asociacijas tarp įėjimo ir išėjimo verčių. Tipinė autoasociatyvinio neuroninio tinklo struktūra pavaizduota 2 paveiksle. Įprastai šie

neuroniniai tinklai susideda iš įėjimo, kodavimo, butelio kaklelio, atkodavimo ir išvesties sluoksnių. Sudėtingesnėms užduotims atlikti gali būti įvedami papildomi sluoksniai [30].



**2 pav.** Autoasociatyvinio neuroninio tinklo architektūra

Įėjimo sluoksnis yra pirmasis tinklo komponentas, atsakingas už gaunamų duomenų priėmimą. Šiame sluoksnyje kiekvienas neuronas atitinka tam tikrą įėjimo kintamąjį. Taip pat įėjimo sluoksniui identiškas yra išėjimo sluoksnis. Todėl, autoasociatyvinis neuroninis tinklas išduoda tokios pačios struktūros vektorius kaip gavo įėjime, tik jo vertės būna filtruotos [30].

Kodavimo sluoksnis konvertuoja įėjimo duomenų vertes į kitą būsenos erdvę. Šio sluoksniui neuronai gauna vertes iš įėjimo sluoksniui, atlieka su jais skaičiavimus ir atiduoda vertes tolimesniam sluoksniui. Dekodavimo sluoksnis atvirkščiai atlieka kodavimo sluoksniui funkciją. Abiejų sluoksnių aktyvacijos funkcijos turi būti vienodos ir įprastai būna netiesinės [31].

Butelio kaklelio sluoksnis būtinai turi turėti mažiau neuronų nei kiti sluoksniai. Tai papildomai suspaudžia ir užkoduoja įėjimų vertes, panaikinant nereikalingas koreliacijas tarp duomenų. Aktyvacijos funkcijos šiam sluoksniui gali būti tiek tiesinės, tiek netiesinės [31].

#### **1.4.2. Konvoliuciniai neuroniniai tinklai**

Konvoliuciniai neuroniniai tinklai – tai modeliai daugiausiai naudojami vaizdų atpažinimui bei klasifikavimui. Savo struktūra konvoliuciniai neuroniniai tinklai bei autoasociatyviniai neuroniniai tinklai turi daug panašumų, leidžiančių jiems suvokti ir apdoroti sudėtingas duomenų struktūras. Abu tinklų modeliai yra sukurti apdoroti daugiadimensinius įėjimus bei geba išskirti hierarchinę informaciją iš duomenų.

Konvoliuciniai neuroniniai tinklai įprastai susideda iš konvoliucinio, surinkimo (angl. pooling) ir pilnai sujungto (angl. fully connected) sluoksnių [32].

Konvoliucinis sluoksnis yra svarbiausias sluoksnis konvoliuciniuose neuroniniuose tinkluose. Įėjimo duomenims šis sluoksnis pritaiko nustatytus filtrus, taip išgaunant norimas detales. Kiekvienas filtras yra pritaikomas aptikti specifinius požymius kaip kraštus ar tekstūras (grafiniuose duomenyse). Po šio sluoksnio dažnai naudojamos netiesinės aktyvavimo funkcijos, kurios pagerina tinklą išmokti sudėtingus šablonus bei geriau atskirti norimus požymius [33].

Paprasčiausia konvoliucinė funkcija pateikta 13 lygtyje. Čia  $f$  ir  $g$  yra dvi funkcijos, o  $n$  yra pozicija arba išėjimo signalo laikas. Konvoliucijos operacija žymima  $*$  simboliu [34].

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n - m] \quad (13)$$

Po konvoliucinio sluoksnio naudojamas surinkimo sluoksnis, kuris skirtas sumažinti duomenų dimensijas. Taip išlaikoma svarbiausia duomenų informacija, tačiau sumažinamas bendras duomenų kiekis, sumažinant reikiamo skaičiavimo resurso kiekį [33].

Galiausiai konvoliucinio neuroninio tinklo pabaigoje naudojamas pilnai sujungtas sluoksnis, kuris atlieka galutinį duomenų klasifikavimą. Šio sluoksnio struktūra panaši įprastiniam neuroniniam tinklui, turinčiam kelis sluoksnius. Informacija iš prieš tai buvusių sluoksnių yra apdorojama kiekvieno šio sluoksnio neurono. Taip atliekamas galutinis duomenų įvertinimas pagal išmokus požymius [32].

Konvoliuciniai neuroniniai tinklai taip pat turi kelias variacijas savo architektūroje, kaip „AlexNet“, „VGG“ ar „Inception“. Šie pakeitimai pagerina modelių tikslumus tam tikrose aplikacijose [32].

Konvoliuciniai neuroniniai tinklai yra apmokinami prižiūrimo mokymo metodais. Tai yra, jų apmokymui naudojami sužymėti ir kategorizuoti duomenys. Dažniausiai mokymo procese naudojama nuostolių funkcija, kuri įvertina kiek arti yra modelio išėjimas prie realios vertės. Mokymo metu yra keičiami neuroninio tinklo parametrai, minimizuojant nuostolių funkciją [33].

Autoasociatyviniai neuroniniai tinklai ir konvoliuciniai neuroniniai tinklai yra panašūs požymių apmokymo būdais. Konvoliuciniai neuroniniai tinklai naudoja konvoliucinius sluoksnius iš duomenų išskirti norimus požymius bei surasti panašumus tarp duomenų. Panašiai autoasociatyviniai neuroniniai tinklai išmoksta aptikti tik svarbiausius duomenis. Todėl abu metodai skirti identifikuoti svarbiausias įėjimų vertes.

### 1.4.3. Autoasociatyvinių neuroninių tinklų apmokymas

Neuroninių tinklų apmokymui svarbu nusirodyti nuostolių funkciją. Iš principo tai gali būti bet kokia procesą nusakanti vertė, pagal kurią vertinama modelio kokybė. Praktikoje į nuostolių funkcijas yra stengiamasi įterpti visus svarbiausius išėjimo parametrus. Taip pat kartais nuostolių funkciją naudinga normalizuoti pagal įėjimo vertes, tokiu būdu galima paprasčiau apmokyti modelį [31].

Apsibrėžus nuostolių funkciją, pagal ją galima mokyti neuroninį tinklą. Modelio parametru optimizavimui dažnai naudojamas gradiento mažėjimo metodas, vertinantis nuostolių funkciją. Pagal tai yra keičiami modelio parametrai. Taip su kiekviena mokymo iteracija parenkami vis geresni modelio parametrai ir gaunamas geriau norimą užduotį atliekantis neuroninis tinklas.

Grįžtamosios sklaidos metodas yra naudojamas autoasociatyvinių neuroninių tinklų apmokymui. Apmokymo duomenys yra leidžiami nuosekliai neuroniniu tinklu, kuomet gautiems rezultatams yra pritaikoma nuostolių funkcija. Tuomet šis rezultatas yra grąžinamas iš tinklo galo į priekį. Taip

grįžtamosios sklaidos algoritmas pakoreguoja kiekvieną neuroninio tinklo svorį bei nuokrypį. Tokiu būdu vis keičiant neuroninio tinklo vertes minimizuojant nuostolių funkciją apmokomas autoasociatyvinis neuroninis tinklas [35].

#### **1.4.4. Autoasociatyvinių neuroninių tinklų variacijos**

Triukšmą mažinantis (angl. denoising) autoenkoderis yra specializuotas autoasociatyvinis neuroninis tinklas, skirtas pašalinti įėjimo duomenyse esantį triukšmą. Tai yra atliekama dėl kodavimo ir butelio kaklelio sluoksnių - per juos praleidus duomenis informacija yra suspaudžiama ir triukšmas išfiltruojamas, tuomet dekodavus šią informaciją gaunami duomenys be triukšmo [36].

Retas (angl. sparse) autoenkoderis pasižymi požymių išskyrimo galimybėmis. Tai yra pasiekama praretinant neuroninių tinklų kiekį paslėptuosiuose neuroninio tinklo sluoksniuose. Pastaruoju metu reti autoenkoderiai vis dažniau naudojami dideliuose kalbos modeliuose dėl galimybės efektyviai įvertinti naudojamą žodyną [37].

Abipusiai asociatyvios atminties (angl. bidirectional associative memory) neuroniniai tinklai išplečia vienkrypčių autoasociatyvinių neuroninių tinklų galimybes. Šie tinklai išmoksta asociacijas abiem kryptim, įgalinant juos tam tikram įėjimui išduoti reikiamą išėjimo vertę arba tam tikram išėjimui pateikti reikiamą įėjimo vertę. Tai pasiekama pakeičiant neuroninio tinklo parametrus pervedant į bipolinę formą, kuri leidžia tinklui dirbti abiem kryptim [38].

#### **1.4.5. Autoasociatyvinių neuroninių tinklų taikymas**

Viena iš autoasociatyvinių neuroninių tinklų taikymo sričių yra prarastų duomenų atkūrimas. Neuroniniai tinklais grįstas modelis buvo apmokytas stireno gamybos duomenimis. Gautas modelis gebėjo tiksliai atstatyti iki 3 pradingusių įėjimo kintamųjų. Taip pat bandymo metu buvo vertinta apmokymo duomenų imtis. Nustatyta, jog autoasociatyviniais neuroniniais tinklais grįstas modelis gali tiksliai atstatyti prarastus kintamuosius ir su sąlyginai maža apmokymo imtimi [30].

Kita taikymo sritis yra požymių išskyrimas. Autoasociatyviniai neuroniniai tinklai buvo naudojami linijos spektro išskyrimui iš povandeninio triukšmo. Nustatyta, kad naudojant šį metodą galima aiškiai išskirti reikiamą spektro informaciją iš visų duomenų žymiai efektyviau nei su tradiciniais metodais. Taip pat pastebėta, jog naudojant modelį su autoasociatyviniais neuroniniais tinklais išrenkant norimus požymius galima pašalinti stiprų pašalinį triukšmą [39].

Dar viena autoasociatyvinių neuroninių tinklų taikymo vieta yra anomalijų aptikimas. Šitoks modelis buvo naudotas anomalijų aptikimui Šiaurės jūros lygio stebėjimui. Nustatyta, jog šie tinklai gebėjo aptikti anomalijas duomenyse, kurios svarbios atmosferinių reiškinių nustatymui [40].

## 2. Metodinė dalis

### 2.1. Duomenų paruošimas

Tiriamajame darbe naudojami GFP sintetinio biotechnologiniame procese duomenys. GFP (angl. Green fluorescent protein) – tai natūraliai gamtoje randamas baltymas, paveiktas ultravioletinės šviesos skleidžiantis ryškia žalia fluorescencinę šviesą. GFP yra lengvai prisegamas prie įvairių baltymų, dėl to yra plačiai naudojamas įvairiose biotechnologijų srityse nustatant susintetinto baltymo kiekius fluorescencijos pagalba [41].

Gautų duomenų kintamųjų pavadinimai ir reikšmės pateikti 1 lentelėje. Didžioji dalis duomenų buvo įrašomi dažnai (kas 36 s), nes juos buvo galima tiesiogiai išmatuoti. Dalis duomenų, kaip gliukozės koncentracija ar biomasės koncentracija buvo matuota rečiau (kas 30 arba 60 min), nes šiems duomenims gauti reikia atlikti laboratorinius bandymus. Kadangi duomenys buvo gauti skirtingu dažnumu, norint sudėti į bendrą duomenų lentelę buvo pasirinkti laboratorinių duomenų proceso laikai.

1 lentelė. Duomenų kintamųjų pavadinimai ir reikšmės

Pavadinimas	Matavimo vnt.	Reikšmė
age	h	Proceso laikas
weight	kg	Kultivavimo terpės (su mikroorganizmais, substratais ir t.t.) svoris
acid	ml	Pateiktas rūgšties tūris
afoam	ml	Pateiktas putų slopinimo priemonės tūris
airf	slpm	Aeravimo oro srautas
bazė	ml	Pateiktas šarmo tūris
co2	vol%	CO <sub>2</sub> koncentracija ištekanciose aeravimo dujose
o2	vol%	O <sub>2</sub> koncentracija ištekanciose aeravimo dujose
pH	-	pH vertė
po2	%sat	Terpėje ištirpusio deguonies dalinis slėgis
press	mBar	Slėgis virš kultivavimo skysčio
stir	RPM	Maišymo greitis
temp	°C	Terpės temperatūra
tur	au	Terpės drumstumas
feed	g/h	Pamaitinimo greitis
jtemp_IN	°C	Įtekančio vėsinimo vandens temperatūra
jtemp_OUT	°C	Ištekancio vėsinimo vandens temperatūra
glucose	g/l	Išmatuota gliukozės koncentracija kultivavimo terpėje (laboratoriniai duomenys)
OD	-	Optinis terpės tankis (laboratoriniai duomenys)
BTM	g/l	Išmatuota biomasės koncentracija kultivavimo terpėje (laboratoriniai duomenys)
protein	rFu	Išmatuota tiksulinio baltymo koncentracija kultivavimo terpėje (laboratoriniai duomenys)
spec. prot	rFu/gX	BTM/protein (laboratoriniai duomenys)

### 2.1.1. Išvestinių dydžių skaičiavimas

Vien eksperimentinių duomenų neuroninio tinklo apmokymui neužtenka, nes dažnu atveju neuroninis tinklas neranda tinkamų korelacijų tarp duomenų. Todėl naudojantis turimais duomenimis apskaičiuojami papildomi išvestiniai kintamieji.

Pradėjus baltymų sintezę reikšmingas baltymų kiekis susidaro ne nuo sintezės pradžios, bet praėjus šiek tiek laiko. Tam, kad būtų aišku kada prasideda baltymų indukcija, buvo išvestas indukcijos laikas. Šis kintamasis pradeda skaičiuoti sintezės laiką tik nuo pirmo baltymų matavimo.

Pagal 14 lygtį apskaičiuotas pamaitinimo greitis tenkantis vienam kilogramui kultivavimo terpės.

Duomenys gaunami  $Fw = \frac{\text{kg}}{\text{h}}$ .

$$Fw = \frac{feed}{weight * 0,001} \quad (14)$$

Degunies įsisavinimo greitis apskaičiuotas pagal 15 lygtį. Čia  $MrO_2 = 32 \frac{\text{g}}{\text{mol}}$ ,  $N = 22,4 \frac{\text{g}}{\text{mol}}$ ,  $Co_{2in}$  – pradinė  $O_2$  koncentracija konkrečiam eksperimentui,  $Cco_{2in} = 0,03$ ,  $Co_{2out}$  –  $O_2$  koncentracija,  $Cco_{2out}$  -  $CO_2$  koncentracija. Duomenys gaunami  $OUR = \frac{\text{g}}{\text{h}}$ . Degunies įsisavinimo greitis nusako kaip greitai reaktoriuje esantys mikroorganizmai suvartoja deguonį dėl metabolinių procesų. Taip pat šis kintamasis gali parodyti kiek aktyviai veikia kultivavimo procesas. Šis kintamasis buvo integruojamas visą kultivavimo laiką, todėl ši reikšmė nusako bendrą mikroorganizmų suvartotą deguonies kiekį.

$$OUR = \frac{\frac{airf}{60} * MrO_2 * \left(1 + \frac{press}{1000}\right) * \frac{273}{(273 + temp) * N} * (Co_{2in} - Co_{2out} * \frac{100 - Cco_{2in} - Cco_{2out}}{100 - Cco_{2out} - Cco_{2out}})}{100} \quad (15)$$

Bendras biomasės svoris apskaičiuojamas 16 lygtimi. Kadangi kultivavimo terpės svoris kilogramais yra artimas terpės tūriui litrais, tai duomenys gaunami  $biomasė = g$ .

$$biomasė = BTM * weight \quad (16)$$

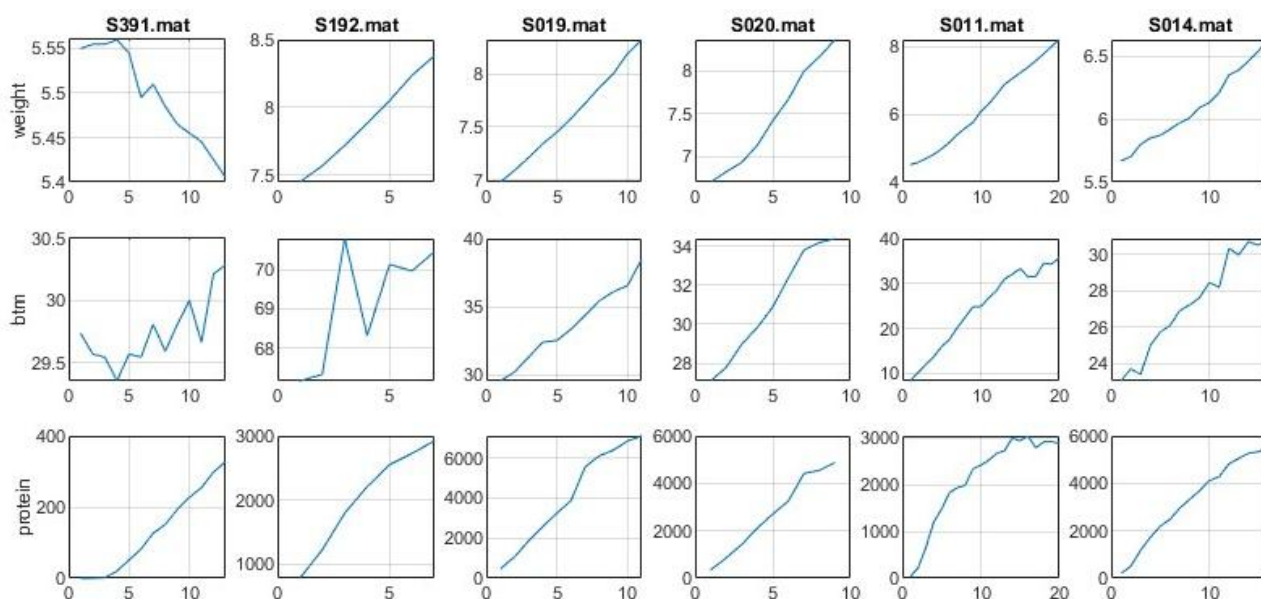
Anglies dioksido gamybos greitis apskaičiuotas pagal 17 lygtį. Čia  $MrCO_2 = 44 \frac{\text{g}}{\text{mol}}$ ,  $N = 22,4 \frac{\text{g}}{\text{mol}}$ ,  $Co_{2in}$  – pradinė  $O_2$  koncentracija konkrečiam eksperimentui,  $Cco_{2in} = 0,03$ ,  $Co_{2out}$  –  $O_2$  koncentracija,  $Cco_{2out}$  -  $CO_2$  koncentracija. Duomenys gaunami  $CPR = \frac{\text{g}}{\text{h}}$ . Kultivavimo metu mikroorganizmai išskiria anglies dioksidą dėl metabolinių procesų, todėl tai yra panašus kintamasis į deguonies įsisavinimo greitį. Taip pat šis kintamasis buvo integruojamas, kad būtų gautas per visą kultivavimo procesą pagamintas anglies dvideginio kiekis.

$$CPR = \frac{\frac{airf}{60} * MrCO_2 * \left(1 + \frac{press}{1000}\right) * \frac{273}{(273 + temp) * N} * (Cco_{2out} * \frac{100 - Cco_{2in} - Cco_{2in}}{100 - Cco_{2out} - Cco_{2out}} - Cco_{2in})}{100} \quad (17)$$

### 2.1.2. Netipinių duomenų atmetimas

Atliekant neuroninio tinklo mokymus su turimais duomenimis buvo pastebėta, jog tinklo rezultatai su tam tikrų eksperimentų duomenimis ženkliai skiriasi nuo rezultatų su likusiais duomenimis. Todėl buvo sukurta Matlab programa, atvaizduojanti pasirinktus duomenų kintamuosius.

Duomenų palyginimo pavyzdys pavaizduotas 3 paveiksle. Matoma, kad 391 eksperimento svorio ir biomasės koncentracijos vertės, o 192 eksperimento tik biomasės koncentracijos vertės išsiskiria lyginant juos su kitų eksperimentų duomenimis. Tam, kad apmokant neuroninį tinklą būtų efektyviau, visi šitie netipiniai duomenys yra atmetami.



3 pav. Duomenų palyginimas

Tokia duomenų atranka leidžia sumažinti netipinių ar klaidingų eksperimentų įtaką modelio mokymui. Pašalinus nekokybiškus duomenis, neuroninis tinklas gali tiksliau identifikuoti tikrąsias priklausomybes tarp kintamųjų ir sumažina persimokymo riziką.

### 2.1.3. Duomenų sujungimas

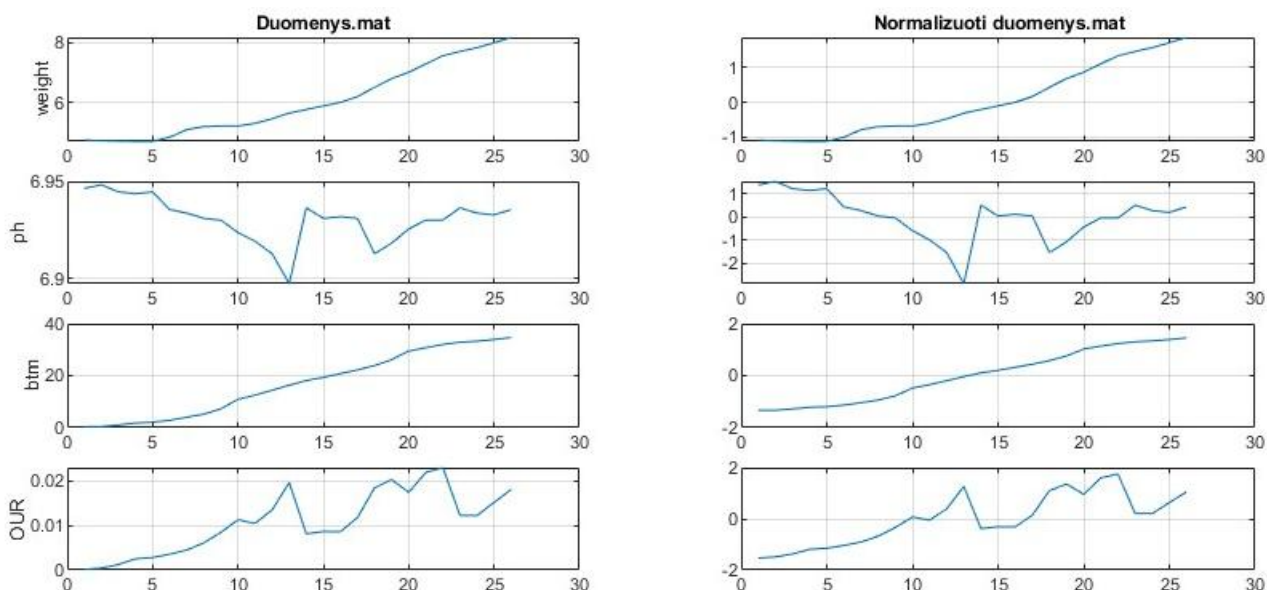
Kadangi eksperimentiniai duomenys buvo gauti skirtingais laiko intervalais ir jie buvo suskirstyti į dvi grupes, neuroninio tinklo apmokymui juos reikėjo sutraukti į vieną lentelę. Tam buvo paruošta Matlab programa, kuri nuskaito duotuosius duomenis bei sudeda juos į vieną bendrą duomenų lentelę kiekvienam eksperimentui. Taip pat ši programa suskaičiuoja išvestinius kintamuosius bei atmeta netipinių eksperimentų duomenis.

Iš viso buvo gauti 102 eksperimentų duomenys. Iš jų 16 eksperimentų buvo atmesti dėl kultivavimo metu atliktų klaidų, o dar 8 eksperimentai atmesti kaip netipiniai, nes jų kintamųjų kitimo pobūdis reikšmingai skyrėsi nuo likusios duomenų imties. Todėl tolimesniam neuroninių tinklų apmokymui ir validavimui buvo naudojami 78 eksperimentų duomenys.

### 2.1.4. Apmokymo duomenų normalizavimas

Norint apmokyti neuroninį tinklą reikia normalizuoti naudojamus duomenis. Normalizavimui naudota DSCALE.m paprogramė [42]. Ši paprogramė kiekvieną duomenų vektorių normalizuoja nulinio vidurkio bei standartinio nuokrypio funkcijomis. Tai neleidžia atskiriems duomenims iškreipti rezultatų dėl savo verčių.

Duomenų palyginimas su normalizuotais duomenimis pateiktas 4 paveiksle. Čia kairiame stulpelyje pavaizduoti originalūs duomenys, o dešiniame – normalizuoti. Matoma, kad originaliuose duomenyse duomenų verčių eilė stipriai skiriasi, o normalizavus duomenis jų visų reikšmės būna  $\pm 2$ . Tai leidžia kintamiesiems turėti vienodą svarbą neuroninio tinklo apmokyje.



4 pav. Duomenų palyginimas su normalizuotais duomenimis

## 2.2. Neuroninių tinklų apmokymas

Šiame darbe naudoti autoasociatyviniai neuroniniai tinklai buvo apmokyti naudojant Levenbergo – Marquardto algoritimą. Autoasociatyvinis neuroninis tinklas buvo naudojamas siekiant atkurti trūkstamojo kintamojo reikšmes remiantis kitais tarpusavyje susijusiais proceso kintamaisiais.

### 2.2.1. Neuroninio tinklo struktūra

Neuroninio tinklo struktūros parinkimas yra vienas svarbiausių etapų kuriant modelį, galintį tiksliai aproksimuoti nagrinėjamo biotechnologinio proceso priklausomybes. Autoasociatyvinio neuroninio tinklo atveju architektūra yra ypač svarbi, nes tinklas turi ne tik apdoroti įėjimo duomenis, bet ir išmokti kompaktišką jų vidinę reprezentaciją, iš kurios būtų galima atkurti pradinis duomenis.

Bendruoju atveju vieno neurono išėjimas gali būti aprašomas 18 lygtimi, kur  $x_i$  yra neurono įėjimo signalai,  $\omega_{ji}$  – svoriai,  $b_j$  – nuokrypis,  $f()$  – aktyvavimo funkcija, o  $y_j$  – neurono išėjimas. Ši lygtis parodo, kad neuronas pirmiausia suformuoja svertinę įėjimų sumą, o tuomet jai taiko aktyvavimo funkciją. Aktyvavimo funkcijos įveda netiesiškumą į modelį ir leidžia neuroniniams tinklams modeliuoti sudėtingas priklausomybes tarp kintamųjų [23].

$$y_j = f\left(\sum_{i=1}^n \omega_{ji}x_i + b_j\right) \quad (18)$$

Šiame darbe autoasociatyvinio neuroninio tinklo įėjimui buvo parinkti keturi proceso kintamieji, apie kurių parinkimą bus aprašyta plačiau. Kadangi autoasociatyvinis neuroninis tinklas yra mokomas atkurti savo įėjimo vektorius, išėjimo sluoksnyje taip pat naudojamas toks pat neuronų skaičius kaip ir įėjime.

Neuroninio tinklo struktūra buvo sudaryta iš keturių pagrindinių dalių: kodavimo sluoksnio, butelio kaklelio sluoksnio, dekodavimo sluoksnio ir išėjimo sluoksnio. Kodavimo sluoksnis transformuoja įėjimo duomenis į vidinę reprezentaciją, butelio kaklelio sluoksnis sumažina duomenų dimensiją, o dekodavimo sluoksnis siekia atkurti pradinę informaciją. Toks struktūrinis principas yra būdingas autoasociatyviniams neuroniniams tinklams ir leidžia efektyviai išskirti svarbiausius duomenų požymius.

Kodavimo ir dekodavimo sluoksniuose šiame darbe buvo naudojama 19 lygtyje pavaizduota hiperbolinio tangento aktyvavimo funkcija. Ši funkcija pasirinkta todėl, kad ji į modelį įveda netiesiškumą ir leidžia neuroniniam tinklui modeliuoti sudėtingas priklausomybes tarp proceso kintamųjų. Be to, hiperbolinio tangento funkcijos reikšmės yra apribotos intervale  $[-1, 1]$ , todėl ji yra tinkama normalizuotiems duomenims. Tokios funkcijos gali pagerinti mokymo stabilumą ir konvergavimo greitį [23].

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (19)$$

Butelio kaklelio sluoksnyje buvo naudojama 20 lygtyje aprašyta tiesinė aktyvavimo funkcija. Šis sluoksnis yra esminė autoasociatyvinio neuroninio tinklo dalis, nes jame vyksta duomenų suspaudimas. Butelio kaklelio sluoksnis priverčia tinklą išskirti svarbiausias įėjimo duomenų savybes ir pašalinti perteklinę informaciją. Tokia funkcija leidžia sumažinti duomenų dimensiją ir pagerinti modelio gebėjimą apibendrinti duomenis [29].

$$f(x) = x \quad (20)$$

Išėjimo sluoksnyje taip pat buvo naudojama tiesinė aktyvavimo funkcija, nes modelio tikslas buvo atkurti realias tęstines proceso kintamųjų reikšmes. Tiesinė funkcija neapriboja išėjimo intervalo ir yra dažnai naudojama regresiniuose neuroniniuose tinkluose [21].

Viso neuroninio tinklo veikimą galima aprašyti kaip kelių sluoksnių transformacijų kompoziciją, pateiktą 21 lygtyje. Čia  $x$  – įėjimo vektorius,  $y$  – neuroninio tinklo išėjimo vektorius,  $W_i$  – svorių matricos,  $b_i$  – nuokrypiai, o  $f_i$  – atitinkamų sluoksnių aktyvavimo funkcijos. Tokia struktūra leidžia modeliuoti sudėtingas netiesines priklausomybes tarp proceso kintamųjų.

$$y = W_4 f_3(W_3 f_2(W_2 f_1(W_1 x + b_1) + b_2) + b_3) + b_4 \quad (21)$$

Atliekant eksperimentus buvo nagrinėjamos skirtingos tinklo architektūros, keičiant neuronų skaičių kodavimo, butelio kaklelio ir dekodavimo sluoksniuose. Tokiu būdu buvo siekiama nustatyti optimalų modelio sudėtingumo ir tikslumo santykį. Didesnis neuronų skaičius didina modelio lankstumą, tačiau kartu gali didinti persimokymo riziką, kai modelis pradeda prisitaikyti prie triukšmo duomenyse [25, 28].

### 2.2.2. Neuroninio tinklo struktūros parinkimo principai

Neuroninio tinklo struktūros parinkimas šiame darbe buvo atliekamas eksperimentiškai, siekiant rasti optimalų modelio sudėtingumo ir tikslumo santykį. Nors teorijoje sudėtingesni neuroniniai tinklai gali modeliuoti sudėtingesnes priklausomybes, praktikoje per didelį neuronų skaičių dažnai nepadidina modelio tikslumo. Tai susiję su tuo, kad didesnio sudėtingumo modeliai yra labiau linkę prisitaikyti prie mokymo duomenų triukšmo, o ne prie bendrųjų priklausomybių tarp kintamųjų [43].

Pagrindinis kriterijus, pagal kurį buvo vertinama struktūra, buvo modelio gebėjimas tiksliai atkurti įėjimo duomenis bei stabiliai veikti su skirtingomis duomenų imtimis. Kadangi autoasociatyvinis neuroninis tinklas mokomas atkartoti savo įėjimą, tinklo struktūra turi būti pakankamai sudėtinga, kad galėtų užfiksuoti svarbiausias priklausomybes tarp kintamųjų, tačiau ne per daug sudėtinga, kad nepradėtų modeliuoti triukšmo. Tokia pusiausvyra tarp modelio sudėtingumo ir apibendrinimo gebėjimo yra vienas iš pagrindinių mašininio mokymosi uždavinių [44].

Atliekant struktūros parinkimą buvo keičiami kodavimo, butelio kaklelio ir dekodavimo sluoksnių neuronų skaičiai. Ypatingas dėmesys buvo skiriamas butelio kaklelio sluoksniui, nes būtent šiame sluoksnyje vyksta duomenų suspaudimas. Per mažas neuronų skaičius gali lemti informacijos praradimą, tuo tarpu per didelis – sumažinti suspaudimo efektą ir pabloginti modelio gebėjimą išskirti svarbiausias duomenų savybes. Literatūroje pažymima, kad tinkamai parinktas paslėptasis sluoksnis yra esminis veiksnys užtikrinant efektyvų autoenkoderių veikimą [45].

Taip pat svarbu pažymėti, kad neuroninio tinklo struktūros parinkimas buvo glaudžiai susijęs su naudojamų duomenų kiekiu. Kadangi šiame darbe naudojamas ribotas eksperimentinių duomenų kiekis, pernelyg sudėtingi modeliai gali būti netinkami dėl per didelio parametrų skaičiaus. Tokiu atveju modelis gali prisitaikyti prie mokymo duomenų, tačiau prasčiau veikti su naujais duomenimis. Šis reiškinys literatūroje apibūdinamas kaip persimokymas ir yra viena pagrindinių problemų taikant neuroninius tinklus praktikoje [43].

Praktiniu požiūriu struktūros parinkimas buvo atliekamas bandymų keliu, testuojant skirtingas neuroninio tinklo konfigūracijas ir vertinant jų rezultatus pagal suminę kvadratinę paklaidą. Tokia metodika leidžia objektyviai palyginti skirtingas architektūras ir pasirinkti tinkamiausią sprendimą konkrečiam uždaviniui. Panašus metodas dažnai taikomas eksperimentiniuose tyrimuose, kai optimalūs modelio parametrai nėra iš anksto žinomi [44].

Apibendrinant galima teigti, kad neuroninio tinklo struktūros parinkimas šiame darbe buvo grindžiamas ne tik teorinėmis prielaidomis, bet ir eksperimentiniais rezultatais, leidusiais nustatyti optimalų modelio sudėtingumą.

### **2.2.3. Autoasociatyvinio neuroninio tinklo apmokymas taikant Levenbergo – Marquardo algoritmą**

Apmokyti autoasociatyvinį neuroninį tinklą buvo pasirinktas Levenbergo – Marquardo algoritmas [46]. Mokymo metu tinklas sudaro ryšius tarp kintamųjų, todėl yra labai svarbu mokymo duomenims parinkti tuos kintamuosius, kurie turi sąryšį tarpusavyje. Kitu atveju, neuroninis tinklas neras sąryšio tarp duomenų ir duos blogus rezultatus.

Levenbergo – Marquardo algoritmas yra vienas iš plačiausiai naudojamų skaitinių metodų sprendžiant netiesinių mažiausių kvadratų optimizavimo uždavotus. Šio algoritmo tikslas – efektyviai ir stabiliai surasti tokius modelio parametrus, kurie kuo tiksliau atitiktų eksperimentinius duomenis. Dėl savo efektyvumo jis yra plačiai naudojamas duomenų aproksimacijoje, netiesinėje regresijoje bei neuroninių tinklų mokyme.

Algoritmo pagrindinė dalis aprašoma kaip klaidos funkcija. Jeigu turimi duomenys aprašomi  $(t_j, y_j)$ , o netiesinis modelis aprašomas kreive  $f(t|p)$ , tuomet kreivės parametrai  $p$  yra optimizuojami taip, kad išvestinių kvadratų suma

$$S(p) = \sum_{i=1}^m [y_i f(t|p)]^2 \quad (22)$$

būtų mažiausia.

Kadangi funkcija  $f$  yra netiesinė, jos minimumo negalima rasti tiesiogiai. Todėl Levenbergo – Marquardto algoritmas yra iteracinė procedūra, kiekvienoje iteracijoje koreguojanti parametrus. Pradedant minimizaciją būtina pateikti pradines reikšmes parametrų vektoriui  $p$ , kurios įprastai yra parenkamos atsitiktiniu būdu. Kiekvienai iteracijai parametrų vektorius  $p$  yra pakeičiamas naujai apskaičiuojamu vektoriumi  $p + q$ . Atliekant šiuos skaičiavimus naudojamas Gauso – Niutono metodas, kuris daro prielaidą, jog aplink sprendimo tašką liekanų funkciją galima aproksimuoti tiesiškai [47]. Nustatant  $q$ , funkcijos  $f_i(p + q)$  aproksimuojamos pagal jų linearizacijas

$$f(p + q) \approx f(p) + J(q) \quad (23)$$

kur  $J$  yra funkcijos  $f$  su parametru  $p$  Jokūbo matricos determinantas, sudaromas iš daug kintamųjų funkcijos dalinių išvestinių.

Kvadratų sumos  $S$  minimumas yra  $\nabla_q S = 0$ . Tuomet lygtyje apskaičiavus dešinės pusės išvestinę ir ją prilyginus nuliui gaunama

$$(J^T J)q = -J^T f \quad (24)$$

iš ko galima apskaičiuoti  $q$  transponuojant  $J^T J$ . Levenbergo – Marquardto algoritmo principas yra šios lygties modifikavimas įvedant slopinimo parametą  $\lambda$ :

$$(J^T J + \lambda I)q = -J^T f \quad (25)$$

Slopinimo koeficientas kontroliuoja algoritmo veikimą ir yra keičiamas kiekvienos iteracijos metu. Kuomet  $\lambda$  yra didelis, algoritmas elgiasi panašiai kaip gradientų nusileidimo metodas. Kuomet  $\lambda$  yra mažas, algoritmas artėja prie Gauso – Niutono metodo, kuris leidžia greitai konverguoti.

Po kiekvienos iteracijos apskaičiuojama nauja klaidos funkcijos reikšmė. Jeigu ji sumažėja, laikoma, kad žingsnis buvo sėkmingas, todėl slopinimo parametras  $\lambda$  sumažinamas. Jeigu klaida padidėja, parametrų atnaujinamas atmetamas, o  $\lambda$  padidinamas. Šis mechanizmas leidžia algoritmui adaptyviai koreguoti žingsnio dydį.

Geometriniu požiūriu šį algoritmą galima interpretuoti kaip judėjimą klaidos funkcijos paviršiumi. Kuomet paviršius yra netolygus, renkami maži žingsniai. Artėjant prie minimumo klaidos paviršius būna glotnesnis, todėl galima naudoti didesnius žingsnius tikslui pasiekti.

Autoasociatyvinio neuroninio tinklo apmokymui buvo parašyta Matlab programa, naudojanti Levenbergo – Marquardto algoritmą neuroninio tinklo parametrų radimui.

Šiame darbe autoasociatyvinis neuroninis tinklas taikomas pagal „sugedusio jutiklio“ principą. Jo esmė yra ta, kad tinklas pirmiausiai apmokomas su pilnu tarpusavyje susijusių proceso kintamųjų rinkiniu, kuriame visi kintamieji yra žinomi. Apmokymo metu tinklas mokomas atkurti savo įėjimo vektorius, todėl išmoksta vidines priklausomybes tarp pasirinktų proceso kintamųjų.

Vėliau, siekiant įvertinti pasirinktą būsenos kintamąjį, daroma prielaida, kad šio kintamojo matavimas yra neprieinamas arba jutiklis yra „sugedęs“. Tokiu atveju ieškomo kintamojo reikšmė

įėjimo vektoriuje nėra naudojama kaip patikima matavimo reikšmė, o tinklas ją atkuria remdamasis kitais tuo pačiu metu žinomais proceso kintamaisiais. Šiame darbe tokiu būdu atkuriamas biomasės kintamasis, o jo įvertinimui naudojami su biomasės kitimu susiję kintamieji: OUR integralas, bazė ir CPR integralas.

Taigi autoasociatyvinis neuroninis tinklas šiuo atveju veikia kaip programinis jutiklis: jis netiesiogiai įvertina biomasės reikšmę pagal kitus proceso metu registruojamus kintamuosius. Gauta atkurta biomasės reikšmė vėliau lyginama su realia, iš laboratorinių duomenų apskaičiuota biomasės reikšme, taip įvertinant modelio tikslumą.

Ši programa nuskaito apmokymo duomenis ir turi galimybę juos padalinti į apmokymo ir testavimo imtis. Visiems autoasociatyvinio neuroninio tinklo bandymams buvo naudojama ta pati po atmetimo likusi 78 eksperimentų duomenų imtis. Kiekvieno duomenų padalijimo metu 80 % šios imties buvo skiriama tinklo apmokymui, o 20 % - validavimui. Tokiu būdu apmokymui buvo naudojama apie 62 eksperimentų duomenys, o validavimui – 16 eksperimentų duomenys. Kadangi duomenų suskirstymas buvo atliekamas atsitiktine tvarka ir kartojamas kelis kartus, konkretūs į mokymo ir validavimo imtis patekę eksperimentai tarp pakartojimų galėjo skirtis.

Toliau programoje aprašoma norimo autoasociatyvinio neuroninio tinklo struktūra. Nurodoma kiek neuronų skiriama kiekvienam tinklo sluoksniui. Taip pat parenkami naudojami duomenų kintamieji bei nurodomas duomenų kintamasis, kurį neuroninis tinklas turės atkartoti.

Atliekant neuroninio tinklo mokymą Levenbergo – Marquardto algoritmu, neuroninis tinklas yra visuomet tikrinamas. Neuroniniui tinklui skiriami mokymo duomenys ir yra skaičiuojama vidutinė kvadratinė paklaida tarp tinklo išėjimo ir įėjimo reikšmės. Šis skaičius nusako kiek gautas rezultatas skiriasi nuo realios vertės. Ši suminė kvadratinė paklaida turėtų būti kuo įmanoma mažesnė.

Kadangi mokymo metu pradinės neuroninio tinklo svorių reikšmės yra sugeneruojamos atsitiktine tvarka, neuroninio tinklo apmokymo rezultatas kiekvienu bandymu skiriasi. Todėl neuroninis tinklas yra apmokinamas daug kartų. Geriausiai apmokytu neuroniniu tinklu laikomas tas, kurio suminė kvadratinė paklaida yra mažiausia.

Apmokytas tinklas validacijos metu naudojamas pagal „sugedusio jutiklio“ principą. Tai reiškia, kad vienas iš įėjimo vektorių kintamųjų laikomas nežinomu arba nepatikimu. Šiame darbe tokiu kintamuoju laikoma biomasė. Likę įėjimo vektorių kintamieji – OUR integralas, bazė ir CPR integralas – laikomi žinomais validacijos taško duomenimis.

Validacijos metu sugedusio jutiklio vertė yra laikoma optimizuojamu kintamuoju. Kiekvienam validacijos duomenų taškui sudaromas įėjimo vektorius, kuriame trys proceso kintamieji yra žinomi, o biomasės reikšmė iteratyviai keičiama optimizavimo algoritmo metu. Kiekvienos iteracijos metu sudarytas įėjimo vektorius perduodamas apmokytam autoasociatyviniam neuroniniam tinklui, o tinklo išėjimas lyginamas su pačiu įėjimo vektoriumi.

Tokiu būdu sprendžiamas optimizavimo uždavinys, kurio tikslas – rasti tokią biomasės reikšmę, kuriai bendra rekonstrukcijos paklaida tarp autoasociatyvinio neuroninio tinklo įėjimo ir išėjimo būtų mažiausia. Bendruoju atveju šį uždavinį galima užrašyti taip:

$$\hat{x}_s = \arg \min_{x_s} \sum_{i=1}^n (x_i - y_i)^2 \quad (26)$$

Čia  $\hat{x}_s$  – optimizavimo metu nustatyta sugedusio jutiklio reikšmė,  $x_s$  – ieškomas įėjimo vektoriaus kintamasis,  $x_i$  – autoasociatyvinio neuroninio tinklo įėjimo vektoriaus komponentai,  $y_i$  – tinklo išėjimo vektoriaus komponentai, o  $n$  – įėjimo ir išėjimo vektorių kintamųjų skaičius.

Šiame darbe naudojamoje programoje optimizavimas atliekamas taikant Matlab fminsearch funkciją. Pirmame validacijos taške pradinė ieškomojo kintamojo reikšmė parenkama pagal normalizuotą validacijos duomenų reikšmę. Kituose validacijos taškuose kaip pradinė reikšmė naudojama ankstesniame taške rasta optimali reikšmė. Taip užtikrinamas nuoseklus biomasės reikšmės atkūrimas validacijos duomenų sekoje.

Optimizavimo metu rastoji normalizuota biomasės reikšmė įrašoma į sugedusio jutiklio vietą įėjimo vektoriuje. Tuomet šis vektorius dar kartą perduodamas autoasociatyviniam neuroniniam tinklui, o gautas išėjimas naudojamas kaip atkurta biomasės reikšmė. Gauta reikšmė vėliau perskaičiuojama į pradines matavimo vienetų reikšmes ir palyginama su realia biomasės verte. Tokiu būdu autoasociatyvinis neuroninis tinklas veikia kaip programinis jutiklis, kuris trūkstamą arba nepatikimą proceso kintamąjį atkuria remdamasis kitais tuo pačiu metu žinomais proceso kintamaisiais ir apmokymo metu išmoktomis jų tarpusavio priklausomybėmis.

Galiausiai programa išduoda gauto validavimo rezultato suminę kvadratinę paklaidą. Grafiškai atvaizduojamos ieškomo kintamojo įėjimo ir išėjimo vertės.

#### 2.2.4. Vertinimo kriterijai

Modelių veikimo kokybei vertinti šiame darbe naudota absoliutinių nuokrypių suma tarp realių ir neuroninio tinklo atkurtų arba prognozuotų biomasės reikšmių. Papildomai apskaičiuotas vidutinis absoliutinis nuokrypis ir svartinė vidutinė absoliutinė procentinė paklaida. Mažesnės šių rodiklių reikšmės rodo tikslesnį modelio veikimą.

Vertinant modelių tikslumą buvo lyginamos realios biomasės reikšmės su neuroninio tinklo atkurtomis arba prognozuotomis reikšmėmis. Absoliutinių nuokrypių suma apskaičiuota pagal 27 lygtį:

$$S_{AE} = \sum_{i=1}^n |y_i - \hat{y}_i| \quad (27)$$

čia  $y_i$  – reali biomasės reikšmė,  $\hat{y}_i$  – neuroninio tinklo atkurta arba prognozuota biomasės reikšmė,  $n$  – validacijos duomenų reikšmių skaičius.

Vidutinis absoliutinis nuokrypis apskaičiuotas absoliutinių nuokrypių sumą padalijus iš validacijos duomenų reikšmių skaičiaus:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (28)$$

Absoliutinė procentinė paklaida šiame darbe apskaičiuojama kaip svartinė vidutinė absoliutinė procentinė paklaida (wMAPE), kuri gaunama absoliutinių nuokrypių sumą padalijus iš realių reikšmių sumos:

$$wMAPE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{\sum_{i=1}^n |y_i|} * 100 \% \quad (29)$$

Šis rodiklis parodo bendrą modelio nuokrypį procentais nuo realių biomasės reikšmių sumos. Kadangi šiame darbe prognozuojamas biomasės kiekis yra išreikštas gramais, absoliutinių nuokrypių suma ir vidutinis absoliutinis nuokrypis taip pat išreiškiami gramais, o absoliutinė procentinė paklaida – procentais.

Kadangi neuroninių tinklų apmokymas yra stochastinis procesas, kiekvieno apmokymo metu gaunami šiek tiek skirtingi rezultatai net ir naudojant tuos pačius duomenis ir modelio struktūrą. Dėl šios priežasties kiekvienai neuroninio tinklo konfigūracijai buvo atliekami keli apmokymo bandymai, o galutiniam modelių palyginimui naudotos vidutinės paklaidų rodiklių reikšmės.

Pagrindiniu vertinimo rodikliu laikomas absoliutinių nuokrypių sumos vidurkis, kuris leidžia įvertinti bendrą modelio nuokrypį validacijos duomenų aibėje. Papildomai naudotas vidutinis absoliutinis nuokrypis ir svartinė vidutinė absoliutinė procentinė paklaida, leidžiantys įvertinti ne tik absoliutų, bet ir santykinį modelio tikslumą.

Literatūroje pažymima, kad neuroninių tinklų rezultatų variacija tarp skirtingų apmokymo paleidimų yra natūrali ir susijusi su optimizavimo proceso netiesiškumu bei daugiamačiu parametru erdvės pobūdžiu [43]. Dėl šios priežasties statistinis rezultatų įvertinimas yra svarbi modelių analizės dalis.

Apibendrinant galima teigti, kad šiame darbe naudojamas vertinimo metodas leidžia pakankamai išsamiai įvertinti modelių tikslumą ir stabilumą bei pasirinkti tinkamiausią neuroninio tinklo struktūrą nagrinėjamam uždaviniui.

#### **2.2.5. Apmokymo duomenų kintamųjų parinkimas**

Autoasociatyvinio neuroninio tinklo apmokymui kintamieji parenkami atsižvelgus į biotechnologinio proceso kintamuosius bei stebint tinklo veiklą su parinktais kintamaisiais. Ypatingas dėmesys buvo skiriamas grafiniam duomenų atvaizdavimui, nes tai leidžia aiškiau suprasti tinklo gebėjimus aproksimuoti išėjimo duomenis. Analizuojant neuroninio tinklo rezultatus grafinėje formoje, galima įvertinti ar tinklas sėkmingai apmokytas.

Kintamųjų rinkinys buvo parinktas remiantis proceso fizikinių ir biologinių priklausomybių analize bei preliminariu modelio veikimo įvertinimu. Tolimesniuose tyrimuose naudotas tas pats kintamųjų rinkinys, siekiant užtikrinti rezultatų palyginamumą tarp skirtingų neuroninio tinklo struktūrų.

#### **2.2.6. Automatinis neuroninio tinklo apmokymas**

Automatiniam autoasociatyvinio tinklo apmokymui buvo sukurta Matlab programa, leidžianti nurodyti norimas autoasociatyvinio neuroninio tinklo struktūras ir jas automatiškai apmokyti. Gauti rezultatai įrašomi į lentelę, kad būtų patogų palyginti su kitais bandymais. Programa įrašo kiekvieno bandymo skaičiavimo validavimo suminės kvadratinės paklaidos vertę.

Eksperimentų metu visiems bandymams buvo naudoti iš anksto atsitiktine tvarka suskirstyti duomenys į mokymo ir validavimo imtis. Tarp bandymų duomenų paskirstymas nebuvo keičiamas, kad visi bandymai būtų atliekami su identiškais duomenimis. Norint atmesti priklausomybę nuo atsitiktinio duomenų parinkimo, viskas buvo kartojama 5 kartus, atsitiktinai perrenkant duomenų imtis.

### 3. Tyrimo rezultatai

#### 3.1. Autoasociatyvinis neuroninis tinklas

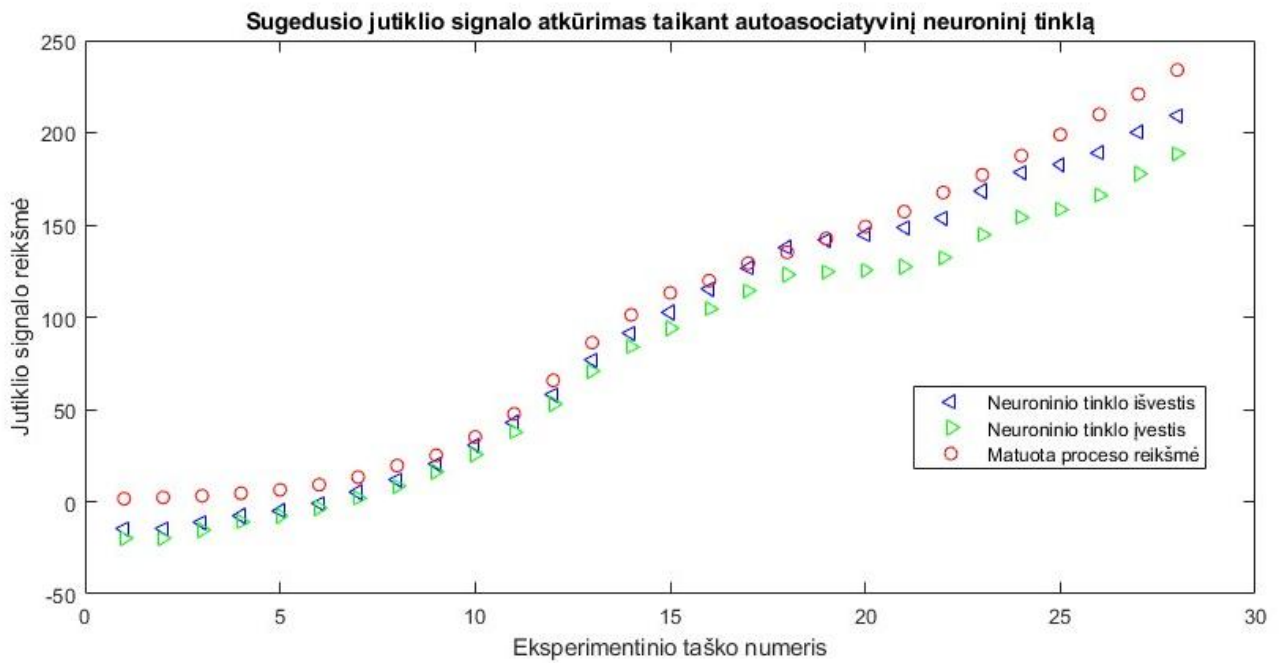
##### 3.1.1. Duomenų kintamųjų parinkimas

Autoasociatyvinio neuroninio tinklo apmokymui kintamieji buvo parenkami atsižvelgiant į biotechnologinio proceso fizikines ir biologines priklausomybes. Kadangi šiame darbe siekiama įvertinti proceso būsenos kintamąjį, pagrindiniu atkuriamu kintamuoju pasirinkta biomasė, išreikšta gramais. Šis kintamasis yra svarbus kultivavimo proceso eigai apibūdinti, nes nusako bendrą biomasės kiekį kultivavimo terpėje. Kadangi biomasė apskaičiuojama remiantis laboratoriniais biomasės koncentracijos matavimais ir kultivavimo terpės mase, jos netiesioginis įvertinimas pagal kitus proceso metu registruojamus kintamuosius yra aktualus uždavinys.

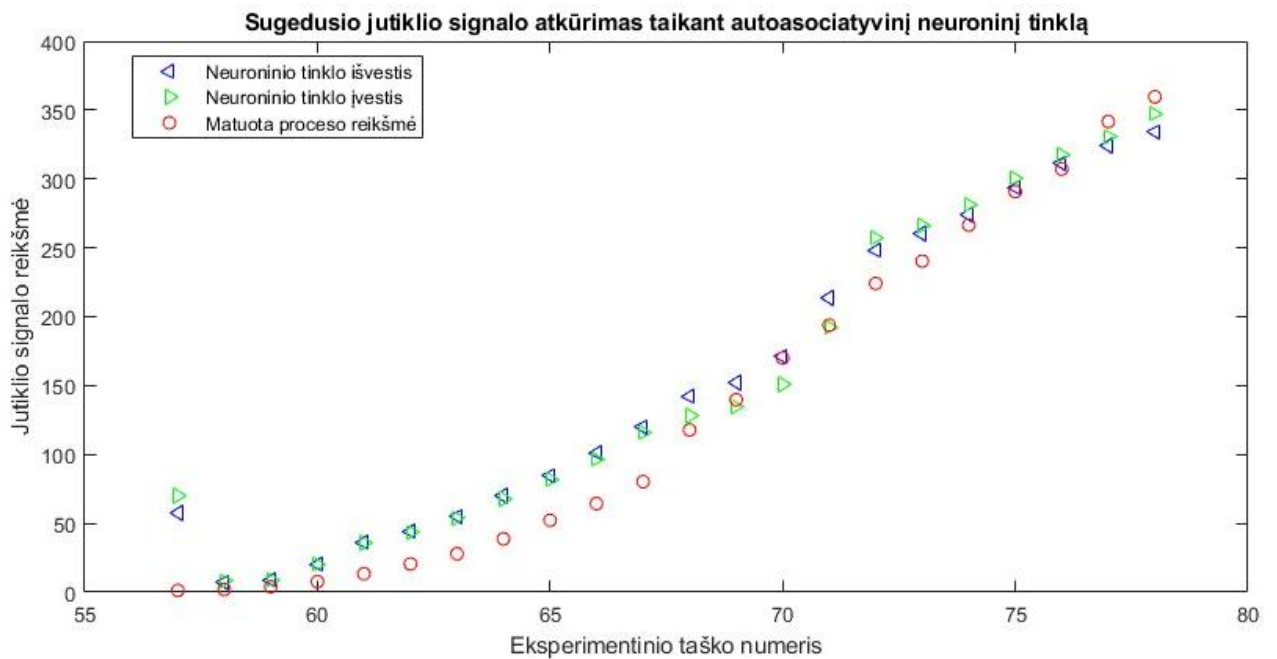
Autoasociatyvinio neuroninio tinklo įėjimo vektoriui parinkti šie kintamieji: biomasė, OUR integralas, bazė ir CPR integralas. OUR integralas ir CPR integralas atspindi bendrą mikroorganizmų metabolinį aktyvumą, kuris yra susijęs su biomasės augimu, o bazė netiesiogiai nusako terpės rūgštingumo koregavimo poreikį kultivavimo metu. Todėl šis kintamųjų rinkinys yra pagrįstas ne tik eksperimentiniu tinklo veikimo įvertinimu, bet ir biotechnologinio proceso pobūdžiu.

Taikant „sugedusio jutiklio“ principą, biomasės reikšmė laikoma atkuriamu kintamuoju, o jos įvertinimas atliekamas remiantis kitų proceso kintamųjų ir apmokymo metu išmoktomis tarpusavio priklausomybėmis.

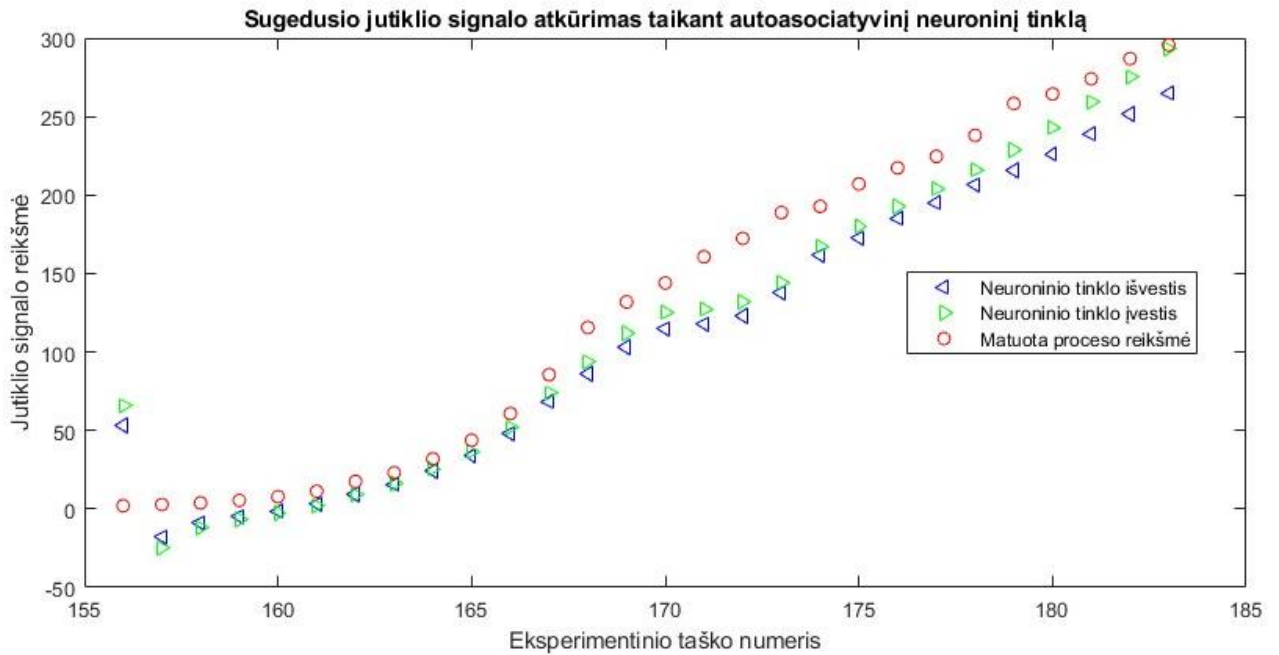
Gauti autoasociatyvinio neuroninio tinklo rezultatai, apmokus jį su pasirinktais kintamaisiais, pateikti 5–7 paveiksluose. Kadangi iš visų 78 eksperimentų rezultatų pateikiamas grafikuose apsinkintų jų analizę ir aiškumą, 5-7 paveiksluose pateikiama tik dalis validavimo rezultatų. Šiuose paveiksluose parodyti 3 eksperimentų iš 78 naudotos eksperimentų imties rezultatai. Šie grafikai skirti iliustruoti modelio veikimo pobūdį. Grafikuose apskritimais pavaizduota reali biomasės vertė, o trikampaiais – neuroninio tinklo atkurta biomasės vertė. Iš grafikų matoma, kad didžiąją proceso dalį neuroninio tinklo išėjimas yra artimas realioms biomasės vertėms, todėl pasirinktas kintamųjų rinkinys buvo naudotas tolimesniems autoasociatyvinio neuroninio tinklo struktūros tyrimams.



**5 pav.** Biomasės įverčiai, gauti naudojant autoasociatyvinį neuroninį tinklą (apmokymui naudoti kintamieji: biomasė, OUR integralas, bazė, CPR integralas)



**6 pav.** Biomasės įverčiai, gauti naudojant autoasociatyvinį neuroninį tinklą (apmokymui naudoti kintamieji: biomasė, OUR integralas, bazė, CPR integralas)



7 pav. Biomasės įverčiai, gauti naudojant autoasociatyvinį neuroninį tinklą (apmokymui naudoti kintamieji: biomasė, OUR integralas, bazė, CPR integralas)

### 3.1.2. Automatinis neuroninio tinklo apmokymas

Automatiniam autoasociatyvinio neuroninio tinklo apmokymui buvo sukurta Matlab programa, leidžianti lanksčiai apibrėžti įvairias neuroninio tinklo struktūras bei automatizuotai vykdyti jų apmokymo procesą. Programa suteikia galimybę sistemingai testuoti skirtingas architektūras ir jų parametrus, o gauti rezultatai automatiškai kaupiami lentelės pavidalu.

Siekiant padidinti rezultatų patikimumą ir sumažinti atsitiktinių veiksnių įtaką, kiekvienas automatinio apmokymo eksperimentas buvo kartojamas po 10 kartų. Toks pakartojimų skaičius leidžia geriau įvertinti modelio stabilumą, kadangi neuroninių tinklų mokymo procesas yra stochastinis, o gauti rezultatai gali reikšmingai svyruoti tarp skirtingų paleidimų. Kiekvieno bandymo metu programa fiksuoja validacijos duomenų aibės absoliutinių nuokrypių sumą, kuri naudojama kaip pagrindinis modelio veikimo kokybės rodiklis.

Eksperimentų metu visiems bandymams buvo naudojamas tas pats pradinis duomenų suskirstymas į mokymo ir validacijos imtis. Toks sprendimas pasirinktas siekiant užtikrinti eksperimentų palyginamumą – kiekviena tinklo struktūra buvo vertinama naudojant identiškus duomenis, todėl rezultatų skirtumai gali būti siejami tik su modelio architektūra ar mokymo proceso, o ne su duomenų paskirstymo variacija.

Vis dėlto, siekiant įvertinti rezultatų priklausomybę nuo atsitiktinio duomenų padalijimo, visas eksperimentų ciklas buvo papildomai pakartotas 5 kartus, kiekvieną kartą iš naujo atsitiktinai formuojant mokymo ir validacijos imtis. Tai leidžia gauti patikimesnį bendrą modelių įvertinimą ir sumažina riziką, kad rezultatai bus nulemti specifinio duomenų paskirstymo.

Iš viso autoasociatyvinio neuroninio tinklo struktūros tyrimui buvo nagrinėta 18 skirtingų vidinių sluoksnių konfigūracijų. Kiekviena konfigūracija buvo apmokoma po 10 kartų, o visas duomenų paskirstymo ciklas papildomai kartotas 5 kartus, atsitiktinai perrenkant mokymo ir validavimo imtis.

Taigi kiekvienai struktūrai buvo atlikta 50 apmokymo bandymų, o bendrai autoasociatyvinio neuroninio tinklo struktūros parinkimo tyrime atlikta 900 apmokymo bandymų.

Bandymų rezultatai pateikti 2 lentelėje, kurioje vaizduojama autoasociatyvinio neuroninio tinklo absoliutinės paklaidos sumos priklausomybė nuo pasirinktos neuroninio tinklo struktūros. Lentelėje pateikiami skirtingų architektūrų rezultatai leidžia įvertinti, kaip tinklo konfigūracija daro įtaką modelio tikslumui ir stabilumui.

Kadangi autoasociatyvinio neuroninio tinklo įėjimo ir išėjimo sluoksniuose naudojamas tas pats kintamųjų skaičius, šių sluoksnių neuronų skaičius visais atvejais buvo vienodas ir lygus keturiems. Todėl 2 lentelėje pateikiama tik kintanti vidinių sluoksnių konfigūracija, t. y. kodavimo, požymių ir dekodavimo sluoksnių neuronų skaičius. Pavyzdžiui, 2:2:2 konfigūracija lentelėje atitinka pilną autoasociatyvinio neuroninio tinklo struktūrą 4:2:2:2:4.

**2 lentelė.** Autoasociatyvinio neuroninio tinklo validacijos paklaidų priklausomybė nuo neuroninio tinklo struktūros

Kodavimo sluoksnių neuronų skaičius	Požymių sluoksnių neuronų skaičius	Dekodavimo sluoksnių neuronų skaičius	Absoliutinių nuokrypių sumos vidurkis, g	Vidutinis absoliutinis nuokrypis, g	Absoliutinė procentinė paklaida, %
2	2	2	4485.463	10.863	9.050
3	2	3	4782.942	11.579	9.653
4	2	4	4833.180	11.709	9.745
5	2	5	4925.832	11.942	9.927
6	2	6	4726.668	11.454	9.533
7	2	7	4697.302	11.381	9.470
8	2	8	4729.501	11.458	9.532
9	2	9	4504.154	10.905	9.085
10	2	10	4620.008	11.195	9.310
2	3	2	4844.137	11.741	9.768
3	3	3	4671.092	11.310	9.423
4	3	4	5376.316	13.035	10.848
5	3	5	6355.504	15.364	12.836
6	3	6	6279.468	15.225	12.668
7	3	7	8271.706	19.975	16.731
8	3	8	10007.821	24.229	20.175
9	3	9	9793.064	23.764	19.707
10	3	10	11655.991	28.168	23.585

Pagal 2 lentelėje pateiktus rezultatus matyti, kad geriausių rodiklius pasiekė neuroninis tinklas, kurio vidinių sluoksnių konfigūracija buvo 2:2:2, o pilna struktūra – 4:2:2:2:4. Šio modelio vidutinė absoliutinių nuokrypių suma buvo mažiausia ir siekė 4485,5 g, o vidutinis absoliutinis nuokrypis bei absoliutinė procentinė paklaida taip pat buvo mažiausios – atitinkamai 10,86 g ir 9,05 %. Tai rodo, kad ši tinklo struktūra tiksliausiai atkūrė nagrinėjamus duomenis ir pasižymėjo geriausiu

prognozavimo tikslumu iš visų tirtų variantų. Dėl šių priežasčių galima teigti, kad 2:2:2 konfigūracija buvo tinkamiausia nagrinėjamam uždaviniui.

Antrą geriausią rezultatą parodė 9:2:9 konfigūracijos neuroninis tinklas. Jo vidutinė absoliutinių nuokrypių suma buvo 4504,1 g, o vidutinis absoliutinis nuokrypis ir absoliutinė procentinė paklaida siekė atitinkamai 10,91 g ir 9,09 %. Šie rezultatai yra labai artimi geriausiai konfigūracijai, todėl šį modelį taip pat galima laikyti pakankamai tikslu. Vis dėlto ši tinklo konfigūracija yra ženkliai sudėtingesnė, nei geriausius rezultatus parodžiusi 2:2:2 konfigūracija.

Gautų rezultatų analizė leidžia pastebėti aiškia priklausomybę tarp neuroninio tinklo sudėtingumo ir jo veikimo kokybės. Nors iš pirmo žvilgsnio gali atrodyti, kad didesnis neuronų skaičius turėtų pagerinti modelio tikslumą, šiame tyrime pastebima priešinga tendencija. Didėjant neuronų skaičiui kodavimo ir dekodavimo sluoksniuose, daugeliu atvejų didėja vidutinė absoliutinių nuokrypių suma, vidutinis absoliutinis nuokrypis bei absoliutinė procentinė paklaida.

Toks rezultatas gali būti siejamas su tuo, kad sudėtingesni neuroniniai tinklai turi daugiau parametrų, todėl jiems gali reikėti didesnio duomenų kiekio arba kruopštesnio mokymo parametrų parinkimo. Galima daryti prielaidą, kad dalis sudėtingesnių modelių galėjo prasčiau apibendrinti validacijos duomenis, tačiau vien pagal pateiktus lentelės rezultatus persimokymo vienareikšmiškai patvirtinti negalima.

Mažiausią paklaidą turintis 2:2:2 neuronų konfigūracijos tinklas pasižymi paprasta struktūra, todėl geriau išskiria pagrindines duomenų priklausomybes, neįsimindamas triukšmo. Be to, šio tinklo paklaidų rodikliai yra mažiausi iš visų tirtų konfigūracijų, todėl galima teigti, kad ši struktūra tiksliausiai veikė pagal pasirinktus vertinimo kriterijus.

Tuo tarpu didesnių neuroninių tinklų, tokių kaip 8:3:8, paklaidų rodikliai yra ženkliai didesni. Pavyzdžiui, 8:3:8 konfigūracijos vidutinė absoliutinių nuokrypių suma siekia 10007,8 g, o absoliutinė procentinė paklaida – 20,18 %, todėl ši struktūra pagal pasirinktus vertinimo kriterijus yra mažiau tinkama. Tai rodo, kad tokie modeliai yra jautresni pradiniam parametrui ir mokymo duomenų variacijoms. Dėl šios priežasties jų naudojimas praktikoje būtų mažiau patikimas, net jei kai kuriais atvejais jie pasiekia panašų tikslumą.

Gauti rezultatai rodo, kad autoasociatyvinis neuroninis tinklas yra jautrus modelio struktūros pasirinkimui. Kai kuriais atvejais net ir nedidelis neuronų skaičiaus padidėjimas gali padidinti paklaidų rodiklius, todėl šio tipo modeliams būtina tinkamai parinkti modelio architektūrą.

Apibendrinant galima teigti, kad nagrinėjamam uždaviniui optimalus neuroninio tinklo sudėtingumas yra palyginti nedidelis. Tai rodo, kad tarp pasirinktų proceso kintamųjų egzistuoja pakankamai aiškios priklausomybės, kurių modeliavimui nereikia sudėtingų gilių architektūrų.

Analizuojant visus rezultatus galima pastebėti bendrą tendenciją, kad didėjant neuronų skaičiui tinklo struktūroje, modelio paklaida dažniausiai nemažėja, o kai kuriais atvejais net didėja. Tai rodo, kad sudėtingesnės architektūros šiuo atveju nesuteikia papildomos naudos, o priešingai – gali pabloginti modelio gebėjimą apibendrinti duomenis. Todėl galima daryti išvadą, kad nagrinėjamam uždaviniui pakanka sąlyginai paprastos neuroninio tinklo struktūros.

### 3.2. Tiesioginės sklaidos neuroninis tinklas

Tam, kad gautus rezultatus būtų galima palyginti su kitais metodais, buvo pasirinkta naudoti tiesioginės sklaidos neuroninį tinklą. Šio tinklo apmokymui taip pat taikytas anksčiau aprašytas Levenbergo – Marquardto algoritmas.

Remiantis ankstesnių eksperimentų su autoasociatyviu neuroniniu tinklu rezultatais, nustatyta, kad tolimesniam palyginimui tinkamas kintamųjų rinkinys yra biomasė, OUR integralas, bazė ir CPR integralas. Kadangi tiesioginės sklaidos neuroninis tinklas naudojamas vieno išėjimo kintamojo prognozavimui, biomasė buvo pasirinkta kaip tinklo išėjimas, o OUR integralo, bazės ir CPR integralo kintamieji – kaip tinklo įėjimai.

Skirtingai nei autoasociatyviniame neuroniniame tinkle, tiesioginės sklaidos neuroniniame tinkle negalima į įėjimą pateikti prognozuojamo kintamojo. Šiuo atveju ieškoma biomasė naudojama kaip tinklo išėjimas. Atitinkamai, į tinklo įėjimą pateikiami OUR integralo, bazės ir CPR integralo kintamieji, o tinklas mokomas prognozuoti biomasės reikšmę.

Kadangi įėjimo ir išėjimo kintamieji yra apibrėžti, neuroninio tinklo struktūra iš dalies yra iš anksto nustatyta: įėjimo sluoksnį sudaro trys neuronai, atitinkantys tris įėjimo kintamuosius, o išėjimo sluoksnyje yra vienas neuronas. Paslėptojo sluoksnio neuronų skaičius nėra iš anksto žinomas, todėl jis nustatomas eksperimentiškai, atliekant bandymus ir vertinant tinklo veikimo kokybę.

#### 3.2.1. Automatinis neuroninio tinklo apmokymas

Automatiniam tiesioginės sklaidos neuroninio tinklo apmokymui buvo sukurta atskira Matlab programa, kuri pagal iš anksto nurodytus parametrus automatiškai suformuoja neuroninio tinklo struktūrą, atlieka jo apmokymą ir gautus rezultatus įrašo į lentelę.

Kiekviena nagrinėta neuroninio tinklo struktūra buvo testuojama po 100 kartų. Toks pakartojimų skaičius pasirinktas siekiant sumažinti atsitiktinių veiksnių įtaką rezultatams ir patikimiau įvertinti kiekvienos architektūros veikimo stabilumą. Programa po kiekvieno bandymo į rezultatų lentelę įrašydavo gautą validacijos duomenų absoliutinių nuokrypių sumą, pagal kurią vėliau buvo apskaičiuojami vidutiniai paklaidų rodikliai.

Eksperimentų metu buvo naudojami tie patys iš anksto į mokymo ir validavimo imtis suskirstyti duomenys, kurie anksčiau buvo taikyti autoasociatyvini neuroninio tinklo apmokymui. Toks sprendimas leidžia tiesiogiai palyginti skirtingų neuroninių tinklų rezultatus, kadangi visi modeliai vertinami naudojant identiškus duomenų rinkinius. Siekiant sumažinti rezultatų priklausomybę nuo konkretaus atsitiktinio duomenų padalijimo, visas duomenų paskirstymas į atskiras imtis buvo pakartotas 5 kartus.

Tiesioginės sklaidos neuroninio tinklo atveju buvo tiriama 20 skirtingų tinklo struktūrų, keičiant paslėptojo sluoksnio neuronų skaičių nuo 1 iki 20. Kiekviena struktūra buvo apmokoma po 100 kartų, o duomenų paskirstymas į mokymo ir validavimo imtis buvo kartojamas 5 kartus. Taigi kiekvienai tiesioginės sklaidos neuroninio tinklo struktūrai buvo atlikta 500 apmokymo bandymų, o iš viso šio tinklo tyrime atlikta 10000 apmokymo bandymų.

Gauti rezultatai pateikti 3 lentelėje, kurioje parodyta tiesioginės sklaidos neuroninio tinklo absoliutinių nuokrypių rodiklių priklausomybė nuo neuroninio tinklo struktūros. Iš pateiktų rezultatų

matyti, kad geriausius rodiklius pasiekė 3:2:1 neuronų konfigūracijos tinklas. Šio modelio vidutinė absoliutinių nuokrypių suma buvo mažiausia ir siekė 3097 g, vidutinis absoliutinis nuokrypis buvo 7,49 g, o absoliutinė procentinė paklaida – 6,26 %. Tai rodo, kad ši tinklo struktūra pagal visus lentelėje pateiktus paklaidų rodiklius tiksliausiai atkūrė nagrinėjamus duomenis.

Artimus rezultatus taip pat parodė 3:1:1 struktūros neuroninis tinklas. Jo vidutinė absoliutinių nuokrypių suma siekė 3174,2 g, vidutinis absoliutinis nuokrypis buvo 7,68 g, o absoliutinė procentinė paklaida – 6,41 %. Nors šie rezultatai yra šiek tiek prastesni nei 3:2:1 konfigūracijos, skirtumas nėra didelis, todėl 3:1:1 struktūrą galima laikyti paprastesne, bet vis dar pakankamai tikslia alternatyva.

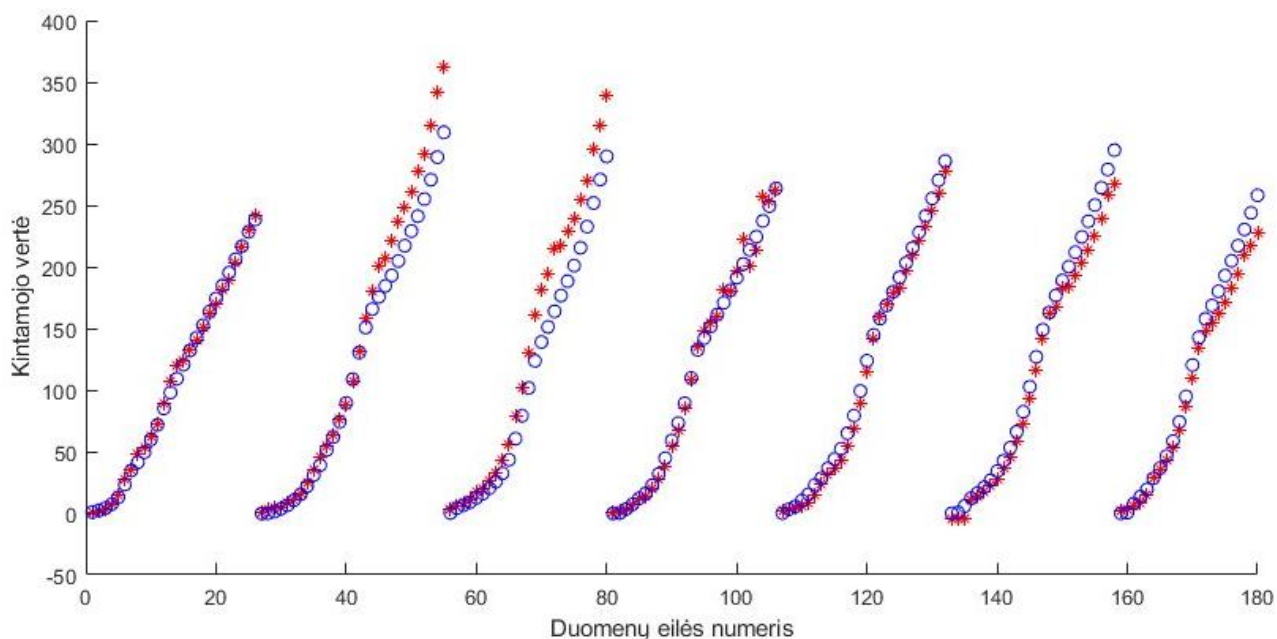
Skirtingai nei autoasociatyvinio neuroninio tinklo atveju, tiesioginės sklaidos neuroninio tinklo paklaidų rodikliai didėjant paslėptojo sluoksnio neuronų skaičiui kinta santykinai nedaug. Mažiausia procentinė paklaida siekia 6,26 %, o didžiausia – 6,83 %, todėl galima teigti, kad šio tinklo rezultatai yra mažiau jautrūs struktūros pokyčiams.

**3 lentelė.** Tiesioginės sklaidos neuroninio tinklo validacijos paklaidų priklausomybė nuo neuroninio tinklo struktūros

Iėjimo sluoksnio neuronų skaičius	Paslėptojo sluoksnio neuronų skaičius	Išėjimo sluoksnio neuronų skaičius	Absoliutinių nuokrypių sumos vidurkis, g	Vidutinis absoliutinis nuokrypis, g	Absoliutinė procentinė paklaida, %
3	1	1	3174.158	7.677	6.414
3	2	1	3097.047	7.491	6.258
3	3	1	3159.242	7.642	6.383
3	4	1	3248.602	7.858	6.564
3	5	1	3255.007	7.873	6.577
3	6	1	3271.797	7.914	6.611
3	7	1	3292.025	7.963	6.652
3	8	1	3318.809	8.027	6.706
3	9	1	3320.858	8.032	6.710
3	10	1	3303.238	7.990	6.674
3	11	1	3321.659	8.034	6.712
3	12	1	3312.119	8.011	6.692
3	13	1	3325.100	8.043	6.719
3	14	1	3332.871	8.061	6.734
3	15	1	3303.416	7.991	6.675
3	16	1	3334.209	8.065	6.737
3	17	1	3378.852	8.172	6.827
3	18	1	3370.347	8.152	6.810
3	19	1	3321.458	8.034	6.711
3	20	1	3345.349	8.092	6.759

Tiesioginės sklaidos neuroninio tinklo su 3:2:1 struktūra rezultatų grafinė vizualizacija pateikta 8 paveiksle. 8 paveiksle pateikiama tik dalis tiesioginės sklaidos neuroninio tinklo validavimo rezultatų, siekiant aiškiai parodyti realių ir modeliavimo prognozuotų biomasės reikšmių sutapimą.

Šiame grafike realūs proceso duomenys pavaizduoti raudonais žymekliais, o neuroninio tinklo prognozuotos reikšmės – mėlynais žymekliais. Iš grafiko galima matyti, kad modelio apskaičiuoti rezultatai gana tiksliai atkartoja realių duomenų kitimą bei geba sekti tiek lėtesnes, tiek spartesnes proceso kitimo tendencijas, todėl galima teigti, kad sudarytas neuroninis tinklas geba pakankamai gerai aproksimuoti nagrinėjamą procesą.



**8 pav.** Biomasis įverčiai, gauti naudojant tiesioginės sklaidos neuroninį tinklą

Palyginus tiesioginės sklaidos neuroninio tinklo rezultatus su autoasociatyvinio neuroninio tinklo rezultatais, galima pastebėti, kad tiesioginės sklaidos modelis pasiekė mažesnes paklaidų rodiklių reikšmes. Geriausio tiesioginės sklaidos tinklo absoliutinė procentinė paklaida siekė 6,26 %, o geriausio autoasociatyvinio neuroninio tinklo – 9,05 %. Tai rodo, kad tiesioginės sklaidos neuroninis tinklas šiame tyrime tiksliau atkūrė nagrinėjamus duomenis.

Taip pat pastebima, kad tiesioginės sklaidos neuroninio tinklo paklaidų rodikliai tarp skirtingų struktūrų skiriasi santykinai nedaug. Tai leidžia teigti, kad šio tipo tinklas yra mažiau jautrus pasirinktos struktūros pokyčiams nei autoasociatyvinis neuroninis tinklas.

### 3.3. Gautų rezultatų apibendrinimas

Šiame darbe buvo atlikta dviejų skirtingų neuroninių tinklų tipų – autoasociatyvinio neuroninio tinklo ir tiesioginės sklaidos neuroninio tinklo – analizė, siekiant nustatyti biotechnologinio proceso būsenos kintamąjį.

Autoasociatyvinio neuroninio tinklo tyrimai parodė, kad geriausius rezultatus pasiekė 2:2:2 tinklo konfigūracija. Šio modelio vidutinė absoliutinių nuokrypių suma buvo 4485,463 g, vidutinis absoliutinis nuokrypis buvo 10,86 g, o absoliutinė procentinė paklaida buvo 9,05 %. Šie rezultatai buvo geriausi iš visų tirtų autoasociatyvinio neuroninio tinklo struktūrų, todėl galima teigti, kad nagrinėjamam uždaviniui tinkamiausia buvo paprasta 2:2:2 konfigūracija. Didesnio neuronų

skaičiaus naudojimas daugeliu atvejų paklaidų rodiklių nesumažino, o kai kuriais atvejais juos padidino. Tai rodo, kad nagrinėjamam uždaviniui nėra būtinos sudėtingos gilos architektūros.

Tiesioginės sklaidos neuroninio tinklo analizė parodė, kad šis modelis pasiekė mažesnes paklaidų rodiklių reikšmes nei autoasociatyvinis neuroninis tinklas. Geriausi rezultatai buvo gauti naudojant 3:2:1 tinklo konfigūraciją, kurios vidutinė absoliutinių nuokrypių suma siekė 3097,047 g, vidutinis absoliutinis nuokrypis buvo 7,49 g, o absoliutinė procentinė paklaida – 6,26 %. Šie rezultatai rodo, kad tiesioginės sklaidos neuroninis tinklas tiksliau atkūrė nagrinėjamo proceso kintamąjį pagal pasirinktus vertinimo kriterijus.

Palyginus abu modelius galima pastebėti, kad tiesioginės sklaidos neuroninis tinklas pasiekė mažesnes paklaidų rodiklių reikšmes nei autoasociatyvinis neuroninis tinklas. Geriausio tiesioginės sklaidos tinklo absoliutinė procentinė paklaida buvo 6,26 %, o geriausio autoasociatyvinio tinklo – 9,05 %. Atitinkamai vidutinė absoliutinių nuokrypių suma sumažėjo nuo 4485,463 g iki 3097,047 g. Tai rodo, kad tiesioginės sklaidos neuroninis tinklas buvo efektyvesnis konkrečiau kintamojo prognozavimo užduočiai, kadangi jis tiesiogiai optimizuojamas vieno išėjimo kintamojo tikslumui. Tuo tarpu autoasociatyvinis neuroninis tinklas šiame darbe sprendžia netiesioginį kintamojo atkūrimo uždavinį: kiekviename validacijos taške biomasės reikšmė randama optimizuojant įėjimo vektorių taip, kad būtų minimizuojama bendra tinklo įėjimo ir išėjimo rekonstrukcijos paklaida. Dėl šios priežasties šis metodas nėra tiesiogiai optimizuojamas vien tik biomasės prognozavimo paklaidai mažinti, todėl jo paklaidų rodikliai šiame tyrime buvo didesni nei tiesioginės sklaidos neuroninio tinklo.

## Išvados

1. Atlikta literatūros analizė parodė, kad biotechnologinių procesų būsenos kintamųjų nustatymui vis dažniau taikomi duomenimis grįsti metodai, o dirbtiniai neuroniniai tinklai yra tinkami sudėtingų netiesinių priklausomybių tarp proceso kintamųjų modeliavimui.
2. Paruošus eksperimentinius duomenis neuroninių tinklų apmokymui, buvo nustatyta, kad netipinių duomenų atmetimas ir duomenų normalizavimas yra būtini siekiant sumažinti klaidingų ar išsiskiriančių duomenų įtaką modelių mokymui ir rezultatų patikimumui.
3. Autoasociatyvinio neuroninio tinklo tyrimo metu nustatyta, kad biomasės kintamajam atkurti tinkamiausias buvo 4:2:2:2:4 struktūros neuroninis tinklas. Šios struktūros neuroninio tinklo vidutinė absoliutinių nuokrypių suma buvo mažiausia ir siekė 4485,5 g, vidutinis absoliutinis nuokrypis buvo 10,86 g, o absoliutinė procentinė paklaida – 9,05 %. Todėl ši struktūra buvo įvertinta kaip tiksliausia iš tirtų autoasociatyvinio neuroninio tinklo struktūrų.
4. Nustatyta, kad autoasociatyvinio neuroninio tinklo tikslumas priklauso nuo parinktos struktūros. Didinant kodavimo, butelio kaklelio ir dekodavimo sluoksnių neuronų skaičių, modelio paklaidų rodikliai daugeliu atvejų nemažėjo, o kai kuriais atvejais didėjo. Todėl galima teigti, kad nagrinėjamam uždaviniui sudėtingesnė autoasociatyvinio neuroninio tinklo konfigūracija nebuvo tikslinga.
5. Tiesioginės sklaidos neuroninio tinklo tyrimo metu geriausi rezultatai buvo gauti taikant 3:2:1 struktūros neuroninį tinklą. Šios struktūros tinklo vidutinė absoliutinių nuokrypių suma buvo 3097 g, vidutinis absoliutinis nuokrypis buvo 7,49 g, o absoliutinė procentinė paklaida buvo 6,26 %. Šie rodikliai buvo mažiausi iš visų tirtų tiesioginės sklaidos neuroninio tinklo struktūrų, todėl 3:2:1 struktūra laikytina tinkamiausia nagrinėjamam uždaviniui.
6. Palyginus autoasociatyvinį ir tiesioginės sklaidos neuroninius tinklus nustatyta, kad tiesioginės sklaidos neuroninis tinklas buvo tinkamesnis biomasės prognozavimo užduočiai. Geriausio tiesioginės sklaidos tinklo absoliutinė procentinė paklaida siekė 6,26 %, o geriausio autoasociatyvinio tinklo – 9,05 %. Tai rodo, kad tiesioginės sklaidos neuroninis tinklas pagal pasirinktus vertinimo rodiklius tiksliau prognozavo biomasės kintamąjį.

## Literatūros sąrašas

1. BOUAZZAOUI, A. and ABDELLATIF, A.A. Vaccine Delivery Systems and Administration Routes: Advanced Biotechnological Techniques to Improve the Immunization Efficacy. *Vaccine: X*, 2024. pp. 100500 DOI <https://doi.org/10.1016/j.jvacx.2024.100500>.
2. NOLL, P. and HENKEL, M. History and Evolution of Modeling in Biotechnology: Modeling & Simulation, Application and Hardware Performance. *Computational and Structural Biotechnology JOURNAL*, 2020, vol. 18. pp. 3309–3323 DOI <https://doi.org/10.1016/j.csbj.2020.10.018>.
3. LYUBENOVA, V., KOSTOV, G. and DENKOVA-KOSTOVA, R. Model-Based Monitoring of Biotechnological Processes—a Review. *Processes*, 2021, vol. 9, no. 6. pp. 908 DOI <https://doi.org/10.3390/pr9060908>.
4. MICHAEL, J. and MODELL, H. Validating the Core Concept of “Mass Balance”. *Advances in Physiology Education*, 2021, vol. 45, no. 2. pp. 276–280 DOI <https://doi.org/10.1152/advan.00235.2020>.
5. PESSOA, A., VITOLO, M. and LONG, P.F. *Pharmaceutical Biotechnology: A Focus on Industrial Application*. CRC Press, 2021 DOI <https://doi.org/10.1201/9781003178361>.
6. SAKAS, G., et al. Dynamic Energy and Mass Balance Model for an Industrial Alkaline Water Electrolyzer Plant Process. *International JOURNAL of Hydrogen Energy*, 2022, vol. 47, no. 7. pp. 4328–4345 DOI <https://doi.org/10.1016/j.ijhydene.2021.11.126>.
7. LARROCHE, C., SANROMAN, M.A., DU, G. and PANDEY, A. *Current Developments in Biotechnology and Bioengineering: Bioprocesses, Bioreactors and Controls*. Elsevier, 2016 DOI <https://doi.org/10.1016/C2014-0-04871-X>.
8. URREA, C. and AGRAMONTE, R. Kalman Filter: Historical Overview and Review of its use in Robotics 60 Years After its Creation. *JOURNAL of Sensors*, 2021, vol. 2021, no. 1. pp. 9674015 DOI <https://doi.org/10.1155/2021/9674015>.
9. FENG, S., et al. A Review: State Estimation Based on Hybrid Models of Kalman Filter and Neural Network. *Systems Science & Control Engineering*, 2023, vol. 11, no. 1. pp. 2173682 DOI <https://doi.org/10.1080/21642583.2023.2173682>.
10. BAILLIEUL, J. and SAMAD, T. *Encyclopedia of Systems and Control*. Springer, 2021 DOI [https://doi.org/10.1007/978-3-030-44184-5\\_300014](https://doi.org/10.1007/978-3-030-44184-5_300014).
11. ALMA, M., DAROUACH, M. and ALI, H.S. Adaptive Observer Design for a Class of Nonlinear Descriptor Systems. *Automatica*, 2023, vol. 155. pp. 111143 DOI <https://doi.org/10.1016/j.automatica.2023.111143>.
12. BESANÇON, G. Remarks on Nonlinear Adaptive Observer Design. *Systems & Control Letters*, 2000, vol. 41, no. 4. pp. 271–280 DOI [https://doi.org/10.1016/S0167-6911\(00\)00065-7](https://doi.org/10.1016/S0167-6911(00)00065-7).
13. CHAIREZ, I., ANDRIANOVA, O., POZNYAK, T. and POZNYAK, A. Adaptive Modeling of Nonnegative Environmental Systems Based on Projectional Differential Neural Networks Observer. *Neural Networks*, 2022, vol. 151. pp. 156–167 DOI <https://doi.org/10.1016/j.neunet.2022.03.028>.

14. KARIM, M.N. and RIVERA, S.L. Artificial Neural Networks in Bioprocess State Estimation. *Modern Biochemical Engineering*, 2005. pp. 1–33 DOI <https://doi.org/10.1007/BFb0000703>.
15. MONCKES, P.C.S., et al. Moisture Content Monitoring in Industrial-Scale Composting Systems using Low-Cost Sensor-Based Machine Learning Techniques. *BioresOURce Technology*, 2022, vol. 359. pp. 127456. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S0960852422007854> ISSN 0960-8524. DOI 10.1016/j.biortech.2022.127456.
16. SIPOS, A., FLOREA, A., ARSIN, M. and FIORE, U. Using Neural Networks to Obtain Indirect Information about the State Variables in an Alcoholic Fermentation Process. *Processes*, 2020, vol. 9, no. 1. pp. 74 DOI <https://doi.org/10.3390/pr9010074>.
17. SCHWEIDTMANN, A.M., ZHANG, D. and VON STOSCH, M. A Review and Perspective on Hybrid Modeling Methodologies. *Digital Chemical Engineering*, 2024, vol. 10. pp. 100136. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S2772508123000546> ISSN 2772-5081. DOI 10.1016/j.dche.2023.100136.
18. LIU, F., HEINER, M. and GILBERT, D. Hybrid Modelling of Biological Systems: Current Progress and Future Prospects. *Briefings in Bioinformatics*, 2022, vol. 23, no. 3. pp. bbac081 DOI <https://doi.org/10.1093/bib/bbac081>.
19. GALVANAUSKAS, V., SIMUTIS, R. and LÜBBERT, A. Hybrid Process Models for Process Optimisation, Monitoring and Control. *Bioprocess and Biosystems Engineering*, 2004, vol. 26. pp. 393–400 DOI <https://doi.org/10.1007/s00449-004-0385-x>.
20. POLAK, J., et al. An Innovative Hybrid Modeling Approach for Simultaneous Prediction of Cell Culture Process Dynamics and Product Quality. *Biotechnology JOURNAL*, 2024, vol. 19, no. 3. pp. 2300473 DOI <https://doi.org/10.1002/biot.202300473>.
21. MONTESINOS LÓPEZ, O.A., MONTESINOS LÓPEZ, A. and CROSSA, J. Multivariate statistical machine learning methods for genomic prediction Springer, 2022 *Fundamentals of Artificial Neural Networks and Deep Learning*, pp. 379–425 DOI [https://doi.org/10.1007/978-3-030-89010-0\\_10](https://doi.org/10.1007/978-3-030-89010-0_10).
22. WEISS, R., KARIMIJAFARBIGLOO, S., ROGGENBUCK, D. and RÖDIGER, S. Applications of Neural Networks in Biomedical Data Analysis. *Biomedicines*, 2022, vol. 10, no. 7. pp. 1469 DOI <https://doi.org/10.3390/biomedicines10071469>.
23. LEDERER, J. Activation Functions in Artificial Neural Networks: A Systematic Overview. *arXiv Preprint arXiv:2101.09957*, 2021 DOI <https://doi.org/10.48550/arXiv.2101.09957>.
24. MANSOURI, S. Application of Neural Networks in the Medical Field. *JOURNAL of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 2023, vol. 14, no. 1. pp. 69–81 DOI <https://doi.org/10.58346/jowua.2023.i1.006>.
25. CHAN, K.Y., et al. Deep Neural Networks in the Cloud: Review, Applications, Challenges and Research Directions. *Neurocomputing*, 2023, vol. 545. pp. 126327 DOI <https://doi.org/10.1016/j.neucom.2023.126327>.

26. FERNANDES, B.J., CAVALCANTI, G.D. and REN, T.I. Constructive Autoassociative Neural Network for Facial Recognition. *PloS One*, 2014, vol. 9, no. 12. pp. e115967 DOI <https://doi.org/10.1371/jOURnal.pone.0115967>.
27. KRAUSE, R. and REIMANN, S. Items Or Relations--what do Artificial Neural Networks Learn?. *arXiv Preprint arXiv:2404.12401*, 2024 DOI <https://doi.org/10.48550/arXiv.2404.12401>.
28. TAHERDOOST, H. Deep Learning and Neural Networks: Decision-Making Implications. *Symmetry*, 2023, vol. 15, no. 9. pp. 1723 DOI <https://doi.org/10.3390/sym15091723>.
29. QIAN, J., et al. A Review on Autoencoder Based Representation Learning for Fault Detection and Diagnosis in Industrial Processes. *Chemometrics and Intelligent Laboratory Systems*, 2022, vol. 231. pp. 104711 DOI <https://doi.org/10.1016/j.chemolab.2022.104711>.
30. AGIN, F., THIBAUT, J. and FAUTEUX-LEFEBVRE, C. Autoassociative Neural Network for Missing Data Imputation: A Case Study Via the Styrene Production Process. *The Canadian JOURNAL of Chemical Engineering*, 2025, vol. 103, no. 1. pp. 339–358 DOI <https://doi.org/10.1002/cjce.25377>.
31. VLADOV, S., et al. The Method of Restoring Lost Information from Sensors Based on Auto-Associative Neural Networks. *Applied System Innovation*, 2024, vol. 7, no. 3. pp. 53 DOI <https://doi.org/10.3390/asi7030053>.
32. TAYE, M.M. Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions. *Computation*, 2023, vol. 11, no. 3. pp. 52 DOI <https://doi.org/10.3390/computation11030052>.
33. ZHAO, X., et al. A Review of Convolutional Neural Networks in Computer Vision. *Artificial Intelligence Review*, 2024, vol. 57, no. 4. pp. 99 DOI <https://doi.org/10.1007/s10462-024-10721-6>.
34. KRICHEN, M. Convolutional Neural Networks: A Survey. *Computers*, 2023, vol. 12, no. 8. pp. 151 DOI <https://doi.org/10.3390/computers12080151>.
35. JIANG, R., XUE, Y. and ZOU, D. Interpretability-Aware Industrial Anomaly Detection using Autoencoders. *IEEE Access*, 2023, vol. 11. pp. 60490–60500 DOI <https://doi.org/10.1109/ACCESS.2023.3286548>.
36. LEE, W., OZGER, M., CHALLITA, U. and SUNG, K.W. Noise Learning-Based Denoising Autoencoder. *IEEE Communications Letters*, 2021, vol. 25, no. 9. pp. 2983–2987 DOI <https://doi.org/10.1109/LCOMM.2021.3091800>.
37. LI, Y., et al. The Geometry of Concepts: Sparse Autoencoder Feature Structure. *Entropy*, 2025, vol. 27, no. 4. pp. 344 DOI <https://doi.org/10.3390/e27040344>.
38. KOSKO, B. Bidirectional Associative Memories: Unsupervised Hebbian Learning to Bidirectional Backpropagation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021, vol. 51, no. 1. pp. 103–115 DOI <https://doi.org/10.1109/TSMC.2020.3043249>.
39. HUANG, C., YANG, K., YANG, Q. and ZHANG, H. Line Spectrum Extraction Based on Autoassociative Neural Networks. *JASA Express Letters*, 2021, vol. 1, no. 1 DOI <https://doi.org/10.1121/10.0003038>.

40. WAHLE, K., STANEV, E.V. and STANEVA, J. Detecting Anomalous Sea-Level States in North Sea Tide Gauge Data using an Autoassociative Neural Network. *Natural Hazards and Earth System Sciences*, 2023, vol. 23, no. 2. pp. 415–428 DOI <https://doi.org/10.5194/nhess-23-415-2023>.
41. TSIEN, R.Y. The Green Fluorescent Protein. *Annual Review of Biochemistry*, 1998, vol. 67, no. 1. pp. 509–544.
42. Norgaard M. *Nnsysd*. [viewed 2026-01-12]. Prieiga per: <https://se.mathworks.com/matlabcentral/fileexchange/87-nnsysid> .
43. GOODFELLOW, I. *Deep Learning*, 2016.
44. HASTIE, T., TIBSHIRANI, R. and FRIEDMAN, J. *The Elements of Statistical Learning*, 2009.
45. BALDI, P. *Autoencoders, Unsupervised Learning, and Deep Architectures*. JMLR Workshop and Conference Proceedings, 2012.
46. ROKONUZZAMAN, M., et al. Levenberg-Marquardt Algorithm-Based Solar PV Energy Integrated Internet of Home Energy Management System. *Applied Energy*, 2025, vol. 378. pp. 124407.
47. GRATTON, S., LAWLESS, A.S. and NICHOLS, N.K. Approximate Gauss–Newton Methods for Nonlinear Least Squares Problems. *SIAM JOURNAL on Optimization*, 2007, vol. 18, no. 1. pp. 106–132.