



**Kauno technologijos universitetas**

Elektros ir elektronikos fakultetas

**Autoasociatyviniais neuroniniais tinklais pagrįstos jutiklių  
monitoringo ir matavimų virtualizavimo sistemos tyrimas**

Baigiamasis magistro projektas

---

**Mantas Sudikas**

Projekto autorius

**Prof. dr. Vytautas Galvanauskas**

Vadovas

---

**Kaunas, 2021**



**Kauno technologijos universitetas**

Elektros ir elektronikos fakultetas

# **Autoasociatyviniais neuroniniais tinklais pagrįstos jutiklių monitoringo ir matavimų virtualizavimo sistemos tyrimas**

Baigiamasis magistro projektas

Valdymo technologijos (6211EX014)

---

**Mantas Sudikas**

Projekto autorius

**Prof. dr. Vytautas Galvanauskas**

Vadovas

**Prof. habil. dr. Rimvydas Simutis**

Recenzentas

---

**Kaunas, 2021**



**Kauno technologijos universitetas**

Elektros ir elektronikos fakultetas

Mantas Sudikas

## **Autoasociatyviniais neuroniniais tinklais pagrįstos jutiklių monitoringo ir matavimų virtualizavimo sistemos tyrimas**

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

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

Mantas Sudikas

*Patvirtinta elektroniniu būdu*

Sudikas Mantas. Autoasociatyviniais neuroniniais tinklais pagrįstos jutiklių monitoringo ir matavimų virtualizavimo sistemos tyrimas. 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: kogeneracinė jėgainė, autoasociatyviniai neuroniniai tinklai, virtualizavimas, jutikliai, monitoringas.

Kaunas, 2021. 69 p.

### **Santrauka**

Šiame baigiamajame magistro darbe tiriamas autoasociatyvinio neuroninio tinklo veikimas naudojant kogeneracinės jėgainės jutiklių matavimo duomenis. Buvo bandoma imituoti jutiklio gedimą ir stebėti kaip autoasociatyvinis neuroninis tinklas jutiklio matavimą virtualizuoja – atstato. Tam tikslui sumodeliuotas ir apmokytas autoasociatyvinio neuroninio tinklo modelis taikant Levenbergo – Marquardto apmokymo algoritimą. Tinklas buvo taikomas dviem kogeneracinės jėgainės sistemoms: termofikacinio vandens aušyklės sistemai ir kondensacinio dūmų ekonomizerio sistemai. Modeliavimas buvo atliktas naudojant MATLAB programinę įrangą.



Sudikas Mantas. Investigation of Sensor Monitoring and Measurement Virtualization System Based on 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 sciences,

Keywords: autoassociative neural networks, virtualization, sensors, monitoring.

Kaunas, 2021. 69 p.

### **Summary**

In this final master's thesis, the operation of an autoassociative neural network, using the measurement data of cogeneration power plant sensors, is investigated. Attempts were made to simulate a sensor failure and to observe how the autoassociative neural network virtualizes and restores the sensor measurement. For this purpose, a model of an autoassociative neural network was modeled and trained using the Levenberg – Marquardt training algorithm. The network was applied to two cogeneration power plant systems: a district heating water cooler system, and a flue gas condenser system. The simulation was performed using MATLAB software.

## Turinys

<b>Lentelių sąrašas .....</b>	<b>8</b>
<b>Paveikslų sąrašas .....</b>	<b>9</b>
<b>Įvadas.....</b>	<b>12</b>
<b>1. Apžvalga .....</b>	<b>13</b>
1.1. Autoasociatyvinis neuroninis tinklas ir jo taikymas jutiklių diagnostikoje.....	13
1.2. Sistemos gedimų monitoringas.....	14
1.2.1. Pamatinis skaičiavimas.....	14
1.2.2. Fizikiniai arba parametriniai modeliai.....	15
1.2.3. Duomenimis pagrįsti modeliai .....	15
1.2.4. Kokybiniai modeliai .....	15
1.3. Gedimų analizės sistema .....	16
1.4. Jutiklių diagnostika.....	17
1.4.1. Pamatinio skaičiavimo sistema.....	17
1.4.2. Duomenų analizės sritis.....	17
1.5. Gedimų diagnostika pramonėje .....	17
<b>2. Neuroniniai tinklai.....</b>	<b>20</b>
2.1. Tiesioginio sklidimo neuroniniai tinklai (TSNT).....	20
2.2. Autoasociatyviniai neuroniniai tinklai (AANT).....	22
2.3. Pagrindiniai neuroninių tinklų apmokymo algoritmai .....	22
2.3.1. Hebo mokymo taisyklė.....	22
2.3.2. Koreliacijos mokymo taisyklė.....	23
2.3.3. Minimalaus dydžio („Instar“) mokymosi taisyklė .....	23
2.3.4. „Nugalėtojas pasiima viską“ taisyklė .....	23
2.3.5. „Outstar“ mokymosi taisyklė .....	23
2.3.6. Perceptrono mokymosi taisyklė .....	24
2.3.7. Widrow – Hoffo (LMS) mokymosi taisyklė .....	24
2.3.8. Tiesinės regresijos mokymosi taisyklė.....	25
2.3.9. „Delta“ mokymosi taisyklė.....	25
2.4. Sudėtingesni neuroninių tinklų apmokymo algoritmai .....	26
2.4.1. Levenbergo – Marquardto apmokymo algoritmas (LM).....	26
2.5. Specifinės tiesioginio sklidimo neuroninių tinklų architektūros .....	27
2.5.1. Polinominiai neuroniniai tinklai .....	27
2.5.2. Funkcinių ryšių neuroniniai tinklai .....	28
2.5.3. Sarajedini ir Hecht – Nielseno neuroninis tinklas .....	29
2.5.4. Diskretaus sklidimo neuroninio tinklo, tiesioginio sklidimo versija.....	29
2.5.5. Mokymosi vektoriaus kvantavimo (LVQ) neuroninis tinklas.....	30
2.5.6. „Nugalėtojas pasiima viską“ neuroninis tinklas .....	31
2.5.7. Kaskadinės koreliacijos neuroninis tinklas .....	32
2.5.8. Radialiniai, bazinių funkcijų neuroniniai tinklai .....	33
2.6. Pasikartojantys neuroniniai tinklai .....	33
2.6.1. Hopfieldo neuroninis tinklas .....	33
2.6.2. Autoasociatyvinė atmintis .....	34
2.6.3. Dvikryptė autoasociatyvinė atmintis (DAA).....	35
2.7. Autoasociatyvinis neuroninis tinklas.....	35

2.7.1. Autoasociatyvinio neuroninio tinklo architektūra .....	35
2.7.2. Autoasociatyviniai neuroniniai tinklai ir tapatybės atvaizdavimas .....	36
2.7.3. Autoasociatyvinio neuroninio tinklo apmokymas, taikant Levenbergo – Marquardto (LMA) algoritmą.....	37
2.7.4. AANT apmokymas triukšmingose situacijose .....	38
2.7.5. AANT našumas triukšmingose situacijose.....	38
<b>3. Kietu kuru kūrenamos kogeneracinės jėgainės, termofikacinio vandens aušyklės AANT tyrimas .....</b>	<b>40</b>
3.1.1. Kietu kuru kūrenamos kogeneracinės jėgainės, termofikacinio vandens aušyklės schema ir matavimų duomenys.....	40
3.1.2. Termofikacinio vandens aušyklės veikimo principas.....	40
3.1.3. Termofikacinio vandens aušyklės schema ir matavimai .....	40
3.1.4. Termofikacinio vandens aušyklės matavimo duomenys.....	41
3.1.5. Matavimo duomenų normalizavimas .....	43
3.1.6. Apmokyto AANT rezultatai .....	44
3.1.7. „Sugedusio“ jutiklio virtualizavimo naudojant apmokytą AANT rezultatai .....	48
3.1.8. Apmokyto AANT tyrimo įvairiuose aušyklės režimuose rezultatai .....	51
3.2. Kietu kuru kūrenamos kogeneracinės jėgainės, dūmų kondensacinio ekonomizerio AANT tyrimas .....	55
3.2.1. Kogeneracinės jėgainės, dūmų kondensacinio ekonomizerio veikimo principas .....	55
3.2.2. Kondensacinio ekonomizerio matavimo duomenys .....	57
3.2.3. Matavimo duomenų normalizavimas .....	58
3.2.4. Apmokyto AANT rezultatai .....	59
3.2.5. „Sugedusio“ jutiklio virtualizavimo naudojant apmokytą AANT rezultatai .....	63
<b>Rezultatai ir išvados .....</b>	<b>66</b>
<b>Literatūros sąrašas .....</b>	<b>67</b>
<b>Priedai.....</b>	<b>70</b>
1 priedas. AANT vykdymo programa MATLAB aplinkoje .....	70
2 priedas. Termofikacinio vandens aušyklės duomenų rinkinio grafikai iš kietu kuru kūrenamos kogeneracinės jėgainės, paskirstytos valdymo sistemos (DCS).....	83
3 priedas. Kondensacinio ekonomizerio duomenų rinkinio grafikai iš kietu kuru kūrenamos kogeneracinės jėgainės, paskirstytos valdymo sistemos (DCS).....	85
4 priedas. Termofikacinio vandens aušyklės AANT apmokymo duomenų rinkinys.....	89
5 priedas. Apmokyto termofikacinio vandens aušyklės AANT neuronų svoriai.....	90
6 priedas. Kondensacinio ekonomizerio AANT apmokymo duomenų rinkinys.....	91
7 priedas. Apmokyto kondensacinio ekonomizerio AANT neuronų svoriai .....	94

## Lentelių sąrašas

<b>1 lentelė.</b> AANT apmokymo rezultatų vidutinės absoliutinės procentinės paklaidos .....	47
<b>2 lentelė.</b> AANT virtualizuotų matavimų, vidutinės absoliutinės procentinės paklaidos .....	50
<b>3 lentelė.</b> AANT rezultatų vidutinės absoliutinės procentinės paklaidos .....	52
<b>4 lentelė.</b> AANT virtualizuotų matavimų, vidutinės absoliutinės procentinės paklaidos .....	55
<b>5 lentelė.</b> AANT rezultatų vidutinės absoliutinės procentinės paklaidos .....	61
<b>6 lentelė.</b> AANT virtualizuotų matavimų, vidutinės absoliutinės procentinės paklaidos .....	65
<b>7 lentelė.</b> Termofikacinio vandens aušyklės AANT apmokymo duomenų rinkinys .....	89
<b>8 lentelė.</b> Apmokyto termofikacinio vandens aušyklės AANT neuronų svoriai .....	90
<b>9 lentelė.</b> Apmokyto termofikacinio vandens aušyklės AANT neuronų svoriai .....	90
<b>10 lentelė.</b> Apmokyto termofikacinio vandens aušyklės AANT neuronų svoriai .....	90
<b>11 lentelė.</b> Apmokyto termofikacinio vandens aušyklės AANT neuronų svoriai .....	91
<b>12 lentelė.</b> Kondensacinio ekonomizerio AANT apmokymo duomenų rinkinys .....	91
<b>13 lentelė.</b> Apmokyto kondensacinio ekonomizerio AANT neuronų svoriai .....	94
<b>14 lentelė.</b> Apmokyto kondensacinio ekonomizerio AANT neuronų svoriai .....	94
<b>15 lentelė.</b> Apmokyto kondensacinio ekonomizerio AANT neuronų svoriai .....	94
<b>16 lentelė.</b> Apmokyto kondensacinio ekonomizerio AANT neuronų svoriai .....	95

## Paveikslų sąrašas

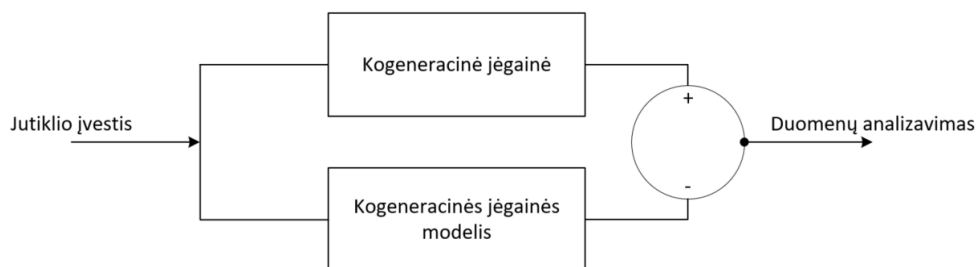
<b>1 pav.</b> Jėgainės modeliu pagrįsta duomenų analizavimo struktūrinė schema.....	12
<b>2 pav.</b> Autoasociatyvinio neuroninio tinklo architektūra [4] .....	13
<b>3 pav.</b> J. W. Hineso jutiklių diagnostikos taikant AANT pavyzdys [6].....	13
<b>4 pav.</b> Bendroji gedimo aptikimo sistemos struktūra.....	14
<b>5 pav.</b> Bendroji gedimo analizavimo sistemos struktūra .....	14
<b>6 pav.</b> Tiesioginio sklidimo neuroninis tinklas [26].....	21
<b>7 pav.</b> Tipinės aktyvavimo funkcijos [26] .....	21
<b>8 pav.</b> Įvesties erdvės atskyrimas keturių neuronų rinkiniu [26] .....	21
<b>9 pav.</b> AANT architektūra [1] .....	22
<b>10 pav.</b> Perceptrono mokymosi taisyklės iliustracija [26] .....	24
<b>11 pav.</b> Vieno sluoksnio neuroninis tinklas su netiesiniais polinomiais termiais [26] .....	28
<b>12 pav.</b> Polinominiai neuroniniai tinklai XOR problemai išspręsti: a) naudojant vienpolius signalus, b) naudojant bipolinius signalus [26] .....	28
<b>13 pav.</b> Vieno sluoksnio neuroninis tinklas su netiesiškais termiais [26].....	28
<b>14 pav.</b> Sarajedini ir Hecht – Nielseno neuroninis tinklas [26] .....	29
<b>15 pav.</b> Sarajedini ir Hecht – Nielseno neuroninio tinklo išvestis yra proporcinga Euklido atstumo kvadratui [26] .....	29
<b>16 pav.</b> Diskretaus sklidimo neuroninis tinklas [26].....	29
<b>17 pav.</b> Diskretaus sklidimo neuroninis tinklas naudojamas kaip analoginė atmintis su analoginiu adresu [26] .....	30
<b>18 pav.</b> Mokymosi vektoriaus kvantavimo (LVQ) neuroninis tinklas [26] .....	30
<b>19 pav.</b> Neuronas kaip Hammingo atstumo klasifikatorius [26].....	31
<b>20 pav.</b> Kaskadinės koreliacijos neuroninis tinklas [26].....	32
<b>21 pav.</b> Radialinis, bazinių funkcijų neuroninis tinklas [26] .....	33
<b>22 pav.</b> Autoasociatyvinė atmintis [26] .....	34
<b>23 pav.</b> Dvikryptė autoasociatyvinė atmintis (DAA) [26].....	35
<b>24 pav.</b> AANT bendroji architektūra .....	36
<b>25 pav.</b> Tapatybės atvaizdavimas.....	36
<b>26 pav.</b> Santykis tarp vidutinės kvadratinės paklaidos = 1 ir 20 iteracijų.....	39
<b>27 pav.</b> Termofikacinio vandens aušyklės P&ID schema.....	40
<b>28 pav.</b> Termofikacinio vandens aušyklės AANT modelis ir architektūra.....	41
<b>29 pav.</b> AANT apmokymo duomenų rinkinys.....	42
<b>30 pav.</b> AANT testavimo duomenų rinkinys .....	43
<b>31 pav.</b> Normalizuotas AANT apmokymo duomenų rinkinys .....	44
<b>32 pav.</b> Normalizuotas AANT testavimo duomenų rinkinys.....	44
<b>33 pav.</b> Apmokyto AANT, normalizuoti apmokymo duomenys tinklo įvestyje („>“) ir išvestyje („<“) .....	45
<b>34 pav.</b> AANT apmokymo rezultatai fizikiniais dydžiais.....	46
<b>35 pav.</b> AANT apmokymo rezultatų santykinės procentinės paklaidos .....	46
<b>36 pav.</b> AANT apmokymo duomenų savybės („*“ – kiekvieno kintamojo apmokymo rezultatai, „o“ – kiekvieno kintamojo testavimo rezultatai) .....	47
<b>37 pav.</b> AANT fazinės trajektorijos požymių sluoksnyje („*“ – apmokymas, „o“ – testavimas) ....	47
<b>38 pav.</b> „Sugedusio“ jutiklio Nr. 1, 10NDG20CQ901XQ01, MW – aušyklės momentinė šiluminė galia, vertės atkūrimas – virtualizavimas .....	48

<b>39 pav.</b> „Sugedusio“ jutiklio Nr. 2, 10NDG20CF901XQ01, kg/s – termofikacinio vandens srautas per šilumokaitį, vertės atkūrimas – virtualizavimas .....	48
<b>40 pav.</b> „Sugedusio“ jutiklio Nr. 3, 10NDA30CT901XQ01, °C – termofikacinio vandens temperatūra prieš šilumokaitį, vertės atkūrimas – virtualizavimas .....	49
<b>41 pav.</b> „Sugedusio“ jutiklio Nr. 4, 10NDG20CT001XQ01, °C – termofikacinio vandens temperatūra už šilumokaičio, vertės atkūrimas – virtualizavimas.....	49
<b>42 pav.</b> „Sugedusio“ jutiklio Nr. 5, 10PGD20CF901XQ01, kg/s – glikolio vandens srautas per šilumokaitį, vertės atkūrimas – virtualizavimas .....	49
<b>43 pav.</b> „Sugedusio“ jutiklio Nr. 6, 10PGD50CT002XQ01, °C – glikolio vandens temperatūra už aušyklės, vertės atkūrimas – virtualizavimas .....	50
<b>44 pav.</b> „Sugedusio“ jutiklio Nr. 7, 10PGD30CT002XQ01, °C – glikolio vandens temperatūra prieš aušyklę, vertės atkūrimas – virtualizavimas .....	50
<b>45 pav.</b> Atsitiktiniai matavimo duomenys AANT bandymui .....	51
<b>46 pav.</b> AANT rezultatų santykinės procentinės paklaidos .....	52
<b>47 pav.</b> „Sugedusio“ jutiklio Nr. 1, 10NDG20CQ901XQ01, MW – aušyklės momentinė šiluminė galia, vertės atkūrimas – virtualizavimas .....	53
<b>48 pav.</b> „Sugedusio“ jutiklio Nr. 2, 10NDG20CF901XQ01, kg/s – termofikacinio vandens srautas per šilumokaitį, vertės atkūrimas – virtualizavimas .....	53
<b>49 pav.</b> „Sugedusio“ jutiklio Nr. 3, 10NDA30CT901XQ01, °C – termofikacinio vandens temperatūra prieš šilumokaitį, vertės atkūrimas – virtualizavimas .....	53
<b>50 pav.</b> „Sugedusio“ jutiklio Nr. 4, 10NDG20CT001XQ01, °C – termofikacinio vandens temperatūra už šilumokaičio, vertės atkūrimas – virtualizavimas.....	54
<b>51 pav.</b> „Sugedusio“ jutiklio Nr. 5, 10PGD20CF901XQ01, kg/s – glikolio vandens srautas per šilumokaitį, vertės atkūrimas – virtualizavimas .....	54
<b>52 pav.</b> „Sugedusio“ jutiklio Nr. 6, 10PGD50CT002XQ01, °C – glikolio vandens temperatūra už aušyklės, vertės atkūrimas – virtualizavimas .....	54
<b>53 pav.</b> „Sugedusio“ jutiklio Nr. 7, 10PGD30CT002XQ01, °C – glikolio vandens temperatūra prieš aušyklę, vertės atkūrimas – virtualizavimas .....	55
<b>54 pav.</b> Kondensacinio ekonomizerio P&ID schema.....	56
<b>55 pav.</b> Kondensacinio ekonomizerio AANT modelis ir architektūra.....	56
<b>56 pav.</b> AANT apmokymo duomenų rinkinys.....	57
<b>57 pav.</b> AANT testavimo duomenų rinkinys .....	58
<b>58 pav.</b> Normalizuoti AANT apmokymo duomenų rinkinys .....	59
<b>59 pav.</b> Normalizuoti AANT testavimo duomenų rinkinys.....	59
<b>60 pav.</b> Apmokyto AANT normalizuoti apmokymo duomenys tinklo įvestyje („>“) ir išvestyje („<“) .....	60
<b>61 pav.</b> AANT apmokymo rezultatai fizikiniais dydžiais.....	61
<b>62 pav.</b> AANT rezultatų santykinės procentinės paklaidos .....	61
<b>63 pav.</b> AANT apmokymo duomenų savybės („*“ – kiekvieno kintamojo apmokymo rezultatai, „o“ – kiekvieno kintamojo testavimo rezultatai) .....	62
<b>64 pav.</b> AANT fazinės trajektorijos požymių sluoksnyje („*“ – apmokymas, „o“ – testavimas) ....	62
<b>65 pav.</b> „Sugedusio“ jutiklio Nr. 1, 10NDA50CQ901XQ01, MW – kondensacinio ekonomizerio momentinė šiluminė galia, vertės atkūrimas – virtualizavimas.....	63
<b>66 pav.</b> „Sugedusio“ jutiklio Nr. 2, 10NDA50CF901XQ01, kg/s – termofikacinio vandens srautas per ekonomizerio šilumokaitį, vertės atkūrimas – virtualizavimas .....	63

<b>67 pav.</b> „Sugedusio“ jutiklio Nr. 3, 10NDA20CT002XQ01, °C – termofikacinio vandens temperatūra prieš ekonomazerio šilumokaitį, vertės atkūrimas – virtualizavimas.....	64
<b>68 pav.</b> „Sugedusio“ jutiklio Nr. 4, 10NDA50CT001XQ01, °C – termofikacinio vandens temperatūra už ekonomazerio šilumokaičio, vertės atkūrimas – virtualizavimas .....	64
<b>69 pav.</b> „Sugedusio“ jutiklio Nr. 5, 10HTA30CT001XQ01, °C – dūmų temperatūra prieš kondensacinį ekonomazerį, vertės atkūrimas – virtualizavimas .....	64
<b>70 pav.</b> „Sugedusio“ jutiklio Nr. 6, 10HNA30CF001ZQ01, Nm <sup>3</sup> /h – dūmų srautas per kondensacinį ekonomazerį, vertės atkūrimas – virtualizavimas .....	65
<b>71 pav.</b> „Sugedusio“ jutiklio Nr. 7, 10HNA30CT001XQ02, °C – dūmų temperatūra kamine, vertės atkūrimas – virtualizavimas.....	65
<b>72 pav.</b> Termofikacinio vandens aušyklės duomenų rinkinys (2000 taškų).....	83
<b>73 pav.</b> Termofikacinio vandens aušyklės duomenų rinkinys (70 taškų).....	84
<b>74 pav.</b> Kondensacinio ekonomazerio duomenų rinkinys (2000 taškų, rinkinys Nr. 1).....	85
<b>75 pav.</b> Kondensacinio ekonomazerio duomenų rinkinys (2000 taškų, rinkinys Nr. 2).....	86
<b>76 pav.</b> Kondensacinio ekonomazerio duomenų rinkinys (70 taškų, rinkinys Nr. 1).....	87
<b>77 pav.</b> Kondensacinio ekonomazerio duomenų rinkinys (70 taškų).....	88

## Įvadas

Sugedus jutikliams, valdymo sistemos tampa nepatikimos. Net ir naudojant pačius pažangiausius prietaisus ir valdymo algoritmus, sugedusio jutiklio duomenimis pagrįstas kontrolės sprendimas gali sukelti valdymo sistemos gedimą ir sustabdyti jėgainę. Vienas iš žinomiausių gedimų nustatymo būdų yra jėgainės modelių pagrįstas metodas, kai skaičiavimo modelis sukurtas numatyti tikrąją sistemos išvestį gaunant tą patį įėjimą. 1 paveiksle parodyta bendra jėgainės modelių pagrįsta duomenų analizavimo sistema.



1 pav. Jėgainės modelių pagrįsta duomenų analizavimo struktūrinė schema

Nepaisant didelio populiarumo naudojant jėgainės modelius jutiklių gedimų nustatymo srityje, šis metodas nėra tinkamas jutiklių diagnostikai. Taip yra todėl, kad jėgainės modelių pagrįstas metodas remiasi tik teisingais įvesties duomenimis, kurie gali būti matuojami naudojant jutiklius. Tokiu būdu daroma prielaida, kad įvestis į jėgainės sistemą ir įvestis į jėgainės modelį yra teisinga – be gedimų. Kai atsiranda didelis skirtumas tarp realios sistemos ir modelio išvesties, daroma prielaida, kad realioje sistemoje yra gedimas. Analizuojant jutiklių diagnostikos temą, daugiausia dėmesio skiriama neveikiantiems jutikliams.

Autoasociatyviniai neuroniniai tinklai (AANT) yra alternatyvus jutiklių diagnostikos sprendimas. M. A. Kramerio [1] ir [2] sukurta AANT koncepcija gali užfiksuoti jėgainės procesų kintamųjų tam tikrą ryšį tarpusavyje. Koreliacijos gautos tarp realių jėgainės kintamųjų yra išsaugojamos autoasociatyvinio neuroninio tinklo jungčių svoriuose.

Autoasociatyvinio neuroninio tinklo taikymas jėgainės jutiklių diagnostikai yra šio tyrimo pagrindinis objektas. Darbe bus siekiama išsiaiškinti bendrus bruožus, kuriuos būtų galima pritaikyti skirtingose jėgainės sistemose. Kogeneracinėje jėgainėje esančio termofikacinio vandens aušintuvo matavimo duomenys bus naudojami AANT veikimui išbandyti.

Darbo tikslas:

Ištirti autoasociatyviniais neuroniniais tinklais pagrįstą jutiklių monitoringo ir matavimų virtualizavimo sistemą.

Darbo uždaviniai:

1. identifikuoti jėgainėje esančias sistemas, kuriose gali būti naudojamas AANT;
2. atrinkti jėgainės procesų matavimo duomenis skirtus apmokyti AANT;
3. parinkti geriausią AANT architektūrą;
4. apmokyti AANT taikant pasirinktą apmokymo algoritmą;
5. parašyti programą ir sumodeliuoti AANT modelį MATLAB sistemoje;
6. ištirti AANT modelį realiais kogeneracinės jėgainės matavimo prietaisų duomenimis.

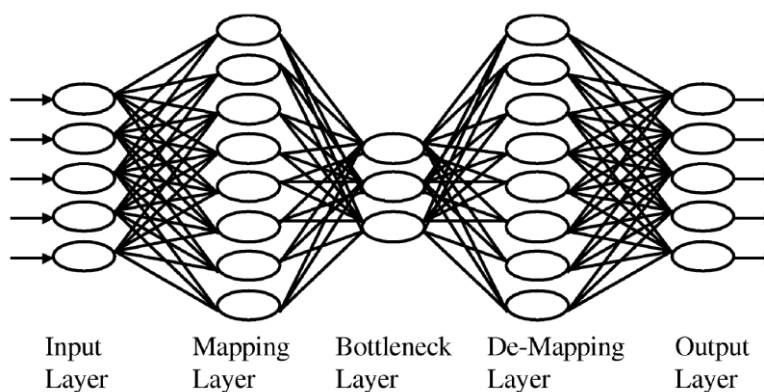


## 1. Apžvalga

### 1.1. Autoasociatyvinis neuroninis tinklas ir jo taikymas jutiklių diagnostikoje

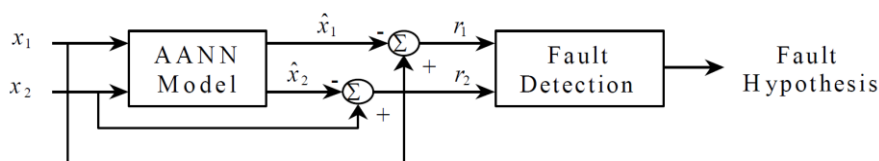
Autoasociatyvinis neuroninis tinklas yra tinklas, kuriame išėjimai yra mokomi imituoti įvestis per tinkamą dinaminį diapazoną. Jėgainės kintamieji, kurie turi tam tikrą tarpusavio ryšį – yra tinklo įėjimai. Tinklo apmokymo metu kintamųjų tarpusavio ryšiai yra įterpiami į neuroninio tinklo, ryšių svorius. Autoasociatyvinis neuroninis tinklas susideda iš įvesties sluoksnio, daugybės paslėptų sluoksnių ir išvesties sluoksnio. Teoriškai pakanka, kad AANT būtų trys paslėpti sluoksniai [1], tačiau praktikoje, norint pagerinti tinklo našumą, gali būti naudojama daugiau paslėptų sluoksnių [3]. Trijų paslėptų sluoksnių AANT architektūra parodyta 2 paveiksle. Išvesties sluoksnis sukuria transformuotą įvesties versiją ir yra lygus įvesties dydžiui.

Jutiklių diagnostikoje visi duomenys, išmatuoti realioje sistemoje (įvesties kintamųjų ir išvesties kintamųjų deriniai) yra įvestys į AANT. Autoasociatyvinis neuroninis tinklas yra mokomas taip, kad jo rezultatai kuo tiksliau, bent jau dėl kvadratinės paklaidos, atitiktų mokymo rinkinį. Kai duomenys, neturintys klaidų (be sugedusių jutiklių), perduodami į apmokytą AANT, idealiu atveju skirtumas tarp AANT įvesties ir išvesties turi būti nulis. Jei duomenys yra užteršti (sugedęs vienas ar daugiau jutiklių), skirtumas tarp įvesties ir išvesties bus nelygus nuliui.



2 pav. Autoasociatyvinio neuroninio tinklo architektūra [4]

Naudojant AANT metodą galima nustatyti ar yra jutiklio gedimas, tačiau vis tiek išlieka sugedusių jutiklių vietos nustatymo problema. J. W. Hinesas ir jo kolegės pasiūlė metodą sugedusiems jutikliams rasti naudojant AANT [4], [5] ir [6]. J. W. Hinesas [4] pažymi, kad tinklo apmokymo metu kintamųjų tarpusavio ryšiai yra įvertinti autoasociatyvinio neuroninio tinklo jungtyse. Dėl to, bet kokia konkreti tinklo išvestis praktiškai nesikeičia, kai atitinkamą įvestį iškraipo triukšmas, blogi duomenys ar duomenų trūkumas. Tai reiškia, kad skirtume tarp kiekvienos AANT įvesties ir atitinkamos išvesties yra pakankamai informacijos, kad padėtų nustatyti sugedusius jutiklius. Jei skirtumas yra lygus nuliui, atitinkamas jutiklis yra sveikas, kitu atveju jutiklis yra sugedęs arba veikia netinkamai (žr. 3 pav.).



3 pav. J. W. Hineso jutiklių diagnostikos taikant AANT pavyzdys [6]

Autoasociatyviniais neuroniniais tinklais pagrįstos jutiklių monitoringo ir matavimų virtualizavimo sistemos tyrimo temą galima suskirstyti į dvi pagrindines kategorijas:

1. sistemos gedimų monitoringas;
2. jutiklių gedimų monitoringas.

Sistemoje gedimų monitoringas nagrinėja sistemos viduje esančius gedimus. Tikrinami visos sistemos komponentai, ar jie tinkamai veikia. Jutiklių gedimų monitoringas nagrinėja gedimus atsirandančius iš jutiklių, matuojančių jėgainės procesus. Sistema tikrina ar jutikliai rodo teisingas vertes.

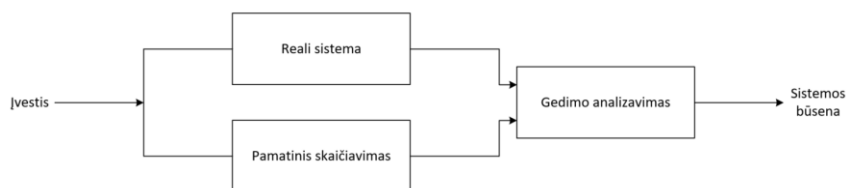
## 1.2. Sistemos gedimų monitoringas

Paprastai gedimo nustatymo procesas yra padalintas į du etapus: 1 – gedimo nustatymas ir 2 – gedimo izoliavimas. Nustatant gedimą tikrinama, ar sistemos veikimas atitinka specifikacijas ar ne. Atskyrus gedimą, jis tampa gedimo šaltiniu.

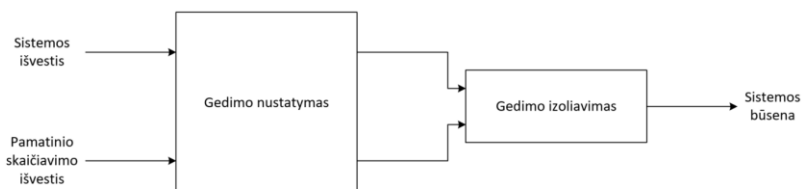
Yra keletas bendrų gedimo nustatymo būdų, tačiau gedimo išskyrimo tema yra vis dar tiriama. Taip yra dėl to, kad gedimo izoliavimo procesas priklauso nuo sistemos specifikacijų. Todėl sunku nustatyti bendrą požiūrį į gedimo izoliavimą.

Bendrąjį sistemos gedimo aptikimo mechanizmą sudaro dvi dalys: 1 – pamatinio skaičiavimo dalis ir 2 – gedimo analizavimo dalis (žr. 4 pav.). Pamatinio skaičiavimo užduotis yra paruošti įrankius įvertinti sistemą. Jis apskaičiuoja etalonines vertes, kurias reikia palyginti su sistemos išvestimis. Gedimo analizavimo dalyje atliekamos dvi funkcijos. Palyginama sistemos išvestis su apskaičiuotomis vertėmis, kad būtų nustatyta ar sistemoje nėra kokių nors gedimų. Nustačius gedimą, sistema bando jį išskirti ir izoliuoti. Todėl gedimo analizavimo sistema yra padalinta į dvi dalis (žr. 5 pav.):

1. gedimo analizavimo dalis;
2. gedimo nustatymo dalis.



4 pav. Bendroji gedimo aptikimo sistemos struktūra



5 pav. Bendroji gedimo analizavimo sistemos struktūra

### 1.2.1. Pamatinis skaičiavimas

Kaip buvo minėta anksčiau, pamatinis skaičiavimas paruošia įrankius sistemai įvertinti. Populiariausias parametrų apskaičiavimo metodas yra modeliu pagrįstas metodas. Šiuo požiūriu

modelis sukuriamas remiantis tikrosiomis sistemos specifikacijomis. Jis numato realios sistemos išvestį su apibrėžtu įėjimu. Populiariausi modelio projektavimo metodai yra šie:

1. fizikiniai arba parametriniai metodai;
2. sistemos duomenimis pagrįsti metodai;
3. kiekybiniai metodai.

### **1.2.2. Fizikiniai arba parametriniai modeliai**

Šie modeliai sukurti remiantis realios sistemos lygtimis ir parametrais. Šiuo požiūriu, pagrindinis klausimas yra išspręsti sistemos lygtis ir identifikuoti parametrus. Kaip pavyzdį galima paminėti M. Heydarzadeho ir M. Nourani [7] atliktą darbą, kuriame jie pateikė pramoninėms sistemoms skirtą, gedimų nustatymo ir gedimų izoliavimo sprendimą.

Mažiau sudėtingose sistemose, fizikinį modelį galima lengvai identifikuoti. Tyrimų metu buvo sukurtas įvertinimo algoritmas, siekiant atnaujinti modelio parametrus pagal išmatuotas jutiklių vertes. Straipsnyje [8] T. A. Najafabadi ir kolegės paaiškina, kaip elektrinių variklių jutiklių matavimo signalai gali būti naudojami parametrams identifikuoti, norint įvertinti svarbią informaciją apie variklio darbo būklę. O. A. Z. Sotomayoras ir D. Odloakas patvirtino tą patį metodą gedimams nustatyti, skirtą chemijos pramonės gamykloms [9]. Jie naudojo tikslius modelius ir įvertinimo algoritmus, kad atnaujintų modelių parametrus pagal išmatuotas procesų vertes.

Tačiau kai sistema yra per daug sudėtinga arba nėra pakankamai informacijos apie fizikines proceso savybes, fizikinių arba parametrinių modelių metodas netinkamas. Taip pat šis metodas negali palaikyti gedimų aptikimo mechanizmo, nes tai daugiausia priklauso nuo sistemos specifikacijų, kurios keičiasi kiekvienu atveju skirtingai.

### **1.2.3. Duomenimis pagrįsti modeliai**

Tai yra bendras sistemų modeliavimo metodas. Šiuo požiūriu modelis sukuriamas remiantis sistemos našumu. Duomenys, sukurti remiantis sistemos archyviniais įrašais, naudojami modelio parametrams suderinti. Dirbtiniai neuroniniai tinklai ir neryškių aibių logikos (angl. *fuzzy logic*) sistemos yra gerai žinomi šio metodo įrankiai. Bo Li, Mo – Yuen Chowas ir jo kolegės, atliko neuroniniais tinklais pagrįstų modelių, skirtų gedimams nustatyti, sistemų tyrimą [10]. Mokslininkai pasiūlė tipinę neuroninio tinklo, atgalinio sklidimo (angl. *backpropagation*) architektūrą, skirtą variklio guolių vibracijų monitoringui ir gedimams diagnozuoti. X. Z. Gao ir S. J. Ovaska [11] pateikė keletą tipišku gedimų diagnozavimo sistemų, pagrįstų dirbtiniais neuroniniais tinklais, neryškiųjų aibių logika, dirbtiniais neuroniniais tinklais – neryškių aibių logika ir genetiniais algoritmais. Mokslininkės palygino šių metodų pranašumus ir trūkumus. M. E. H. Benbouzidas ir H. Nejari [12] taikė neryškių aibių logiką, diagnozuodami asinchroninio elektros variklio statoriaus fazių būsenas.

Duomenimis pagrįstus modelius galima greitai sukurti, tačiau šie modeliai yra geri tik tada, kai naudojami geri duomenys jiems kurti. Jų tikslumas greitai krenta toje veikimo srityje, kurioje nėra modeliui apmokyti skirtų duomenų.

### **1.2.4. Kokybiniai modeliai**

Kokybiniai modeliai kuriami remiantis kokybinėmis sistemos žiniomis. Jie kokybiškai nuspėja sistemos elgseną. Jei įtraukiamos kiekybinės žinios, modelis vadinamas pusiau kokybiniu. Pusiau kokybiniai modeliai gali nustatyti kiekvieno sistemos kintamojo diapazoną. V.

Venkatasubramanias ir jo kolegos [13], [14] ir [15] dirbo su sistemų gedimų diagnostika naudodami kokybinius modelius.

Pagrindinis kokybinių modelių pranašumas yra jų gebėjimas numatyti sistemos veikimą, net kai sistemos parametrai šiek tiek keičiasi. Tais atvejais, kai neįmanoma tiksliai rasti sistemos kintamųjų, kokybinis modelis yra geresnis sprendimas. Kokybiniai modeliai negali būti naudojami kuriant bendrus metodus diagnostikos srityje. Taip yra dėl kokybinių modelių priklausomybės nuo sistemos specifikacijų, kurios kiekvienu atveju keičiasi.

### **1.3. Gedimų analizės sistema**

Gedimų analizės sistemos paskirtis yra palyginti sistemos išvestis su nuorodomis gedimams nustatyti ir išskirti gedimą. Taikant modeliu pagrįstus metodus, gedimų nustatymo sistemoje yra palyginamas realios sistemos išvesties ir modelio išvesties skirtumas su pastovia ar keičiama riba. Ribos vertė priklauso nuo sistemos specifikacijų, sistemos triukšmo lygio, modelio tikslumo ir jutiklio matavimo skiriamosios gebos. Tan Woei Wan ir Huo Hong [16] parodė, kaip diagnostikos sistemoje veikiančioje neuroninio tinklo – neryškių aibių logikos pagrindu, įvertinamas jutiklių matavimo duomenų skirtumas asinchroniniame elektriniame variklyje. Kai skirtumas peržengia nustatytą vertę, gedimų nustatymo sistemos modulyje nustatoma, kad sistemoje yra klaida. Po to pereinama prie gedimo izoliavimo modulio, kad būtų galima išskirti gedimą. Klaidingame rezultate ieškoma iš anksto apibrėžtų reikšmių. Reikšmės apibrėžiamos remiantis sistemos specifikacijomis. Jos gali būti naudojamos kuriant neryškių aibių logiką, apmokant neuroninį tinklą. M. E. H. Benbouzidas ir H. Nejari [12] parodė, kaip reikšmės gali būti išreikštos paprastu, neryškių aibių logikos metodu, kad būtų galima stebėti asinchroninio elektrinio variklio statoriaus būseną.

T. M. Rossi ir J. E. Braunas [17] pasiūlė naudoti dviejų pakopų metodą aušintuvo gedimų diagnostikai. Pirmiausia jie naudojo statistinį modelio atpažinimą, kad patikrintų, ar nėra gedimo. Jei buvo nustatomas gedimas, sistema imdavosi tolimesnių veiksmų, naudodama simptomų matricą diagnozuoti gedimą. S. Wangas ir Jin – Bo Wangas [18] pranešė apie pirmąjį temperatūros jutiklių ir srauto matuoklių gedimų radimo, diagnozavimo ir vertinimo modelį, naudojamą šaldymo įrenginyje, ir pateikė dinaminių modeliavimo rezultatų. Taisyklėmis pagrįstas jutiklis remiasi masės ir energijos tvermės dėsniais. Straipsnyje [18] pateikiamas metodas stebi tik dalį aušintuvo jutiklių. Autoriai padarė prielaidą, kad kitais jutikliais išmatuoti duomenys yra teisingi. Jei yra kokių nors jutiklių gedimų, tai jie turėtų atsispindėti temperatūros ar srauto jutikliuose. Šis požiūris negali būti bendras jutiklių diagnostikos mechanizmas. Apskritai negalima manyti, kad dalis jutiklių yra teisingi, ir remiantis šia prielaida stebėti kitus jutiklius. Prieš tai aprašyti sprendimai yra pagrįsti modeliais grįstu sprendimu. Reikia paminėti, kad modeliu pagrįstas metodas nėra tinkamas bendram jutiklio diagnostikos mechanizmui. Kaip buvo minėta anksčiau, modeliu pagrįstas metodas remiasi teisingais įvesties duomenimis (kurie gali būti matuojami naudojant jutiklius). Galima daryti prielaidą, kad įvestis į tikrąją sistemą ir įvestis į modelį yra teisingos (be gedimų). Diagnozuojant jutiklius tiek įvesties, tiek išvesties duomenys matuojami jutikliais ir nė vienas iš jų nėra patikimas. Modeliu pagrįstas metodas gali būti naudojamas stebint dalį sistemos jutiklių, bet ne visus jutiklius kartu. Taip pat yra keletas kokybinių metodų atliekant aušintuvo diagnostiką, tačiau kokybiniai metodai daugiausia priklauso nuo sistemos specifikacijų. Todėl bet kuriai sistemai turi būti sukurtas atskiras diagnostikos mechanizmas. Šiame projekte ieškoma bendro jutiklio diagnostikos mechanizmo, kuris būtų pritaikytas skirtingose sistemose.

## **1.4. Jutiklių diagnostika**

Kaip ir gedimų diagnostika taip ir jutiklių diagnostikos procesas yra padalintas į du etapus: 1 – jutiklio gedimo nustatymas ir 2 – jutiklio gedimo išskyrimas. Pirmasis žingsnis nustato, ar sistemoje nėra jutiklių problemų, o antrasis žingsnis išskiria sugedusius jutiklius.

Bendras jutiklių gedimų nustatymo mechanizmas yra padalintas į dvi dalis: 1 – pamatinio skaičiavimo sistema ir 2 – analizės sistema. Nors ši klasifikacija yra panaši į gedimų diagnostikos, vidinės struktūros skiriasi.

### **1.4.1. Pamatinio skaičiavimo sistema**

Kaip jau buvo minėta anksčiau, pamatinio skaičiavimo užduotis yra identifikuoti proceso parametrus, skirtus įvertinti sistemos veikimą. Gedimų diagnostikai buvo naudojamas modeliu grįstas metodas paduodant tą pačią įvestį realiai sistemai ir modeliui. Skirtumas tarp jų išėjimų yra geriausias rezultatas nustatyti gedimą.

Nepaisant modeliu grįsto metodo populiarumo gedimų aptikimo srityje, šis metodas netinka jutiklių diagnostikai. Modeliu pagrįstas metodas remiasi teisingais įvesties duomenimis, kurie gali būti matuojami naudojant jutiklius. Tai daro prielaidą, kad įvestis į tikrąją sistemą ir įvestis į modelį yra teisingos – be gedimų. Kai yra didelis skirtumas tarp realios sistemos ir modelio išvesties, daroma prielaida, kad tikrojoje sistemoje egzistuoja gedimas. Jutiklių diagnostikoje daugiausia dėmesio skiriama neveikiantiems jutikliams.

Jutiklių diagnostikoje tiek įvesties, tiek išvesties duomenys matuojami jutikliais, nė vienas iš jų nėra patikimas. Kiekvienas gali būti sugedęs. Negalima manyti, kad įvesties duomenys yra teisingi ir susieti visus išvesties duomenis su išvesties jutikliais. Todėl įprasti, modeliais pagrįsti metodai (modelis, kuris numato sistemos išvestį, kurios įvestyje yra tie patys duomenys), nėra tinkami diagnozuoti jutiklius.

### **1.4.2. Duomenų analizės sritis**

Duomenų analizavimo sistema palygina apskaičiuotas reikšmes pagal sistemos kintamuosius, kad nustatytų, ar nėra kokių nors jutiklių gedimų. Nustačius jutiklio gedimą, analizavimo dalis bando nustatyti sugedusius jutiklius. Analizatorius turi atskirti trikdžius, atsirandančius dėl sistemos triukšmo, nuo tų, kuriuos sukelia jutiklio gedimas. Šios užduotys skiriasi nuo gedimų diagnostikos. Diagnozuojant gedimus analizatorius turėtų surasti sugedusias dalis pagal vertes, apibrėžtas remiantis sistemos specifikacijomis, o diagnozuojant jutiklius analizatorius nustato sugedusius jutiklius, neįtraukdamas sistemos specifikacijų. Taikant jutiklių diagnostiką, galima sugalvoti bendrą metodą, nes jutiklio gedimo aptikimo mechanizmas nebūtinai gali priklausyti nuo sistemos specifikacijų.

## **1.5. Gedimų diagnostika pramonėje**

Gedimų nustatymo sprendimai yra naudojami įvairiuose pramonės sektoriuose. Vienas iš panaudojimo būdų aprašytas M. Elnouro ir jo kolegų straipsnyje [19]. Straipsnyje pateikiamas duomenimis valdomas jutiklių gedimų diagnostikos algoritmas, skirtas daugelio zonų šildymo, vėdinimo ir oro kondicionavimo (ŠVOK) sistemoms, pagrįstoms autoasociatyvinio neuroninio tinklo modeliui. Siūlomas metodas gali būti naudojamas tiek vieno, tiek kelių jutiklių gedimams diagnozuoti, lyginant tinklo įvestį su išėjimu, o tada nustatant ir išskiriant gedimą pagal sugeneruotus skirtumus.

Siūlomo metodo įgyvendinimas ir vertinimo rezultatai pateikiami ir išsamiai parodomi paprastos 2 zonų ŠVOK sistemos darbe, naudojant sistemų modeliavimo programą TRNSYS. AANN metodu pagrįstas metodas yra lyginamas su principiniu komponentų analizės (PKA) metodu, o rezultatai rodo reikšmingą diagnozavimo tikslumo pagerėjimą.

A. Gokcen ir B. Barutcu straipsnyje [20] aprašomas neuroninių tinklų panaudojimas sugedusiems jutikliams diagnozuoti vėjo jėgainių turbinose. Straipsnyje siūlomas jutiklių matavimų patvirtinimo ir gedimų vėjo turbinose aptikimo metodas. Jutiklių matavimų pagrįstumo užtikrinimas yra svarbi bendros būklės stebėsenos dalis, nes jutiklio gedimai lemia neteisingus sistemos stebėjimo rezultatus. Nors staigių jutiklių gedimų nustatymas yra gana paprastas, kalibravimo dreifus aptikti yra sunkiau. Todėl matavimo patvirtinimo tikslais buvo sukurta jutiklio kalibravimo dreifų aptikimo ir išskyrimo technika. Tam tikslui buvo naudojamos vėjo turbinos, jutiklių matavimai iš valdymo ir duomenų surinkimo sistemos. Šiam tikslui buvo pasirinkti dirbtiniai neuroniniai tinklai, atsižvelgiant į jų didelį našumą netiesinėse sistemose.

Jutiklių veikimo būsenos stebėjimas yra svarbus užtikrinti patikimą ir saugų chemijos pramonės gamyklų ir atominių elektrinių veikimą. Autoasociatyvinio neuroninio tinklo (AANT) pagrindu sukurti empiriniai jutiklių modeliai buvo plačiai aprašyti stebint jutiklius. Tačiau dėl blogai pateiktų duomenimis pagrįstų modelių, patikimumas gali būti prastas. Siekiant išspręsti šias problemas, A. Shaheryaro straipsnyje [21] siūlomos kelios reguliavimo euristicos, tokios kaip treniruotės su nervingumu, ryšių svorio mažėjimas ir kryžminis patvirtinimas. Be šių reguliavimo euristicų, tradiciniai klaidų gradientu pagrįsti mokymo algoritmai, skirti daugiasluoksniams AANT modeliams, gali būti įstrigę vietiniame optimalume. Norint išspręsti blogą reguliavimą ir tinklo apmokymo klausimus, straipsnyje siūlomas išmontuoto autoasociatyvinio jutiklio modelis (angl. *denoised autoassociative sensor model, DAASM*), pagrįstas gilaus mokymosi sistema. Siūlomą modelį sudaro daugybė paslėptų sluoksnių, kurie yra iš anksto paruošiami, naudojant automatinio kodavimo architektūrą. Siekiant pagerinti patikimumą, neprižiūrimo išankstinio apmokymo etape atliekami iškrentantys euristiniai ir duomenų sugadinimo procesai. Siūlomas jutiklio modelis yra apmokytas ir išbandytas naudojant atominės elektrinės jutiklių duomenis. Veiklos vertinimui ir palyginimui su Kramerio [1] išsamiai aprašytu penkių sluoksnių AANT modeliu, naudojančiu tikslumo, savaiminio jautrumo, sklidimo ir nuoseklaus tikimybės santykio testo, gedimų aptikimo metriką.

Autoasociatyvinių neuroninių tinklų panaudojimas elektros energiją gaminančioje jėgainėje taip pat aprašomas L. Fromaigeato ir G. Nicchiotti tyrime [22]. 300 MW galingumo garo turbina, gaminanti elektrą ir kūrenama anglimi, yra aprūpinta automatine apsaugos sistema, jėgainės būklės stebėjimo ir analizės bei automatine diagnostikos programine įranga. Apsaugos sistemos paskirtis yra nustatyti aukštas turbinos vibracijas. Duomenys vėliau apdorojami naudojant autoasociatyvinius neuroninius tinklus (AANT), kurie yra apmokyti naudojant sveikus jėgainės matavimo duomenis. AANT metodikos plačiai naudojamos anomalijas nustatyti. Tyrimo rezultatai rodo, kad tokiu požiūriu būtų galima iš anksto nustatyti gedimo įvykį, palyginti su automatine diagnostikos sistema, paremta taisyklėmis, parodant požiūrio pagrįstumą šiame kontekste. Straipsnyje pateikiami įvairūs su vibracija susiję aspektai: apsauga, būklės stebėjimas, analizė, automatinė diagnostika naudojant taisykles ir neuroninius tinklus bei aptariami jų rezultatai.

T. A. Lemma ir kolegų darbe [23] pristatoma gedimų aptikimo ir diagnostikos sistema, naudojanti diskrečios bangos transformacijos ir autoasociatyvinio neuroninio tinklo derinį. Neuroninis tinklas apmokomas Levenbergo – Marquardto apmokymo algoritmu. Darbas yra unikalus dėl to, kad jis

skirtas signalams stebėti dujų turbinoje, elektros generatoriaus ritėse, tepimo sistemoje ir vibracijų jutikliuose. Autoriai norėdami patikrinti sukurtos sistemos pagrįstumą atliko tyrimus simuliuodami gedimus turbinos sistemoje. Rezultatai parodė, kad siūlomas metodas gali nustatyti 10% šališkumą, kai vidutinis tikrasis aptikimas yra didesnis nei 95%, o diagnozės rodikliai yra nuo 96 iki 100%. Kadangi modelis buvo sukurtas ir išbandytas remiantis tikrais duomenimis, jis gali būti laikomas kompetentingu praktiniam naudojimui.

Šiuolaikinės jėgainės yra labai priklausomos nuo patikimų ir tikslių jutiklių rodmenų stebėjimo ir valdymo, todėl jutikliai tampa svarbia bet kurios elektrinės dalimi. Sugedę jutikliai gali priversti neoptimaliai dirbti jėgainę ar jos komponentus, dėl ko gali visiškai sustabdyti arba blogiausiu atveju sugadinti jėgainės komponentus. Atsižvelgiant į jutiklių svarbą, juos reikia reguliariai kalibruoti ir prižiūrėti, o tai užima daug laiko, todėl tai brangus procesas. T. Palmės, M. Fasto ir M. Therno straipsnyje [24] pateikiamas jutiklio tikslumo įvertinimo metodas, kuriuo siekiama sumažinti kalibravimo poreikį ir tuo pačiu išvengti jėgainės sustojimų dėl jutiklių gedimų. Siūlomas metodas pagrįstas dirbtinių neuroninių tinklų mokymu kaip klasifikatoriais, kad būtų galima atpažinti jutiklių dreifus. Metodas vertinamas, taikant jį dviejų tipų dujų turbinoms. Vienos ir dviejų ašių agregatams. Rezultatai rodo, kad metodas gali iš anksto nustatyti abiejų tipų mašinų jutiklių dreifus ir tiksliai virtualizuoti jutiklių matavimus. Išvados rodo, kad neuroninių tinklų naudojimas patvirtinti jutiklius galėtų prisidėti prie ekonomiškės priežiūros, taip pat padidinti elektrinių prieinamumą ir patikimumą.

Orlaivio reaktyvinis variklis, kuris yra sudėtinga sistema, kuri naudoja platų jutiklių tinklą, kad galėtų stebėti jo veikimą valdymo ir variklio būsenos valdymo tikslais. Abiem atvejais reikalingas tikslus visų jutiklių veikimas, tačiau dėl sunkių darbo sąlygų jutiklių veikimas ir tikslumas gali būti sugadintas. Svarbu anksti nustatyti matavimo dreifą arba jutiklių gedimą, nes tai gali padėti išvengti klaidingų pavojaus signalų orlaivio valdymo sistemoje. Ne mažiau svarbu numatyti matavimą, kai jutiklis sugenda ir jį virtualizuoti, kol bus pakeistas tikrasis jutiklis. AANT pagrindu sukurtas jutiklių patvirtinimo modulis yra analitiškai perteklinis jutiklių tinklas, teikiantis nuolatinę jutiklio būsenos informaciją ir įvertinantis matavimo vertę vietoje sugedusių matavimų, atliekant duomenų tikrinimą. Variklių stebėjimui naudojamų jutiklių skaičius yra didelis ir nėra perspektyvu stebėti visus jutiklius naudojant vieną AANT. Taigi B. Shaho ir kolegų straipsnyje [25] naudojamas naujas požiūris į jutiklių patvirtinimą ir įvertinimą, kai jutikliai yra sugrupuoti į mažesnes grupes, atsižvelgiant į jų vietą ir fizinius ryšius. Skirstant tinklą į mažesnes grupes, gaunama dviguba nauda. Pirma, tai sumažina sudėtingumą, atsirandantį dėl didesnio matmens, antra, tai užtikrina daugkartinį kiekvieno jutiklio patvirtinimą per įvairius tinklus. Tinklas mokomas naudojant duomenis, gautus iš turboreaktyvinio variklio modelio. Pateiktas metodas yra patvirtintas atliekant supaprastintą eksperimentą, o rezultatai rodo greitą gedimo identifikavimą ir pakankamą tikslumą jutiklio vertės numatymui.

## 2. Neuroniniai tinklai

Toliau pateiktimi įvairūs neuroninių tinklų apmokymo metodai, įskaitant prižiūrimus (su mokytoju) ir neprižiūrimus (be mokytojo) apmokymo algoritmus, kurie iliustruojami pavyzdžiais. Aptariama bendra mokymosi taisyklė kaip gaunamų signalų funkcija. Taip pat kitos neuroninių tinklų apmokymo taisyklės, tokios kaip Hebo apmokymo taisyklė (angl. *Hebbian learning*), perceptroninis mokymasis, mažiausių vidutinių kvadratų taisyklė, „delta“ taisyklė, „nugalėtojas pasiima viską“ (angl. „*Winner Takes All*“) taisyklė ir PKA – pagrindinio komponento analizės taisyklė, pateikiama kaip bendrosios mokymosi taisyklės išvedimas. Aprašomi specifinės neuroninių tinklų architektūros mokymosi algoritmai kaskadinės koreliacijos tinklams, Sarajedini ir Hecht – Nielsen tinklams, funkcinų ryšių tinklams, polinominiams tinklams, diskretaus sklidimo tinklams (angl. *counterpropagation*), radialiniams bazinių funkcijų tinklams (angl. *radial basis function*). Taip pat aprašomi specialūs lustinio neuroninio tinklo mokymosi algoritmai. Analizuojamos pagrindinės konvergencijos sunkumų priežastys, tokios kaip vietiniai minimumai. Labiau pažangių gradientais pagrįstų metodų, įskaitant pseudo inversijos mokymąsi, konjuguoto gradiento, Niutono ir LM – Levenbergo – Marquardto algoritmas.

Tikėtina, kad dėl besitęsiančios kompiuterinių technologijų raidos galime tikėtis, kad programinis skaičiavimas žymiai pagerins tradicinius skaičiavimo metodus. Natūrali šio spartaus augimo pasekmė yra intelektualijų sistemų atradimas. Mašinos intelektualų elgesį lemia architektūros lankstumas, gebėjimas realizuoti mašininį žmogaus patirties integravimą, dėsnius. Visi šie pavadinimai yra pagrindinės tyrimo srities, pavadintos „Dirbtinis intelektas“ arba „Programinis skaičiavimas“, sudedamosios dalys. Tai yra praktinė alternatyva sprendžiant matematiškai neišsprendžiamas ir sudėtingas problemas. Pagrindiniai objektai yra dirbtiniai neuroniniai tinklai ir neryškių aibių logikų sistemos.

Mašininio intelekto matematinė galia paprastai priskiriama naudojamai neuroninei sistemos architektūrai ir gedimų tolerancijai, atsirandančiai dėl masiškai sujungtos struktūros. Tokioms sistemoms būdingas sunkus lygiagretus apdorojimas. Deja, paskutinė funkcija yra prarasta, jei algoritmai įgyvendinami naudojant įprastus mikroprocesorius ar skaitmeninius kompiuterius.

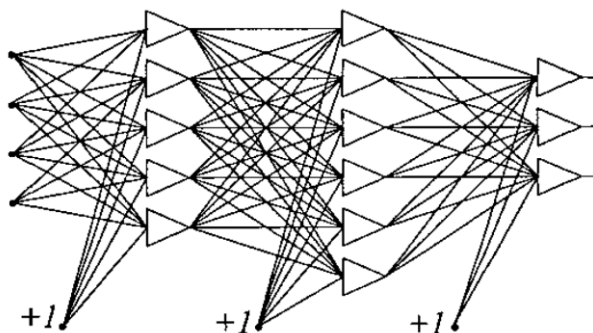
Kitas programinių skaičiavimų, kompiuterinių sistemų aspektas yra tas, kad vietoje „nulio“ ir „vieno“ skaitmeninių lygių jie naudoja neryškius / nepertraukiamus lygius ir tokiu būdu per sistemą perduodama daug daugiau informacijos. Įprasti skaitmeniniai kompiuteriai nėra tinkami tokiam signalo apdorojimui.

Trečiasis programinio skaičiavimo kompiuterinių sistemų bruožas yra jų išgyvenamumas esant gedimams. Tai reiškia, kad jie gali veikti tinkamai, jei yra iš dalies sugadinti. Priešingai, vieno bito gedimas tradiciniuose kompiuteriuose gali sukelti katastrofiškus rezultatus. Todėl labai domimasi specialios aparatinės įrangos, skirtos programiniam skaičiavimui, kūrimu.

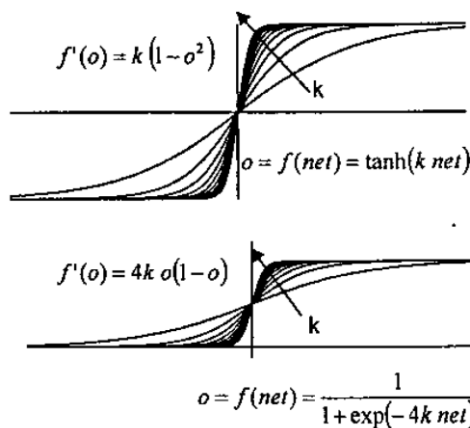
### 2.1. Tiesioginio sklidimo neuroniniai tinklai (TSNT)

Tiesioginio sklidimo neuroniniai tinklai leidžia signalo srautą tik viena kryptimi. Be to, dauguma tiesioginio sklidimo neuroninių tinklų yra organizuoti sluoksniais. Trijų sluoksnių tiesioginio sklidimo neuroninio tinklo pavyzdys parodytas 10 paveiksle. Šį tinklą sudaro įvesties mazgai, du paslėpti sluoksniai ir išvesties sluoksnis. Tipiškos aktyvavimo funkcijos parodytos 7 paveiksle. Šios aktyvavimo funkcijos leidžia daugiasluoksnius neuroninius tinklus apmokyti gradientu [26].



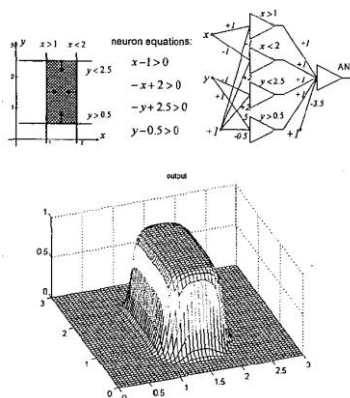


6 pav. Tiesioginio sklidimo neuroninis tinklas [26]



7 pav. Tipinės aktyvavimo funkcijos [26]

Vienas neuronas gali padalinti tik tiesiškai atskirtus modelius. Norint pasirinkti tik vieną regioną  $n$  – matmenų įvesties erdvėje, turėtų būti naudojami daugiau nei  $n + 1$  neuronų. Jei reikėtų pasirinkti daugiau įvesties klasterių, įvesties (paslėpto) sluoksnio neuronų skaičius turėtų būti atitinkamai padaugintas. Jei neuronų skaičius įvesties (paslėptame) sluoksnyje nėra ribojamas, visas klasifikavimo problemas galima išspręsti naudojant trijų sluoksnių neuroninį tinklą.



8 pav. Įvesties erdvės atskyrimas keturių neuronų rinkiniu [26]

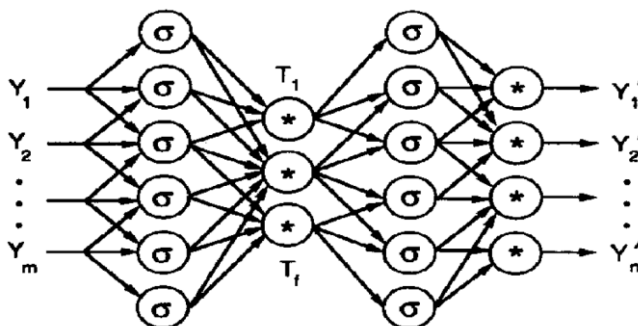
Pirmojo paslėpto sluoksnio neuronai sukuria įvesties grupių atskyrimo linijas. Antrojo paslėpto sluoksnio neuronai atlieka „IR“ (angl. „AND“) operaciją. Išvesties neuronai atlieka „ARBA“ (angl. „OR“) operaciją kiekvienai kategorijai.

Tiesioginio sklidimo neuroninis tinklas taip pat naudojamas netiesinei daugiadimensinio įvesties kintamojo transformacijai į kitą daugiadimensinį kintamąjį išvestyje. Teoriškai bet koks įvesties ir

išvesties atvaizdavimas turėtų būti įmanomas, jei neuroniniame tinkle yra pakankamai neuronų paslėptuose sluoksniuose (išvesties sluoksnio dydį nustato reikiamų išėjimų skaičius). Praktiškai tai nėra lengva užduotis. Šiuo metu nėra patenkinamo metodo apibrėžti, kiek neuronų reikia naudoti paslėptuose sluoksniuose. Dažniausiai tai nustatoma bandymo klaidų metodu. Apskritai yra žinoma, kad jei naudojama daugiau neuronų, galima atvaizduoti sudėtingesnes formas. Kita vertus, tinklai, turintys daug neuronų, praranda gebėjimą apibendrinti informaciją ir labai tikėtina, kad toks tinklas bandys susieti ir kartu su įėjimu paduodamą triukšmą.

## 2.2. Autoasociatyviniai neuroniniai tinklai (AANT)

Autoasociatyviniai neuroniniai tinklai (AANT) yra alternatyvus jutiklių diagnostikos sprendimas. AANT neskiria įvesties ir išvesties. Tokio neuroninio tinklo koncepciją, pirmą kartą sukūrė M. A. Krameris [1] 1992 m. 9 paveiksle parodyta bendra AANT architektūra. Iš esmės AANT yra tapatybės susiejimas. Jutiklių diagnostikoje tikrasis sistemos įėjimų ir išėjimų rinkinys riboja įėjimą į tinklą [1], [4] ir [5]. AANT yra apmokomas atkurti įvestį kaip išvestį. Kai duomenų įvestis į AANT yra teisinga, AANT išvestis yra tokia pati kaip įvestis. Kai įvesties duomenys yra blogi (yra jutiklio gedimas), AANT išvestis nebus tokia pati kaip jo įvestis.



9 pav. AANT architektūra [1]

## 2.3. Pagrindiniai neuroninių tinklų apmokymo algoritmai

Panašiai kaip biologiniuose neuronuose, dirbtinių neuronų svoris koreguojamas apmokymo metu. Yra sukurta įvairiausių apmokymo algoritmų ir tik keli yra tinkami daugiasluoksniams neuronų tinklams. Vieni neuronuose naudoja tik vietinius signalus, kiti reikalauja informacijos iš išėjimų, kai kurie reikalauja mokytojo, kuris žino, kokie turėtų būti išėjimai pagal pateiktus modelius, kitiems – neprižiūrimiems algoritams, tokios informacijos nereikia.

### 2.3.1. Hebo mokymo taisyklė

Hebo (1949) mokymosi taisyklė remiasi prielaida, kad jei du gretimi neuronai turi būti įjungiami ir išjungiami tuo pačiu metu, svoris, jungiantis šiuos neuronus, turėtų padidėti. Neuronams, veikiantiems priešingoje fazėje, svoris tarp jų turėtų sumažėti. Jei koreliacijos nėra, svoris turėtų likti nepakitęs. Šią prielaidą galima apibūdinti pagal formulę:

$$\Delta w_{ij} = c x_i o_j \quad (1)$$

Kur  $w_{ij}$  yra svoris nuo  $i$  - ojo iki  $j$  - ojo neurono,  $c$  yra mokymosi konstanta,  $x_i$  yra signalas  $i$  - ajame įėjime, o  $o_j$  yra išėjimo signalas. Apmokymo procesas paprastai prasideda nuo to, kad visų svorių vertės nustatomos į nulį. Ši mokymosi taisyklė gali būti naudojama tiek minkšto, tiek kieto slenksčio

neuronams. Kadangi norimi neuronų atsakai mokymosi procese nėra naudojami, tai yra apmokymo be mokytojo taisyklė. Absoliutinės svorių vertės paprastai yra proporcingos mokymosi laikui, o tai yra nepageidaujamas poveikis.

### 2.3.2. Koreliacijos mokymo taisyklė

Koreliacijos mokymo taisyklė remiasi panašiu principu kaip ir Hebo mokymosi taisyklė. Daroma prielaida, kad svoris tarp vienu metu reaguojančių neuronų turėtų būti daugiausia teigiamas, o svoris tarp priešingos reakcijos neuronų turėtų būti daugiausia neigiamas. Matematiškai tai galima parašyti, kad svoriai turėtų būti proporcingi sujungtų neuronų būsenų sandaugai. Priešingai nei Hebo taisyklei, koreliacijos taisyklė yra mokymosi su mokytoju taisyklė. Svorio pokyčiui apskaičiuoti vietoje faktinio atsako naudojama norima reakcija:

$$\Delta w_{ij} = cx_i d_j \quad (2)$$

Šis mokymo algoritmas paprastai pradedamas nuo svorio lygaus nuliui nustatymo.

### 2.3.3. Minimalaus dydžio („Instar“) mokymosi taisyklė

Jei įvesties vektoriai ir svoriai yra normalizuoti arba jie turi tik dvejetaines dvipoles reikšmes (-1 arba +1), tada *net* vertė turės didžiausią teigiamą vertę, kai svoriai turi tas pačias vertes kaip ir įvesties signalai. Todėl svorius reikėtų keisti tik tuo atveju, jei jie skiriasi nuo signalų:

$$\Delta w_i = c(x_i - w_i) \quad (3)$$

Reikia atkreipti dėmesį, kad svoriui reikalinga informacija imama tik iš įvesties signalų. Tai labai lokalus ir mokymosi be mokytojo apmokymo algoritmas.

### 2.3.4. „Nugalėtojas pasiima viską“ taisyklė

„Nugalėtojas pasiima viską“ (angl. „*Winner Takes All*“) mokymosi algoritmas yra „Instar“ algoritmo modifikacija, kai svoriai modifikuojami tik neuronui, kurio grynoji vertė yra didžiausia. Likusių neuronų svoris lieka nepakitęs. Kartais šis algoritmas modifikuojamas taip, kad tuo pačiu metu modifikuojami keli neuronai, kurių grynoji vertė yra didžiausia. Šis mokymosi be mokytojo algoritmas turi visuotinį pobūdį. Visų tinklo neuronų grynosios vertės turėtų būti lyginamos kiekviename mokymo etape. Kohoneno [26] (1982) sukurtas algoritmas dažnai naudojamas automatiniam grupavimui ir statistinėms įvesties duomenų savybėms išgauti.

### 2.3.5. „Outstar“ mokymosi taisyklė

Pagal šią mokymosi taisyklę reikalaujama, kad svoriai, prijungti prie tam tikro mazgo, būtų lygūs norimiems per tuos svorius sujungtų neuronų rezultatams:

$$\Delta w_{ij} = c(d_j - w_{ij}) \quad (4)$$

kur  $d_j$  yra norima neurono išvestis, o  $c$  yra minimali mokymosi konstanta, kuri dar labiau sumažėja mokymosi procedūros metu. Tai yra mokymosi su mokytoju procedūra, nes reikia žinoti norimus rezultatus. Tiek „Instar“, tiek „Outstar“ mokymosi taisykles sukūrė Grossberg (1974) [26].

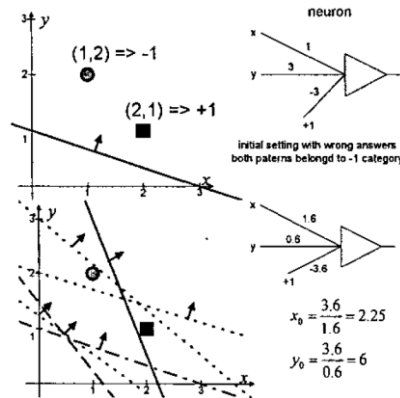
### 2.3.6. Perceptrono mokymosi taisyklė

$$\Delta w_i = c\sigma x_i \quad (5)$$

$$\sigma = d - o \quad (6)$$

$$\Delta w_i = \alpha x_i (d - \text{sign}(\text{net})) \quad (7)$$

$$\text{net} = \sum_{i=1}^n w_i x_i \quad (8)$$



10 pav. Perceptrono mokymosi taisyklės iliustracija [26]

### 2.3.7. Widrow – Hoffo (LMS) mokymosi taisyklė

Widrowas ir Hoffas [26] (1962) sukūrė su mokytoju apmokymo algoritmą, kuris leidžia apmokyti neuroną norimam atsakui. Ši taisyklė buvo išvesta, todėl  $\text{net}$  ir išvesties vertės skirtumo kvadratas yra minimizuotas:

$$\text{Error}_j = \sum_{p=1}^P (\text{net}_{jp} - d_{jp})^2 \quad (9)$$

Kur  $\text{Error}_j$  yra  $j$ -ojo neurono paklaida,  $P$  yra pritaikytų modelių skaičius,  $d_{jp}$  yra pageidaujama  $j$ -ojo neurono išvestis, kai taikomas  $p$ -asis šablonas, o  $\text{net}$  vertė pateikiama lygtimi (7). Ši taisyklė taip pat vadinama LMS (mažiausio vidutinio kvadrato) taisykle. Apskaičiavus (8) lygtį  $w_i$  atžvilgiu, galima rasti svorio pokyčio formulę:

$$\Delta w_{ij} = c x_{ij} \sum_{p=1}^P (d_{jp} - \text{net}_{jp}) \quad (10)$$

Reikia atkreipti dėmesį, kad svorio pokytis yra kiekvieno, atskiro, pritaikyto modelio, pokyčių suma. Todėl po kiekvieno atskiro modelio pritaikymo galima koreguoti svorį. Tai yra laipsniškas atnaujinimas ir kaupimas, kai svoriai keičiami pritaikius visus modelius. Laipsniškas atnaujinimas parodo sprendimą greičiau, tačiau jis jautriai reaguoja į schemų taikymo tvarką. Jei pasirenkama maža  $c$  mokymosi konstanta, tai abu metodai duoda tą patį rezultatą. Ši mokymosi taisyklė gerai veikia su visų rūšių aktyvavimo funkcijomis. Metodas bando priversti  $\text{net}$  vertę tapti lygia norimai vertei. Kartais tai nėra tai, ko mes ieškome. Dažniausiai, visiškai nesvarbu, kokia yra  $\text{net}$  vertė, tačiau svarbu,

ar *net* vertė yra teigiama ar neigiama. Pavyzdžiui, labai didelė *net* vertė su tinkamu ženklu sukels didelę klaidą ir tai gali būti pageidaujamas sprendimas.

### 2.3.8. Tiesinės regresijos mokymosi taisyklė

Ši neuroninių tinklų mokymosi taisyklė reikalauja šimtų ar tūkstančių iteracijų, kol ji sutampa su tinkamu sprendiniu. Naudojant tiesinę regresiją, tą patį rezultatą galima gauti tik vienu žingsniu. Atsižvelgiant į vieną neuroną ir naudojant vektorinį žymėjimą įvesties modelio  $X$  rinkiniui, taikomam per svorį  $w$ , *net* verčių, *net* vektorius apskaičiuojamas naudojant formulę:

$$Xw = net \quad (11)$$

arba

$$\begin{bmatrix} \sum_{p=1}^P x_{p1}x_{p1} & \sum_{p=1}^P x_{p1}x_{p2} & \cdots & \sum_{p=1}^P x_{p1}x_{pN} \\ \sum_{p=1}^P x_{p2}x_{p1} & \sum_{p=1}^P x_{p2}x_{p2} & \cdots & \sum_{p=1}^P x_{p2}x_{pN} \\ \cdots & \cdots & \cdots & \cdots \\ \sum_{p=1}^P x_{pN}x_{p1} & \sum_{p=1}^P x_{pN}x_{p2} & \cdots & \sum_{p=1}^P x_{pN}x_{pN} \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_N \end{bmatrix} = \begin{bmatrix} \sum_{p=1}^P d_p x_{p1} \\ \sum_{p=1}^P d_p x_{p2} \\ \cdots \\ \sum_{p=1}^P d_p x_{pN} \end{bmatrix} \quad (12)$$

Reiktų atkreipti dėmesį, kad modelio įvesties reikšmė visada padidinama vienetu, ir šis papildomas svoris yra atsakingas už slenkstį. Šis metodas, panašus į LMS taisyklę, naudoja linijinę aktyvavimo funkciją, todėl *net* vertės *net* turėtų būti lygios norimoms išėjimo vertėms  $d$ .

$$Xw = d \quad (13)$$

Paprastai  $p > n + 1$ , o aukščiau pateiktą lygtį galima išspręsti tik mažiausios vidutinės kvadratinės paklaidos prasme:

$$W = (X^T X)^{-1} X^T d \quad (14)$$

### 2.3.9. „Delta“ mokymosi taisyklė

LMS metodui naudojama tiesinė aktyvavimo funkcija  $net = 1$ , o gautas sprendimas kartais toli gražu nėra optimalus. Jei klaida apibrėžiama kaip:

$$Error_j = \sum_{p=1}^P (o_{jp} - d_{jp})^2 \quad (15)$$

Tada klaidos išvestis svorio atžvilgiu yra:

$$\frac{d Error_j}{d w_{ij}} = 2 \sum_{p=1}^P (o_{jp} - d_{jp}) \frac{df(net_{jp})}{d net_{jp}} x_i \quad (16)$$

Ši išraiška yra proporcinga aktyvavimo funkcijos  $f'(net)$  išvestinei.

Taikant kaupiamąjį metodą, neurono svoris  $w_{ij}$  turėtų būti keičiamas atsižvelgiant į gradiento kryptį:

$$\Delta w_{ij} = c x_i \sum_{p=1}^P (d_{jp} - o_{jp}) f_{j'p} \quad (17)$$

Kiekvienam pritaikytam modeliui apmokant palaipsniui:

$$\Delta w_{ij} = c x_i f_{j'} (d_j - o_j) \quad (18)$$

svorio pokytis turėtų būti proporcingas įvesties signalui  $x_i$ , skirtumui tarp norimų ir faktinių išėjimų  $d_{jp} - o_{jp}$ , ir aktyvavimo funkcijos  $f_{j'p}$  išvestinei. Panašiai kaip LMS taisyklėje, svoriai gali būti atnaujinami taikant didėjančią ir kaupiamą metodus. Palyginti su LMS taisykle, „delta“ taisyklė visada veda prie sprendimo, artimo optimaliam.

## 2.4. Sudėtingesni neuroninių tinklų apmokymo algoritmai

### 2.4.1. Levenbergo – Marquardto apmokymo algoritmas (LM)

Staciausio nusileidimo metodas:

$$w_{k+1} = w_k - \alpha g \quad (19)$$

Kur  $g$  yra gradiento vektorius:

$$\text{gradient } \mathbf{g} = \begin{pmatrix} \frac{\partial E}{\partial w_1} \\ \frac{\partial E}{\partial w_2} \\ \vdots \\ \frac{\partial E}{\partial w_n} \end{pmatrix} \quad (20)$$

Niutono metodas:

$$w_{k+1} = w_k - A_k^{-1} g \quad (21)$$

Kur  $A_k$  yra Hesės matrica:

$$\text{Hessian } A = \begin{pmatrix} \frac{\partial^2 E}{\partial w_1^2} & \frac{\partial^2 E}{\partial w_2 \partial w_1} & \cdots & \frac{\partial^2 E}{\partial w_n \partial w_1} \\ \frac{\partial^2 E}{\partial w_1 \partial w_2} & \frac{\partial^2 E}{\partial w_2^2} & \cdots & \frac{\partial^2 E}{\partial w_n \partial w_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial w_1 \partial w_n} & \frac{\partial^2 E}{\partial w_2 \partial w_n} & \cdots & \frac{\partial^2 E}{\partial w_n^2} \end{pmatrix} \quad (22)$$

$$J = \begin{bmatrix} \frac{\partial e_{11}}{\partial w_1} & \frac{\partial e_{11}}{\partial w_2} & \dots & \frac{\partial e_{11}}{\partial w_n} \\ \frac{\partial e_{21}}{\partial w_1} & \frac{\partial e_{21}}{\partial w_2} & \dots & \frac{\partial e_{21}}{\partial w_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial e_{M1}}{\partial w_1} & \frac{\partial e_{M1}}{\partial w_2} & \dots & \frac{\partial e_{M1}}{\partial w_N} \\ \vdots & \vdots & & \vdots \\ \frac{\partial e_{1P}}{\partial w_1} & \frac{\partial e_{1P}}{\partial w_2} & \dots & \frac{\partial e_{1P}}{\partial w_N} \\ \frac{\partial e_{2P}}{\partial w_1} & \frac{\partial e_{2P}}{\partial w_2} & \dots & \frac{\partial e_{2P}}{\partial w_N} \\ \vdots & \vdots & & \vdots \\ \frac{\partial e_{MP}}{\partial w_1} & \frac{\partial e_{MP}}{\partial w_2} & \dots & \frac{\partial e_{MP}}{\partial w_N} \end{bmatrix} \quad (23)$$

$$A = 2J^T J \text{ ir } g = 2J^T e \quad (24)$$

Gauso – Niutono metodas:

$$w_{k+1} = w_k - (J_k^T J_k)^{-1} J_k^T e \quad (25)$$

Levenbergo – Marquardto metodas:

$$w_{k+1} = w_k - (J_k^T J_k + \mu I)^{-1} J_k^T e \quad (26)$$

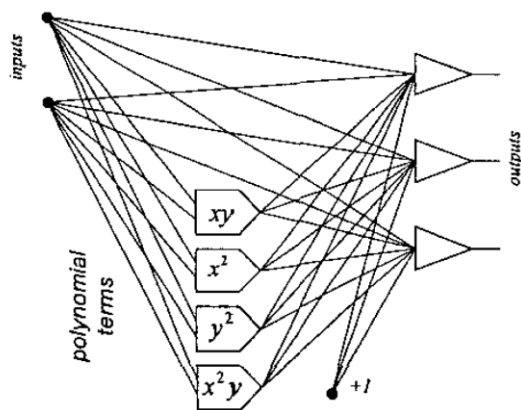
Levenbergo – Marquardto algoritmui reikia apskaičiuoti Jokūbo (angl. *Jacobian*) matricą  $J$ , kiekviename iteracijos etape ir  $J^T J$  kvadratinės matricos inversiją. Reiktų atkreipti dėmesį, kad LM algoritme  $N \times N$  matrica turi būti apversta kiekvienoje iteracijoje. Tai yra priežastis, kodėl didelių dydžių neuroniniams tinklams LM algoritmas nėra praktiškas.

## 2.5. Specifinės tiesioginio sklidimo neuroninių tinklų architektūros

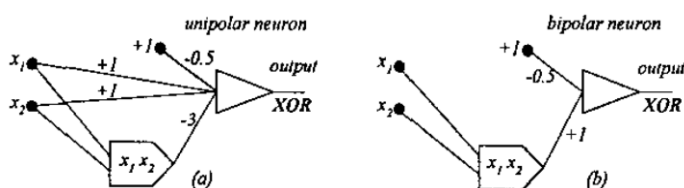
Norint rasti tinkamus sprendimus naudojant atgalinio sklidimo neuroninius tinklus, dažniausiai reikia daug bandymų su skirtingomis tinklo struktūromis ir skirtingais pradiniais atsiktiniais svoriais. Svarbu, kad apmokytas tinklas įgytų apibendrinimo savybę. Tai reiškia, kad apmokytas tinklas turėtų mokėti teisingai valdyti duomenis, kurie nebuvo naudojami apmokymams. Todėl mokymo metu dažnai kai kurie duomenys pašalinami iš apmokymo duomenų rinkinių ir tada šie duomenys naudojami validavimui. Rezultatai, naudojant atgalinio sklidimo neuroninius tinklus, dažnai priklauso nuo sėkmės. Tai paskatino mokslininkus kurti patikimesnius tinklus [26].

### 2.5.1. Polinominiai neuroniniai tinklai

Naudojant netiesinius termus su iš pradžių nustatytomis funkcijomis, padidėja faktinis įėjimų, tiekiamų į vieno sluoksnio neuroninį tinklą, skaičius. Paprasčiausiu atveju netiesiniai elementai yra aukštesnės eilės įvesties modelių daugianariai termai. Vieno sluoksnio mokymosi procedūra yra lengva ir greita. 16 paveiksle parodyta XOR problema, išspręsta naudojant funkcinių ryšių tinklus.



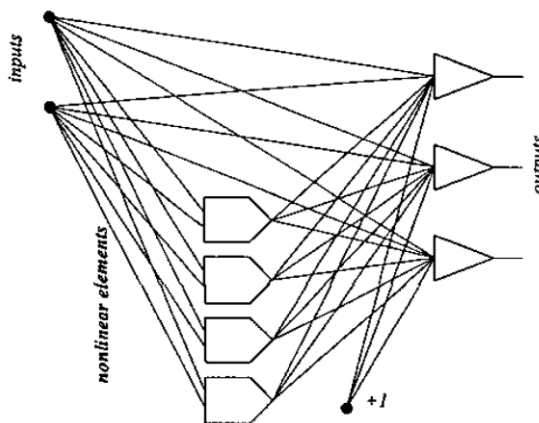
11 pav. Vieno sluoksnio neuroninis tinklas su netiesiniais polinomiais termiais [26]



12 pav. Polinominiai neuroniniai tinklai XOR problemai išspręsti: a) naudojant vienpolius signalus, b) naudojant bipolinius signalus [26]

### 2.5.2. Funkcinių ryšių neuroniniai tinklai

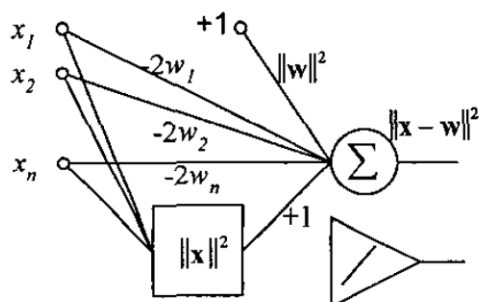
Vieno sluoksnio neuroninius tinklus palyginti lengva apmokyti, tačiau šie tinklai gali išspręsti tik tiesiškai atskirtas problemas. Vieną galimą netiesinių problemų sprendimą parengė Pao [26] (1989), naudodamas funkcinių ryšių neuroninį tinklą, parodytą 18 paveiksle. Funkcinių ryšių neuroninis tinklas gali būti vadinamas, kaip vieno sluoksnio neuroninis tinklas, kur papildomi įvesties duomenys generuojami neprisijungus (angl. *offline*), naudojant netiesines transformacijas.



13 pav. Vieno sluoksnio neuroninis tinklas su netiesiškais termiais [26]

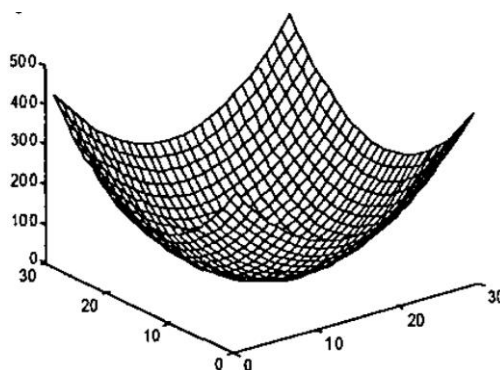


### 2.5.3. Sarajedini ir Hecht – Nienseno neuroninis tinklas



$$\|x - w\|^2 = x^T x + w^T w - 2x^T w = \|x\|^2 + \|w\|^2 - 2net$$

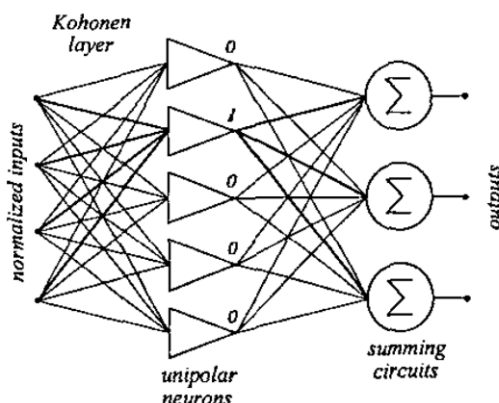
14 pav. Sarajedini ir Hecht – Nienseno neuroninis tinklas [26]



15 pav. Sarajedini ir Hecht – Nienseno neuroninio tinklo išvestis yra proporcinga Euklido atstumo kvadratui [26]

### 2.5.4. Diskretaus sklidimo neuroninio tinklo, tiesioginio sklidimo versija

Diskretaus sklidimo neuroninį tinklą iš pradžių pasiūlė Hecht – Nilsenas (1987). Šiam tinklui, kuris parodytas 21 paveiksle reikalingas paslėptų neuronų skaičius, lygus įvesčių skaičiui arba tiksliau, įvesties grupių skaičiui.



16 pav. Diskretaus sklidimo neuroninis tinklas [26]

Kai atsižvelgiama į dvejetainius įvesties duomenis, tada įvesties svoriai turi būti lygūs įvesties duomenims. Šiuo atveju:

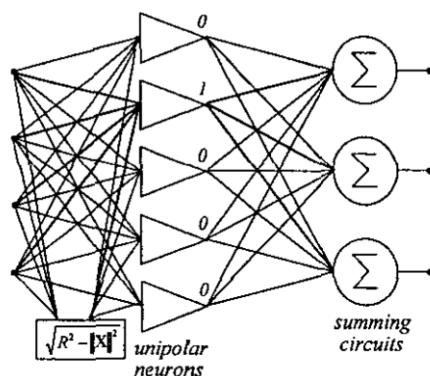
$$net = x^t w = (n - 2HD(x, w)) \quad (27)$$

kur  $n$  yra įėjimų skaičius,  $w$  yra svoriai,  $x$  yra įvesties vektorius, o  $HD(x, w)$  yra Hammingo atstumas tarp įvesties duomenų ir svorių. Kad įvesties sluoksnio neuronas reaguotų tik dėl saugomų duomenų, šio neurono ribinė vertė turėtų būti

$$w_{n+1} = -(n - 1) \quad (28)$$

Jei reikalaujama, kad neuronas taip pat turėtų reaguoti į panašius duomenis, ribinis slenkstis turėtų būti nustatytas į  $w_{n+1} = -(n - (1 + HD))$ , kur  $HD$  yra Hammingo atstumas, apibrėžiantis panašumo diapazoną. Kadangi tam tikram įvesties šablonui tik vienas neuronas pirmajame sluoksnyje gali turėti vienos vertės reikšmę, o likusieji neuronai turi nulinę vertę, išvesties sluoksnio svoriai yra lygūs reikiamai išėjimo vertei.

Tinklas su vienpolio aktyvavimo funkcijomis pirmajame sluoksnyje veikia kaip paieškos lentelė. Kai antrajame sluoksnyje naudojama tiesinė aktyvavimo funkcija (arba jos iš viso nėra), tinklas taip pat gali būti laikomas analogine atmintimi.

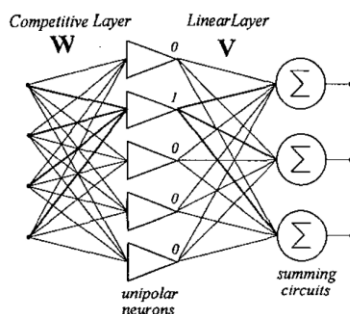


**17 pav.** Diskretaus sklaidimo neuroninis tinklas naudojamas kaip analoginė atmintis su analoginiu adresu [26]

Diskretaus sklaidimo neuroninį tinklą lengva suprojektuoti. Neuronų skaičius paslėptame sluoksnyje turėtų būti lygus duomenų (grupių) skaičiui. Svoriai įvesties sluoksnyje turėtų būti lygūs įvesties duomenims, o išvesties sluoksnio svoriai turėtų būti lygūs išvesties duomenims. Šis paprastas tinklas gali būti naudojamas greitam prototipų kūrimui. Tokiame neuroniniame tinkle paprastai yra daugiau paslėptų neuronų nei reikalaujama.

### 2.5.5. Mokymosi vektoriaus kvantavimo (LVQ) neuroninis tinklas

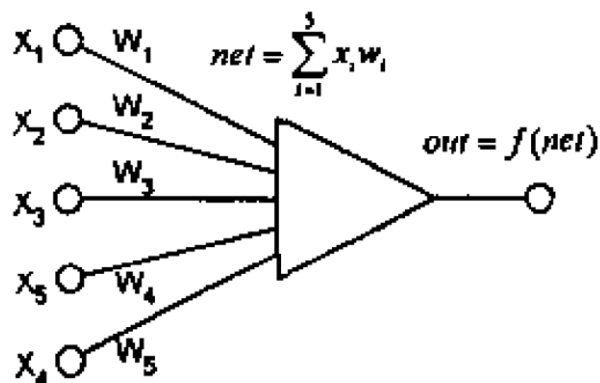
Tokio tipo neuroniniame tinkle pirmasis sluoksnis aptinka poklasius. Antrasis sluoksnis subklases sujungia į vieną klasę. Pirmasis sluoksnis apskaičiuoja Euklido atstumus tarp įvesties duomenų ir saugomų duomenų. „Laimėjęs neuronas“ yra su minimaliu atstumu.



**18 pav.** Mokymosi vektoriaus kvantavimo (LVQ) neuroninis tinklas [26]

## 2.5.6. „Nugalėtojas pasiima viską“ neuroninis tinklas

Šio tinklo (angl. „*Winner Takes All*“) architektūrą pasiūlė Kohonenas [26] (1988). Iš esmės tai yra vieno sluoksnio tinklas, naudojamas mokymo be mokytojo algoritme, statistinei įvesties duomenų savybei išgauti. Pirmajame etape visi įvesties duomenys yra normalizuojami, todėl kiekvieno įvesties vektoriaus ilgis yra vienodas ir paprastai lygus daugumai. Neuronų aktyvavimo funkcijos yra vienpolės ir tęstinės. Mokymosi procesas prasideda nuo svorio pradžios iki mažų atsitiktinių reikšmių.



19 pav. Neuronas kaip Hammingo atstumo klasifikatorius [26]

Jei įvestys yra dvejetainiai skaičiai, pavyzdžiui  $X = [1, -1, 1, -1, -1]$ , tada didžiausia  $net$  vertė

$$net = \sum_{i=1}^5 x_i w_i = XW^T \quad (29)$$

yra tada, kai svoriai yra identiški įvesties modeliui  $W = [1, -1, 1, -1, -1]$ . Šiuo atveju  $net = 5$ . Dvejetainiams svoriams ir šablonams  $net$  vertę galime rasti naudodami lygtį:

$$net = \sum_{i=1}^n x_i w_i = XW^T = n - 2HD \quad (30)$$

kur  $n$  yra įėjimų skaičius ir  $HD$  yra Hammingo atstumas tarp įvesties vektoriaus  $X$  ir svorio vektoriaus  $W$ . Šią sąvoką galima išplėsti į svorius ir duomenis su analoginėmis vertėmis, jei abu svorio vektoriaus ir įvesties duomenų vektorių ilgiai yra vienodi.

Euklido atstumas tarp svorio vektoriaus  $W$  ir įvesties vektoriaus  $X$  yra:

$$\|W - X\| = \sqrt{(w_1 - x_1)^2 + (w_2 - x_2)^2 + \dots + (w_n - x_n)^2}$$

$$\|W - X\| = \sqrt{\sum_{i=1}^n (w_i - x_i)^2} \quad (31)$$

$$\|W - X\| = \sqrt{WW^T - 2WX^T + XX^T} \quad (32)$$

Kai svorio ir įvesties vektorių ilgiai normalizuojami pagal vieno vertę:

$$\|X\| = 1 \text{ ir } \|W\| = 1 \quad (33)$$

Tada lygtys supaprastinamos iki:

$$\|W - X\| = \sqrt{2 - 2WX^T} \quad (34)$$

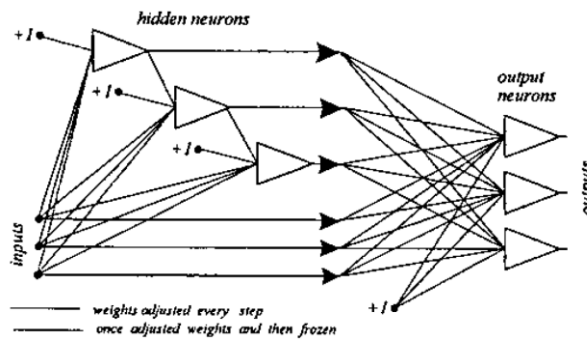
Didžiausia  $net$  vertės  $net = l$  vertė yra tada, kai  $W$  ir  $X$  yra identiški.

Kohoneno WTA tinklai turi keletą trūkumų:

1. normalizavimo metu prarandama svarbi informacija apie vektoriaus ilgį.
2. grupavimas priklauso nuo:
  - a) taikomų duomenų;
  - b) pradinių neuronų skaičiaus;
  - c) pradinių svorių.

### 2.5.7. Kaskadinės koreliacijos neuroninis tinklas

Kaskadinės koreliacijos architektūrą pasiūlė Fahlmanas ir Lebiereas [26] (1990). Tinklo kūrimo procesas prasideda vieno sluoksnio neuroninio tinklo ir prireikus pridedami paslėpti neuronai.



20 pav. Kaskadinės koreliacijos neuroninis tinklas [26]

Kiekviename apmokymo etape pridedamas naujas paslėptas neuronas ir koreguojami jo svoriai, siekiant maksimaliai padidinti koreliacijos tarp naujo paslėpto neurono išėjimo ir likusio klaidos signalo tinklo išvestyje, kurį bandoma pašalinti, dydį. Žemiau apibrėžtas,  $S$  koreliacijos parametras turi būti maksimaliai padidintas:

$$S = \sum_{o=1}^O \left| \sum_{p=1}^P (V_p - \bar{V})(E_{po} - \bar{E}_o) \right| \quad (35)$$

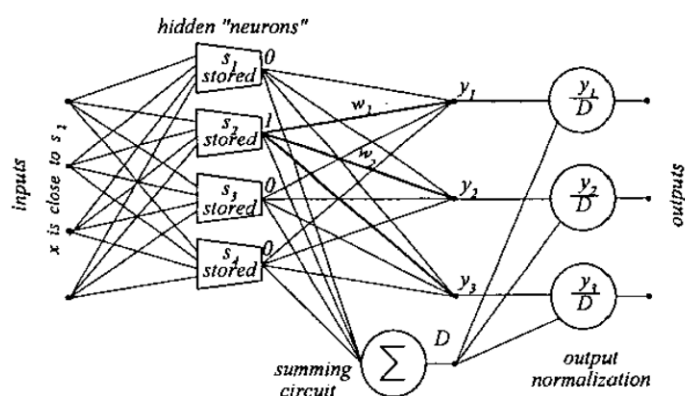
kur  $O$  yra tinklo išėjimų skaičius,  $P$  yra mokymo modelių skaičius,  $V_p$  yra išvestis į naują paslėptą neuroną ir yra tinklo išvesties klaida. Suradus gradientą  $\Delta S / \Delta w_i$ , naujo neurono svorio koregavimą galima rasti kaip:

$$\Delta w_i = \sum_{o=1}^O \sum_{p=1}^P \sigma_o (E_{po} - \bar{E}_o) f_p' x_{ip} \quad (36)$$

Išėjimo neuronai apmokomi naudojant „delta“ (atgalinio sklidimo) algoritmą. Kiekvienas paslėptas neuronas apmokomas tik vieną kartą, tada jo svoris yra užšaldomas. Tinklo mokymo procesas ir kūrimas yra baigiamas tada, kai gaunami norimi rezultatai.

### 2.5.8. Radialiniai, bazinių funkcijų neuroniniai tinklai

Radialinio, bazinių funkcijų tinklo struktūra parodyta 26 paveiksle. Šio tipo tinkle dažniausiai būna tik vienas paslėptas sluoksnius su specialiais neuronais. Kiekvienas iš šių neuronų reaguoja tik į įvesties signalus, artimus išsaugotiems duomenims.



21 pav. Radialinis, bazinių funkcijų neuroninis tinklas [26]

$I$  - ojo paslėpto neurono išėjimo signalas  $h_i$  apskaičiuojamas naudojant formulę:

$$h_i = \exp\left(-\frac{\|x - s_i\|^2}{2\sigma^2}\right) \quad (37)$$

Šio neurono elgesys labai skiriasi nuo biologinio. Šiame neurone sužadinimas nėra įvesties signalų svertinės sumos funkcija. Vietoj to apskaičiuojamas atstumas tarp įvesties ir saugomų duomenų. Jei šis atstumas lygus nuliui, neuronas atsako maksimaliu išėjimo dydžiu, lygiu vienetui. Šis neuronas sugeba atpažinti tam tikrus duomenis ir generuoti išėjimo signalus, kurie yra panašios funkcijos. Tinkamos neuroninio tinklo architektūros parinkimo žingsniai:

1. reikia teoriškai nuspėti mažinantį įvesties skaičių naudojant statistinius metodus;
2. naudoti evoliucinę skaičiavimo techniką;
3. naudoti bandymų ir klaidų metodą;
4. pradėti nuo didelės sistemos ir ją apkarpyti, derinant neuronus su visais duomenimis, kurie turi vienodus ar panašius atsakus;
5. pradėti nuo mažesnės sistemos ir prirėkus pridėti neuronų.

### 2.6. Pasikartojantys neuroniniai tinklai

Skirtingai nuo pažangių neuroninių tinklų, pasikartojančių neuroninių tinklų, neuronų išėjimai gali būti prijungti prie jų įėjimų. Taigi signalai tinkle gali nuolat cirkuliuoti.

#### 2.6.1. Hopfieldo neuroninis tinklas

Vieno sluoksnio pasikartojantį tinklą analizavo Hopfieldas [26] (1982). 21 paveiksle pavaizduotame tinkle yra vienpoliai kietojo slenksčio neuronai, kurių išėjimai yra lygūs 0 arba 1. Svoriai suteikiami

simetriška kvadratine matrica  $W$ , kurios pagrindinėje įstrižainėje su nulių elementų ( $w_{ij} = 0$  for  $i = j$ ). Sistemos stabilumas paprastai analizuojamas naudojant energijos funkciją:

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{ij} v_i v_j \quad (38)$$

Įrodyta, kad signalo cirkuliacijos metu, tinklo energija  $E$  mažėja ir sistema konverguoja į stabilius taškus. Tai ypač aktualu, kai sistemos išvesties vertės atnaujinamos asinchroniniu režimu. Tai reiškia, kad duotame cikle tik viena atsitiktinė išvestis gali būti pakeista į reikiamą vertę. Hopfieldas taip pat įrodė, kad tuos stabilius taškus, į kuriuos sistema konverguoja, galima užprogramuoti koreguojant svorį, naudojant modifikuotą Hebo taisyklę:

$$\Delta w_{ij} = \Delta w_{ji} = (2v_i - 1)(2v_j - 1) \quad (39)$$

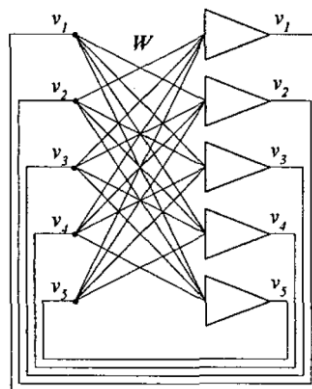
Tokios atminties talpa yra ribota. Remdamasis eksperimentais, Hopfieldas apskaičiavo, kad didžiausias saugomų modelių skaičius yra  $0,15N$ , kur  $N$  yra neuronų skaičius.

### 2.6.2. Autoasociatyvinė atmintis

Hopfieldas (1984) išplėtė savo tinklo sampratą ir įtraukė autoasociatyvinę atmintį. Toje pačioje tinklo struktūroje, kaip parodyta 21 paveiksle buvo naudojami bipoliniai neuronai, kurių išėjimai lygūs nuo  $-1$  iki  $+1$ . Šiame tinklo modelyje  $S_m$ , naudojant autokoreliacijos algoritmą, yra įrašomi į svorio matricą  $W$ .

$$W = \sum_{m=1}^M S_m S_m^T - MI \quad (40)$$

kur  $M$  yra saugomo modelio skaičius, o  $I$  yra vieninga matrica.  $W$  yra kvadratinė simetriška matrica, kurios elementai pagrindinėje įstrižainėje yra lygūs nuliui ( $w_{ji}$  for  $i = j$ ). Naudojant modifikuotą formulę, naujus duomenis galima pridėti arba atimti iš atminties. Kai tokia atmintis bus veikiamą dvejetainių dvipolių duomenų, vykdant pradines tinklo būsenas, po signalo cirkuliacijos, tinklas susilies į panašiausius saugomus duomenis arba juos papildys.

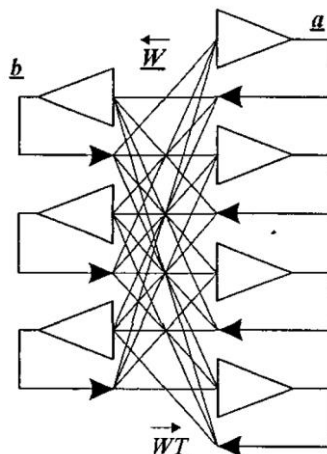


22 pav. Autoasociatyvinė atmintis [26]

Šis stabilus taškas bus artimiausias energijos funkcijos minimumui

$$E(v) = -\frac{1}{2}v^T W v \quad (41)$$

Kaip ir Hopfieldo tinkle, autoasociatyvinės atminties talpa yra ribota, manoma, kad ji yra apytiksliai lygi  $M_{max} \approx 0.15N$ . Kai saugomų šablonų skaičius yra didelis ir artimas atminties talpai, tinklas linkęs pereiti į netikrąsias būsenas, kurios nebuvo saugomos. Šios netikros būsenos yra papildomi energijos funkcijos minimumai.



23 pav. Dvikryptė autoasociatyvinė atmintis (DAA) [26]

### 2.6.3. Dvikryptė autoasociatyvinė atmintis (DAA)

Autoasociatyvinės atminties samprata buvo išplėsta į dvikryptės asociatyvinės atminties sampratą Kosko [26] (1987, 1988). 22 paveiksle pavaizduota atmintis gali susieti poras a ir b.

Tai yra dviejų sluoksnių tinklas, kurio antrojo sluoksnio išvestis yra tiesiogiai sujungta su pirmojo sluoksnio įvestimi. Antrojo sluoksnio svorio matrica yra  $W^T$ , o pirmojo sluoksnio –  $W$ . Stačiakampė svorio matrica  $W$  gaunama kaip kryžminės koreliacijos matricų suma:

$$W = \sum_{m=1}^M a_m b_m \quad (42)$$

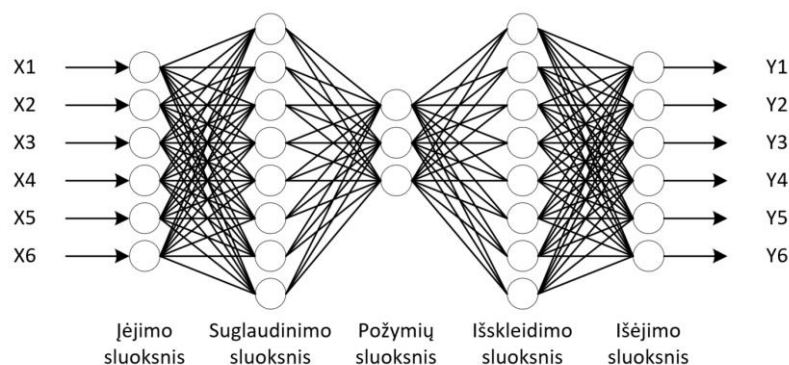
kur  $M$  yra saugomų porų skaičius,  $a_m$  ir  $b_m$  yra saugomos vektorių poros. Tokio tinklo koncepcija gali būti išplėsta sujungiant tris ar daugiau vektorių.

## 2.7. Autoasociatyvinis neuroninis tinklas

Autoasociatyvinis neuroninis tinklas (AANT) yra tinklas, kuriame išvestys yra mokomos imituoti įvestis per tinkamą dinaminį diapazoną. Jėgainės kintamieji, kurie turi tam tikrą tarpusavio ryšį, yra įvestis. AANT apmokymo metu kintamųjų tarpusavio ryšiai yra įterpiami į neuroninio tinklo ryšių svorius [2].

### 2.7.1. Autoasociatyvinio neuroninio tinklo architektūra

Autoasociatyviniai neuroniniai tinklai iš esmės yra tiesioginio sklidimo neuroniniai tinklai. 29 paveiksle parodyta bendra AANT architektūra. AANT architektūroje yra įvesties sluoksnis, keletas paslėptų sluoksnių ir išvesties sluoksnis.



24 pav. AANT bendroji architektūra

Teoriškai pakanka, kad AANT turėtų tris paslėptus sluoksnius [1]. Tačiau praktikoje buvo įrodyta, kad daugiau paslėptų sluoksnių padeda AANT pagerinti našumą. Papildomi paslėpti sluoksniai padeda AANT efektyviau atvaizduoti kintamųjų tarpusavio ryšį [3].

Pirmasis paslėptas sluoksnis vadinamas suglaudavimo sluoksniu. Suglaudavimo sluoksnyje esančių mazgų perdavimo funkcija gali būti sigmoidinė arba kitos panašios netiesinės funkcijos. Antrasis paslėptas sluoksnis vadinamas „butelio kaklelio“ arba požymių sluoksniu. Požymių sluoksnio matmenys yra mažiausi tinkle, šio sluoksnio perdavimo funkcija gali būti tiesinė arba netiesinė. Trečiasis arba paskutinis paslėptas sluoksnis vadinamas išskleidimo sluoksniu, kurio mazgų perdavimo funkcijos yra netiesinės, dažniausiai sigmoidinės.

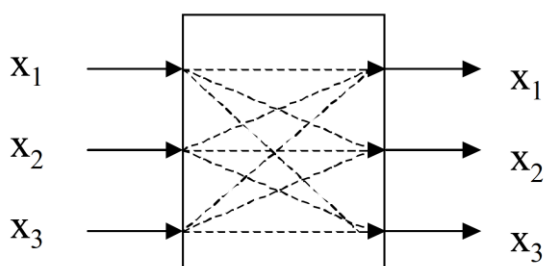
### 2.7.2. Autoasociatyviniai neuroniniai tinklai ir tapatybės atvaizdavimas

Autoasociatyviniai neuroniniai tinklai suteikia tapatybės atvaizdavimą. Tapatybės atvaizdavimo tema laikoma bendrojo atvaizdavimo pogrupiu. Atpažindami tapatybę, įvesties kintamieji susiejami su savimi, nesudarant paprasto individualaus atvaizdavimo. Kiekvienas išėjimas turėtų būti visų įėjimų funkcija. Pavyzdžiui 30 paveiksle pavaizduotas tapatumo atvaizdavimas turi padaryti kiekvieną  $X_i, i = 1, 2$  ir  $3$  kaip  $X_1, X_2, X_3$ , funkciją:

$$X_1 = f(X_1, X_2, X_3) \quad (43)$$

$$X_2 = f(X_1, X_2, X_3) \quad (44)$$

$$X_3 = f(X_1, X_2, X_3) \quad (45)$$



25 pav. Tapatybės atvaizdavimas

Nepaisant to, kad egzistuoja daugybė bendro atvaizdavimo metodų, dauguma jų nėra tinkami tapatybės atvaizdavimui. Kai tapatybės atvaizdavimui naudojama bendrojo atvaizdavimo technika, ji pasirenka lengviausią sprendimą, kuris sukuria paprastą išvesties – įvesties atvaizdavimą. Jei



tapatybės atvaizdavimo sudarymo technika neturi savybių, kad būtų išvengta tapatybės atvaizdavimo sudarymo nuo išvesties – įvesties, tai nėra tinkama tapatybės atvaizdavimui.

Autoasociatyviniai neuroniniai tinklai yra tinkami tapatybės atvaizdavimui, nes jie turi savybių, kad būtų išvengta „išvestis – įvestis“ atvaizdavimo. Šį vaidmenį atlieka požymių sluoksnis (antrasis paslėptasis sluoksnis). Apmokymo metu požymių sluoksnis priverčia AANT suglaudinti įvestis, tada jas išskleisti, kad gautų tinklo išvestis. Apmokymo procese pasirenkami tinklo svoriai, kad išvesties sluoksnyje iš naujo sukurtas matavimo vektorius kuo tiksliau atitiktų įvestį, bent jau nustatytos mažiausių kvadratų paklaidos ribose lyginant su apmokymo duomenų rinkiniu.

### 2.7.3. Autoasociatyvinio neuroninio tinklo apmokymas, taikant Levenbergo – Marquardto (LMA) algoritmą

Kadangi AANT yra tiesioginio sklidimo neuroniniai tinklai, jie apmokomi remiantis Levenbergo – Marquardto (angl. *Levenberg – Marquardt algorithm*) algoritmu [27], [28] ir [29]. Apmokymų metu neuroninis tinklas mokosi kintamųjų tarpusavio ryšių. Į AANT pateikiamų duomenų rinkinyje esantys kintamieji turi būti tarpusavyje susiję. Tai patikrinama analizuojant duomenų kovariacijos matricą. Jei kintamieji nėra susiję, AANT neras jokių ryšių, priešingu atveju, tinklas įsimins duomenis.

Levenbergo – Marquardto algoritmas (LMA) yra skaitinis funkcijos, dažniausiai netiesinės, minimizavimo problemos matematinis sprendimas virš funkcijos parametru erdvės. Ši minimizavimo problema išskyla mažiausių kvadratų (angl. *least squares*) kreivės apibrėžimo paieškoje. LMA interpoliuoja tarp Gauso – Niutono algoritmo (GNA) ir gradientinio nuolydžio metodo. LMA yra labiau atsparus nei GNA. Tai reiškia, kad daugelyje atvejų jis suranda sprendimą, net jei pradžios taškas buvo labai toli nuo galutinio minimumo. Kitą vertus, gerai besielgiančioms funkcijoms ir su gerais pradiniais parametrais, LMA linkęs veikti šiek tiek lėčiau už GNA.

LMA yra labai populiarus kreivių funkcijų paieškos algoritmas, daugumoje programinės įrangos, kuri turi pagrindinius kreivių funkcijų radimo įrankius, realizuotas ir LMA.

Algoritmo pagrindinė užduotis yra mažiausių kvadratų apibrėžimo problema: turint aibę empirinių duomenų porų  $(t_j, y_j)$ , optimizuoti modelio kreivės  $f(t|p)$  parametrus  $p$  taip, kad išvestinių kvadratų suma

$$S(p) = \sum_{i=1}^m [y_i f(t_i|p)]^2 \quad (46)$$

taptų mažiausia.

Kaip ir kiti skaitiniai algoritmai, Levenbergo – Marquardto algoritmas yra iteratyvi procedūra. Pradedant minimizaciją, vartotojas turi pateikti pradines reikšmes parametro vektoriui  $p$ , jas galima parinkti ir atsitiktinai. Bendru atveju standartinis pradinių duomenų apibrėžimas kaip  $p^T = (1, 1, \dots, 1)$  suveiks gerai, kitais atvejais algoritmas konverguoja tik tada, kai parametru vektorius jau yra sąlyginai arti galutinio sprendinio. Kiekviename iteracijos žingsnyje parametro vektorius  $p$  yra pakeičiamas naujai apskaičiuotu  $p + q$ . Siekiant nustatyti  $q$ , funkcijos  $f_j(p + q)$  yra aproksimuojamos pagal jų linearizacijas

$$f(p + q) \approx f(p) + J(q) \quad (47)$$

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

Kvadratų sumos  $S$  minimume turime  $\nabla_q S = 0$ . Apskaičiavus lygties dešinėsios pusės išvestinę ir prilyginus ją nuliui, gauname:

$$(J^T J)q = -J^T f \quad (48)$$

iš kur galima apskaičiuoti  $q$  transponuojant  $J^T J$ . LMA pagrindas yra šios lygties pakeitimas „prislopintu“ variantu

$$(J^T J + \lambda I)q = -J^T f \quad (49)$$

Slopinimo koeficientas (ne neigiamas)  $\lambda$  yra koreguojamas kiekvienoje iteracijoje. Sparčiai mažėjant  $S$  galima naudoti mažesnę reikšmę. Taip algoritmas supanašės su GNA. Kitu atveju, jei iteracija nepakankamai sumažina liekaną,  $\lambda$  reikšmė gali būti padidinta, taip priartinant žingsniu arčiau prie gradientinio nuolydžio krypties. Jeigu gauto žingsnio ilgis arba kvadratų sumos redukcijos iki parametro vektoriaus  $p$  tampa trumpesnis už apibrėžtą minimumą, iteravimas nutraukiamas. Paskutinis parametro vektorius  $p$  yra laikomas sprendimu.

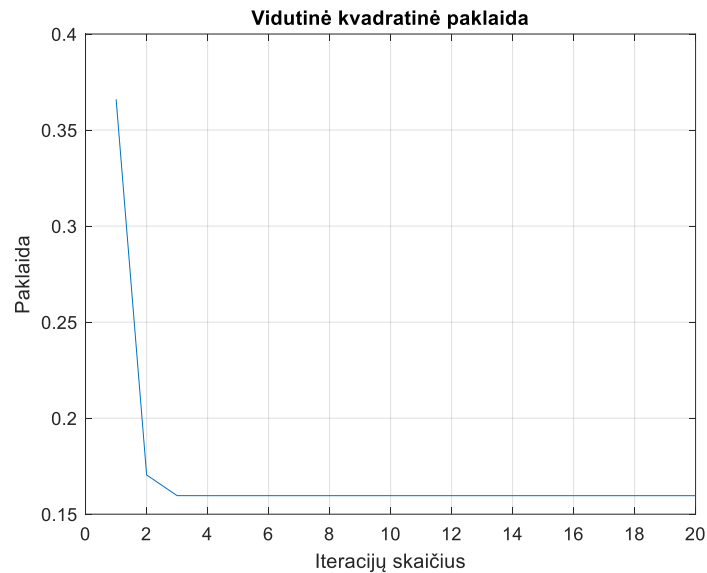
#### 2.7.4. AANT apmokymas triukšmingose situacijose

Apmokymas yra labai svarbus neuroniniams tinklams. Geras apmokymas lemia geras apibendrinimo galimybes, kai apmokytas tinklas gali gerai apytiksliai apibendrinti mokymo rinkinio pavyzdžius. Apmokymo metu, tinklas turi tiksliai išmokti įvesties ir išvesties duomenų sąveiką. Per didelis tikslumas turi priešingą efektą. Jei tinklas per daug gerai išmoksta duomenis, jo apibendrinimo galimybė sumažėja. Jis negali gerai apibendrinti duomenų iš apmokymo duomenų rinkinio.

Kai duomenys yra triukšmingi, jie turi netikrumo lygį. Neapibrėžties lygis priklauso nuo triukšmo lygio. Didesnis triukšmas sukelia didesnę atskirų matavimų neapibrėžtumą. Jei neuroninis tinklas, tvarkantis triukšmingus duomenis, bus priverstas išmokti duomenis tikslumu, viršijančiu neapibrėžtumo lygį, jis įsimins duomenis. Iš pradžių tinklas bando sužinoti duomenis. Jis bando rasti ryšį tarp įvesties ir išvesties. Pamačius, kad joks ryšys negali patenkinti norimo tikslumo, jis ima įsiminti pavyzdžius po vieną. Šioje situacijoje mokymosi procesas užima daug laiko.

#### 2.7.5. AANT našumas triukšmingose situacijose

AANT apmokyti naudojami jutiklių matavimo duomenys iš DCS sistemos. Duomenų rinkinys naudojamas apmokyti tinklą pavaizduotas 39 paveiksle. Šie duomenys jau iš karto yra su triukšmu. AANT architektūra buvo parinkta 7-8-3-8-7 neuroninis tinklas, o aktyvavimo funkcija tangentinė. Iš pradžių apmokymo tikslumas buvo nustatytas 0,001. Po 300 iteracijų AANT nepasiekė norimo tikslumo. 31 paveiksle parodytas santykis tarp iteracijų ir vidutinės kvadratinės paklaidos. Nustačius apmokymo tikslumą į 0,1 AANT nepasiekia norimo tikslumo net ir po 300 ar 200 iteracijų. Tai reiškia, kad 0,01 apmokymų tikslumas yra didesnis nei neapibrėžtumas, kurį sukelia triukšmas, esantis apmokymo duomenyse. Trečiuoju bandymu buvo nustatytas apmokymo tikslumas 1. Šį kartą norimą tikslumą AANT pasiekė mažiau nei per 20 iteracijų.



**26 pav.** Santykis tarp vidutinės kvadratinės paklaidos = 1 ir 20 iteracijų

AANT sumodeliuoti ir apmokyti buvo naudojamos MATLAB funkcijų paprogramės [30]: duomenų normalizavimo – DSCALE.m, AANT suformavimo paprogramė – NNEVALE4E.m, NNEVAL4ERR.m, NNEVAL4NN.m, tinklo apmokymo paprogramė MARQX2NN.m ir aktyvavimo funkcijos PMNTANH.m paprogramė [30]. Paprogramės pateikiamos „MathWorks“ internetiniame puslapyje. Tinklo vykdomoji, apmokymo ir testavimo MATLAB programiniai kodai pateikiami priede Nr. 1.

### 3. Kietu kuru kūrenamos kogeneracinės jėgainės, termofikacinio vandens aušyklės AANT tyrimas

#### 3.1.1. Kietu kuru kūrenamos kogeneracinės jėgainės, termofikacinio vandens aušyklės schema ir matavimų duomenys

Kietu kuru kūrenamos kogeneracinės jėgainės, termofikacinio vandens aušyklės jutiklių matavimo duomenys bus naudojami, siekiant išbandyti sumodeliuotą, autoasociatyviniais neuroniniais tinklais pagrįstos jutiklių monitoringo ir matavimų virtualizavimo sistemą. Šiame skyriuje paaiškinamas kietu kuru kūrenamos kogeneracinės jėgainės, termofikacinio vandens aušyklės veikimas, naudojami valdymo sistemos matavimo duomenys.

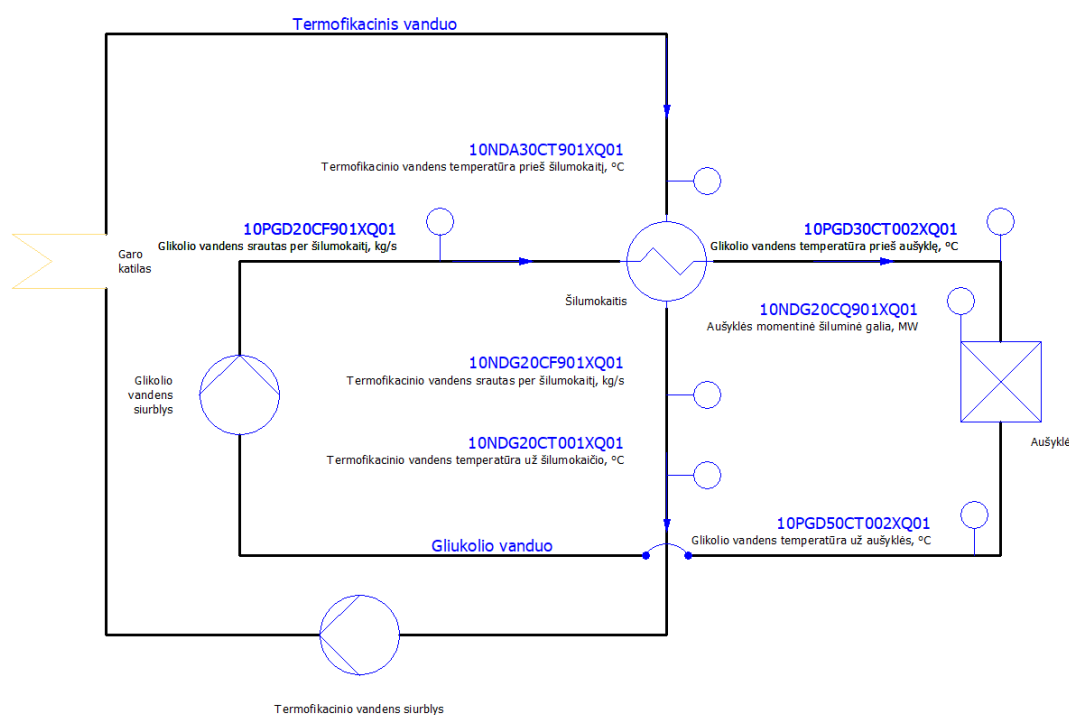
#### 3.1.2. Termofikacinio vandens aušyklės veikimo principas

Kietu kuru kūrenamos kogeneracinės jėgainės, termofikacinio vandens aušyklė skirta ataušinti termofikacinį vandenį ir tokiu būdu atiduoti į aplinką perteklinę šilumos energiją, kad kogeneracinė jėgainė nepertraukiamai galėtų dirbti ir gaminti elektros energiją, kuri yra atiduodama į šalies skirstomuosius elektros tinklus.

Kogeneracinė jėgainė gamina garą, kuris keliauja į garo turbiną. Atidirbęs garas, kondensuojamas ir likusi šilumą perduodama į termofikacinio vandens šildymo sistemą, toliau šiluma atiduodama į miesto šildymo tinklus. Sumažėjus miesto šilumos poreikiui arba miesto šilumos tinklams visai atsijungiant nuo jėgainės šildymo sistemos šiluma yra išaušinama termofikacinio vandens aušyklėje.

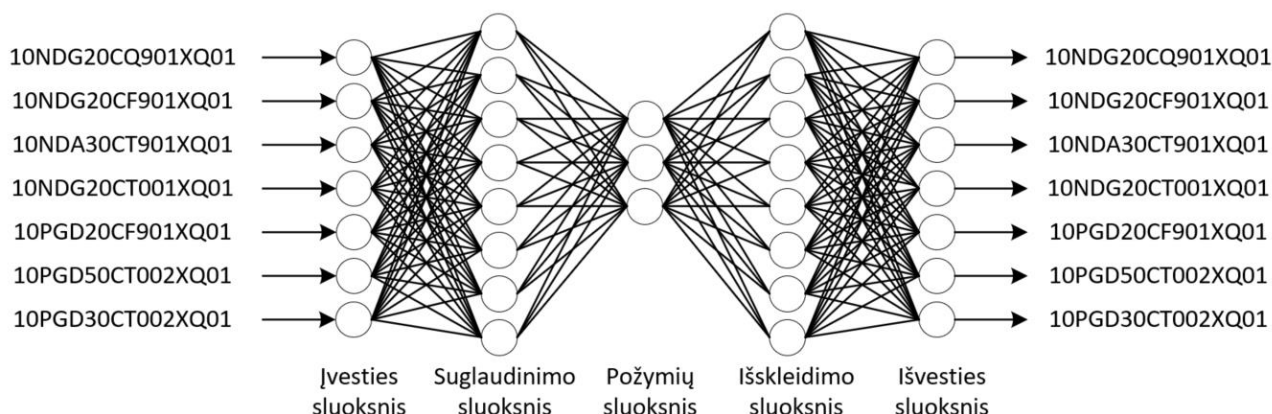
#### 3.1.3. Termofikacinio vandens aušyklės schema ir matavimai

32 paveiksle pateikiama supaprastinta aušyklės vamzdynų ir prietaisų schema (angl. *P&ID – Piping and instrumentation diagram*), kurioje parodomi vamzdynai ir proceso įranga kartu su prietaisų ir valdymo įtaisais.



27 pav. Termofikacinio vandens aušyklės P&ID schema

Kietu kuru kūrenamos kogeneracinės jėgainės, termofikacinio vandens aušyklės AANT modelis ir architektūra. AANT modeliuojamas su septyniais įėjimais, trimis paslėptais sluoksniais ir septyniais išėjimais.



**28 pav.** Termofikacinio vandens aušyklės AANT modelis ir architektūra

Termofikacinio vandens aušyklės AANT įvestys (angl. *inputs*):

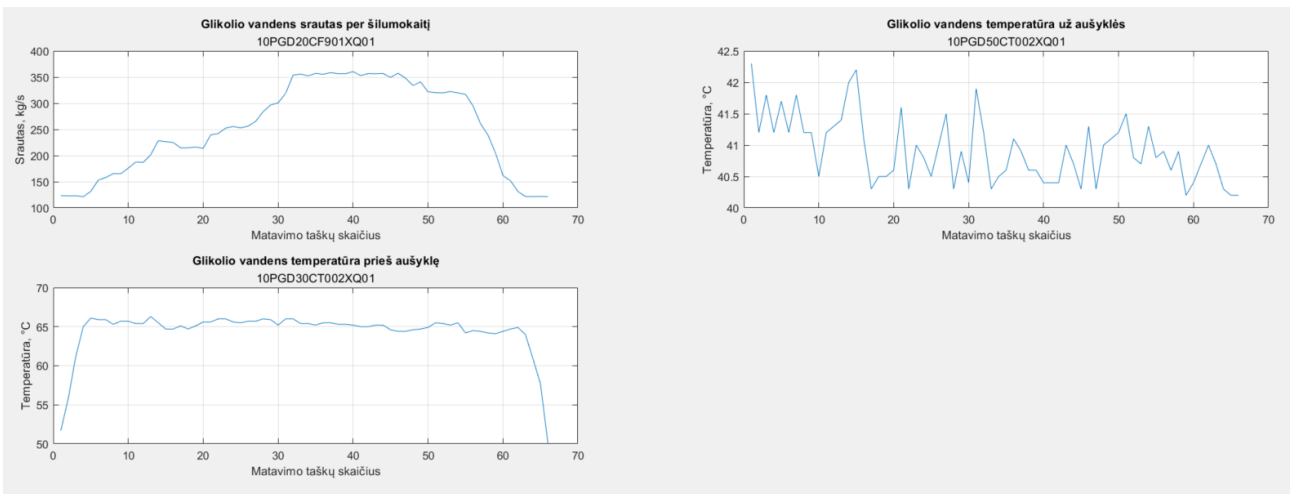
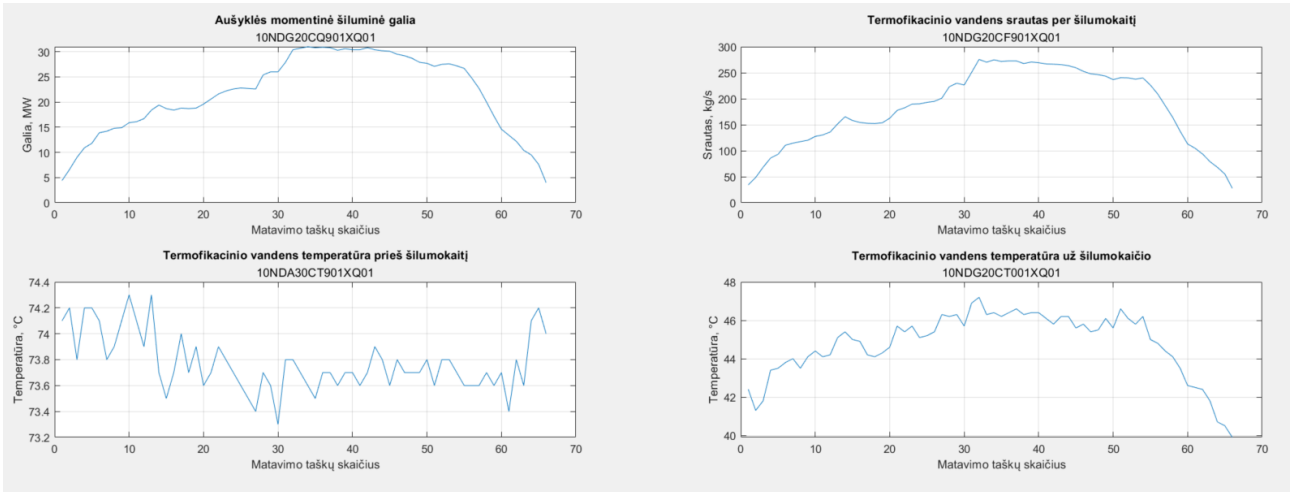
1. aušyklės momentinė šiluminė galia: 10NDG20CQ901XQ01, MW.
2. termofikacinio vandens srautas per šilumokaitį: 10NDG20CF901XQ01, kg/s.
3. termofikacinio vandens temperatūra prieš šilumokaitį: 10NDA30CT901XQ01, °C.
4. termofikacinio vandens temperatūra už šilumokaičio: 10NDG20CT001XQ01, °C.
5. glikolio vandens srautas per šilumokaitį: 10PGD20CF901XQ01, kg/s.
6. glikolio vandens temperatūra už aušyklės: 10PGD50CT002XQ01, °C.
7. glikolio vandens temperatūra prieš aušyklę: 10PGD30CT002XQ01, °C.

Termofikacinio vandens aušyklės AANT išvestys (angl. *outputs*):

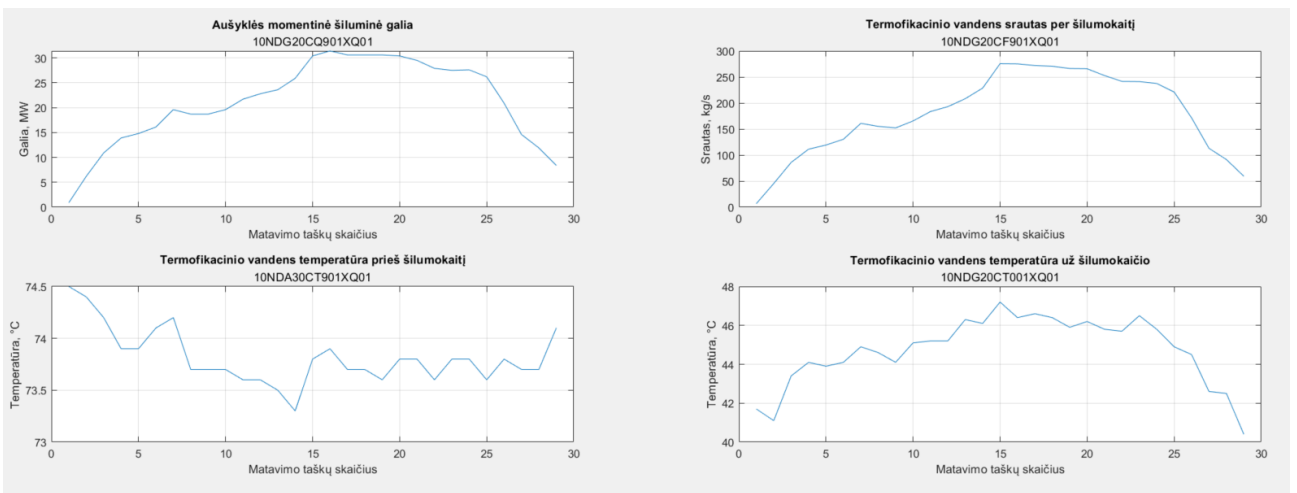
1. aušyklės momentinė šiluminė galia: 10NDG20CQ901XQ01, MW.
2. termofikacinio vandens srautas per šilumokaitį: 10NDG20CF901XQ01, kg/s.
3. termofikacinio vandens temperatūra prieš šilumokaitį: 10NDA30CT901XQ01, °C.
4. termofikacinio vandens temperatūra už šilumokaičio: 10NDG20CT001XQ01, °C.
5. glikolio vandens srautas per šilumokaitį: 10PGD20CF901XQ01, kg/s.
6. glikolio vandens temperatūra už aušyklės: 10PGD50CT002XQ01, °C.
7. glikolio vandens temperatūra prieš aušyklę: 10PGD30CT002XQ01, °C.

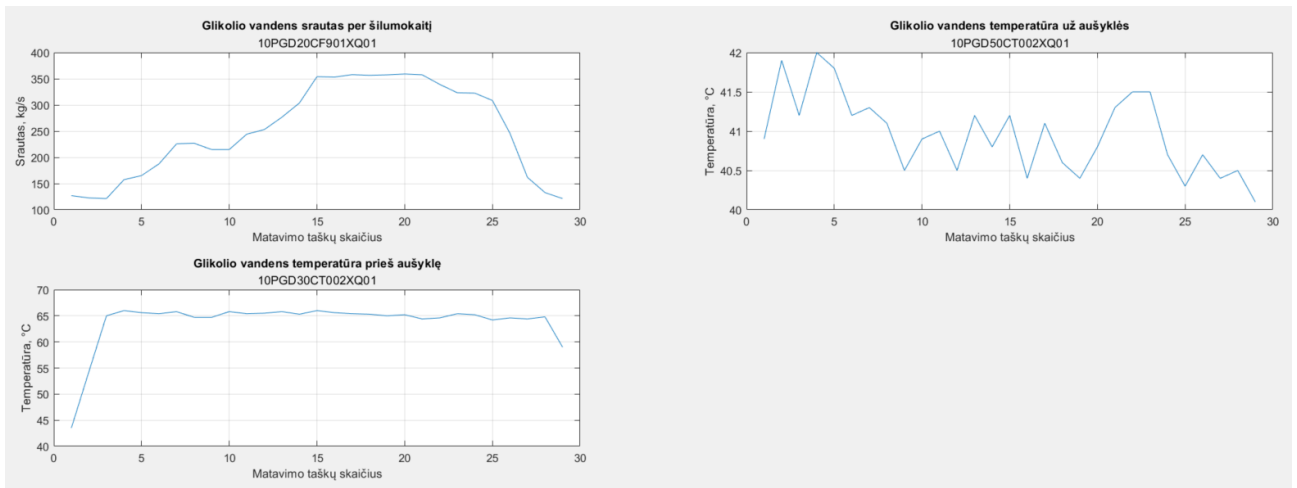
### 3.1.4. Termofikacinio vandens aušyklės matavimo duomenys

Iš kogeneracinės jėgainės, paskirstytos valdymo sistemos (angl. *DCS – Distributed control system*), istorinių proceso matavimo duomenų (angl. *process historian*) buvo paimti 95 matavimo taškai. Duomenys buvo paimti aušyklės galios didėjimo metu (aušyklės apkrovimas) ir aušyklės galios mažėjimo metu (aušyklės nukrovimas). Atsitiktine tvarka buvo atrinkti 29 matavimo taškai, o likusieji 66 matavimo taškai buvo naudojami kaip apmokymo rinkinys. Apmokymo rinkinys naudojamas AANT apmokymui, o testavimo rinkinys naudojamas apmokytam AANT išbandyti – testuoti. 34 paveiksle parodytas apmokymo rinkinys, o 35 paveiksle parodytas testavimo duomenų rinkinys.



29 pav. AANT apmokymo duomenų rinkinys



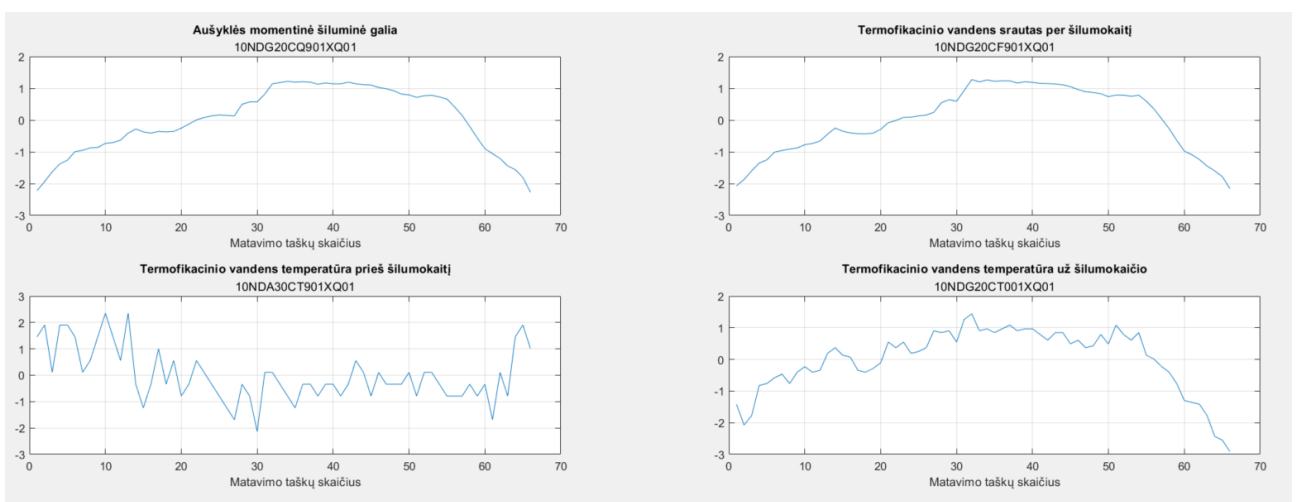


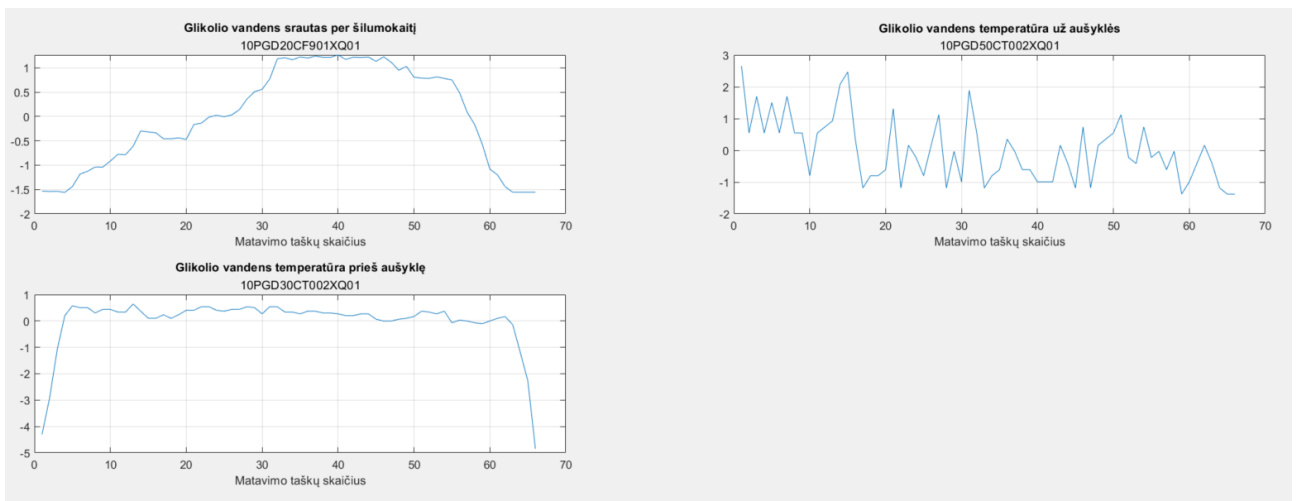
30 pav. AANT testavimo duomenų rinkinys

Kadangi matavimo duomenys paimti veikiant jėgainei, aušyklės apkrovimas ir nukrovimas laike yra ilgas procesas, kuris priklauso tiek nuo miesto šilumos poreikio, tiek nuo elektros gamybos poreikio. Aukščiau atrinkti matavimo taškai yra labiausiai atspindintys aušyklės kintamųjų priklausomybę vienas nuo kito. Valdymo sistemos matavimo duomenų diskretizavimas į istorinių duomenų serverį vyksta kas 1 sekundę, atrinkti taškai skirti tinklo apmokymui diskretizuojami kas 30 minučių, siekiant sumažinti kintamųjų triukšmų lygį, taip pat sumažinti matavimo taškų skaičių, kad išvengtų ilgai trunkančios tinklo apmokymo procedūros.

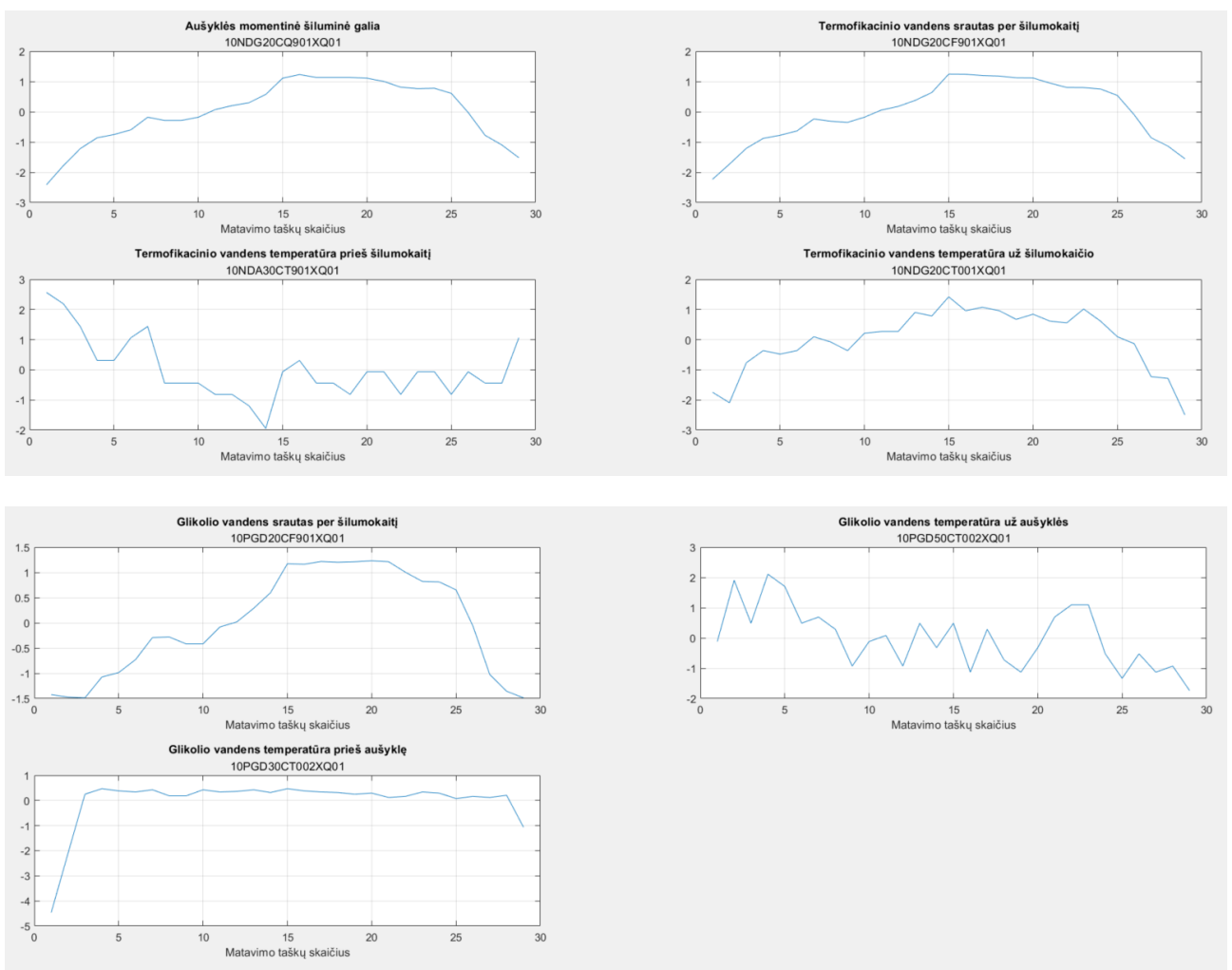
### 3.1.5. Matavimo duomenų normalizavimas

Tinklo apmokymui skirti duomenys normalizuojami naudojant DSCALE.m paprogramę [30], kurioje duomenys normalizuojami kiekvieno duomenų vektoriaus, nulio vidurkio (angl. „zero mean“) ir standartinio nuokrypio (angl. „variance 1“) funkcija. Šis normalizavimas neleidžia vienam jutikliui pakreipti rezultatų dėl didelių ar mažų skaitinių verčių. 36 paveiksle parodytas normalizuotų apmokymo duomenų rinkinys, o 37 paveiksle – normalizuotų testavimo duomenų rinkinys.





**31 pav.** Normalizuotas AANT apmokymo duomenų rinkinys

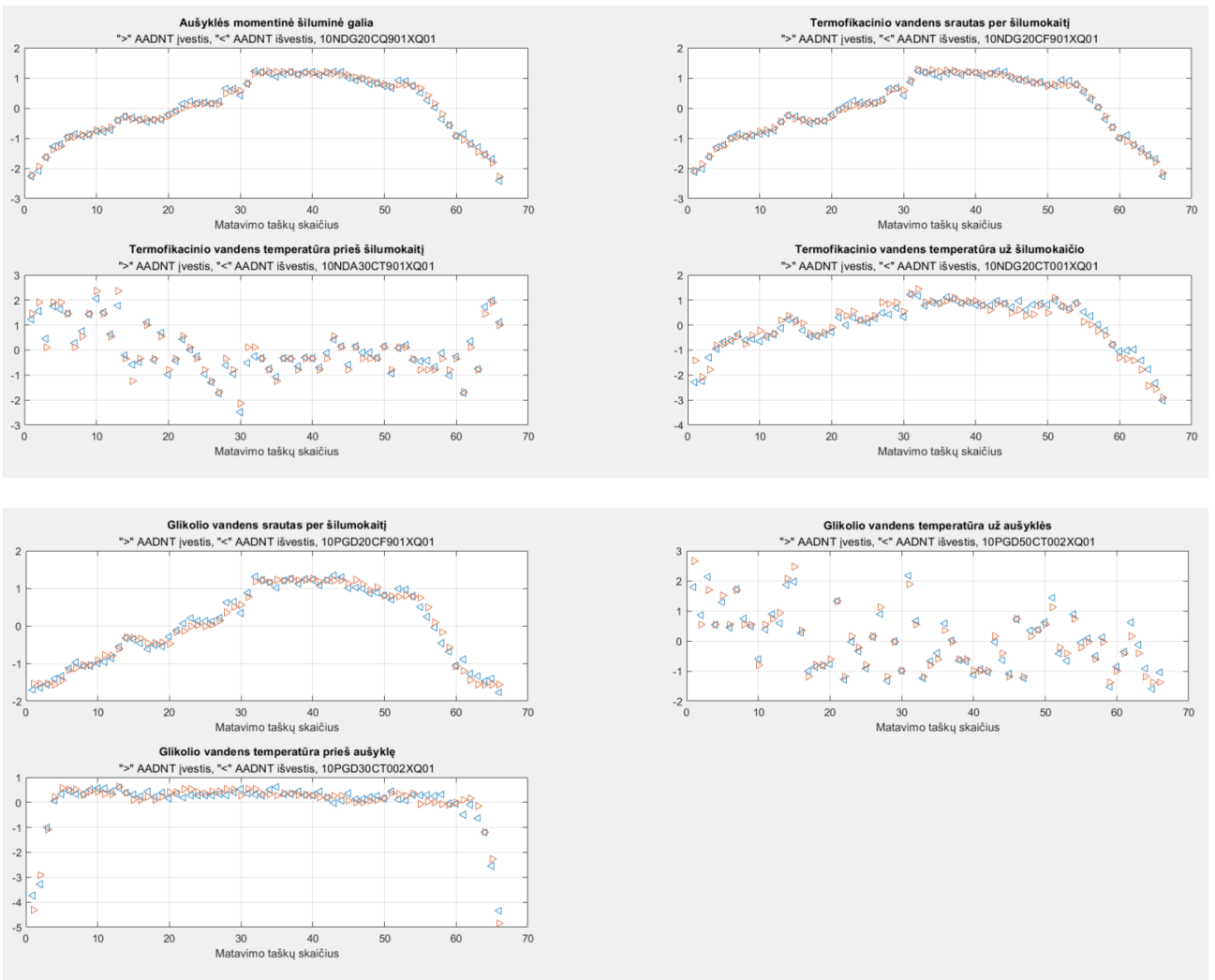


**32 pav.** Normalizuotas AANT testavimo duomenų rinkinys

### 3.1.6. Apmokyto AANT rezultatai

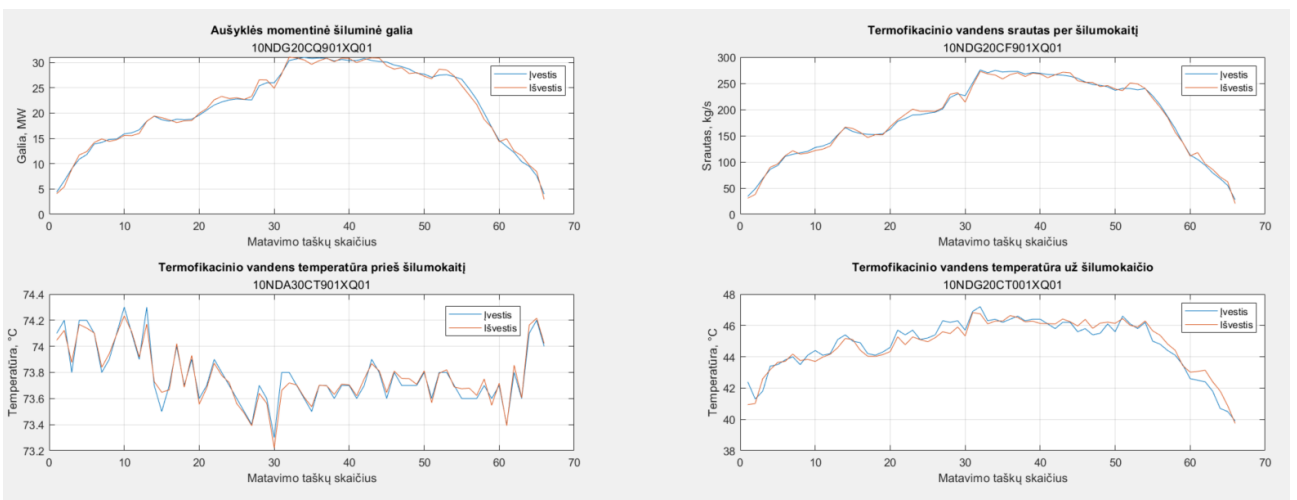
Aukščiau pateiktos AANT architektūros apmokymas su pateiktais kogeneracinės jėgainės duomenimis vyksta 897 sekundes. Toliau pateikiami AANT apmokymo rezultatai.

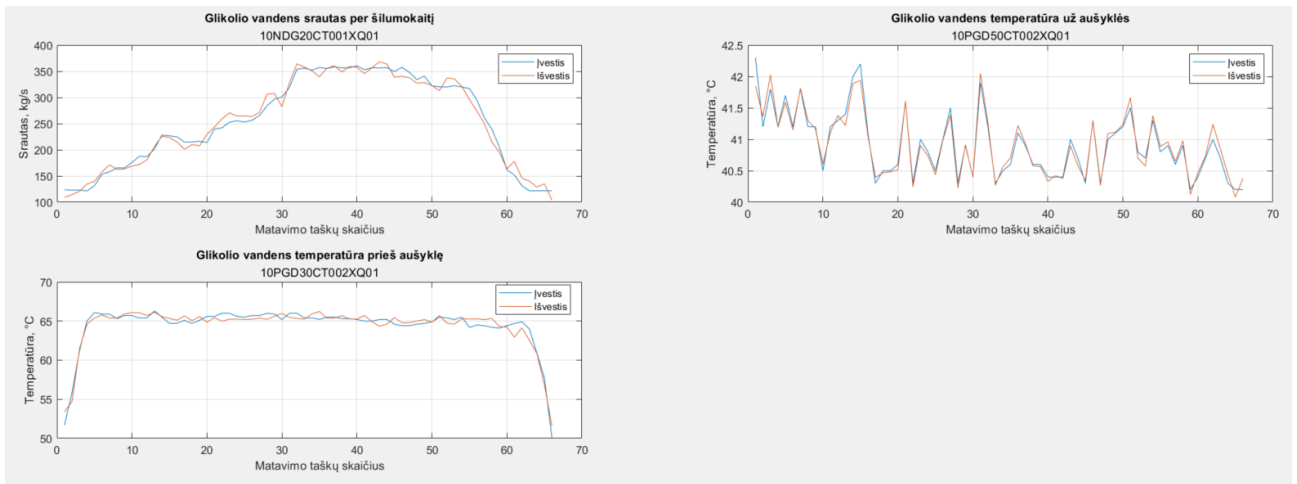




**33 pav.** Apmokyto AANT, normalizuoti apmokymo duomenys tinklo įvestyje („>“) ir išvestyje („<“)

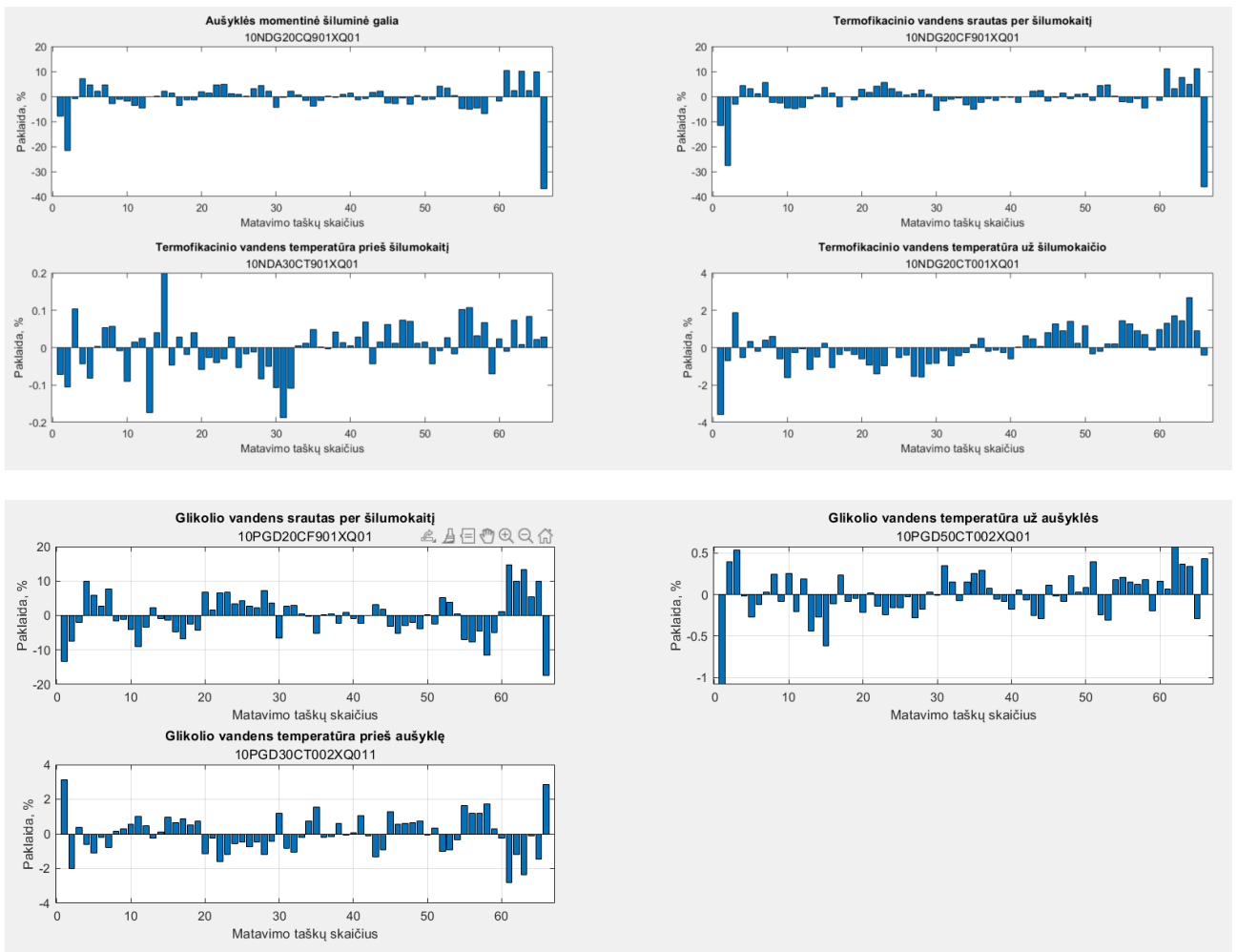
Toliau pateikiami AANT apmokymo rezultatų grafikai fizikiniais dydžiais.





34 pav. AANT apmokymo rezultatai fizikiniais dydžiais

Iš AANT apmokymo rezultatų grafikų matosi, kad apmokytas tinklas sugeba atkartoti kiekvienos įvesties duomenis tinklo išvestyje. Tinklo tikslumui įvertinti toliau pateikiamos AANT apmokymo rezultatų santykinės procentinės paklaidos kiekviename apmokymo taške, lyginant duomenis tinklo įėjime su duomenimis tinklo išėjime.



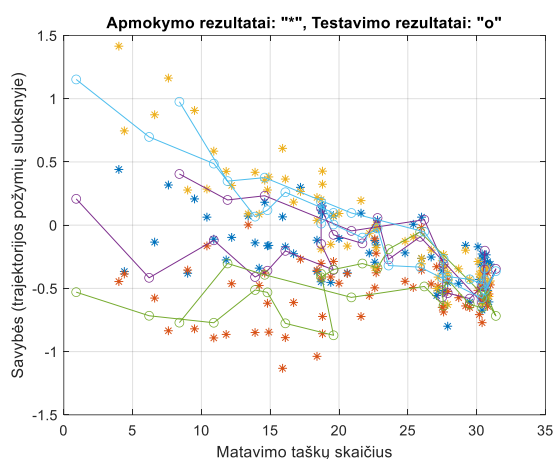
35 pav. AANT apmokymo rezultatų santykinės procentinės paklaidos

**1 lentelė.** AANT apmokymo rezultatų vidutinės absoliutinės procentinės paklaidos

KKS kodas	Signalų pavadinimas	Vidutinė absoliutinė procentinė paklaida, % (MAPE)
10NDG20CQ901XQ01	Aušyklės momentinė šiluminė galia	3,5896 %
10NDG20CF901XQ01	Termofikacinio vandens srautas per šilumokaitį	3,727 %
10NDA30CT901XQ01	Termofikacinio vandens temperatūra prieš šilumokaitį	0,0495 %
10NDG20CT001XQ01	Termofikacinio vandens temperatūra už šilumokaitį	0,7551 %
10PGD20CF901XQ01	Glikolio vandens srautas per šilumokaitį	4,5966 %
10PGD50CT002XQ01	Glikolio vandens temperatūra už aušyklės	0,2085 %
10PGD30CT002XQ01	Glikolio vandens temperatūra prieš aušyklę	0,8557 %

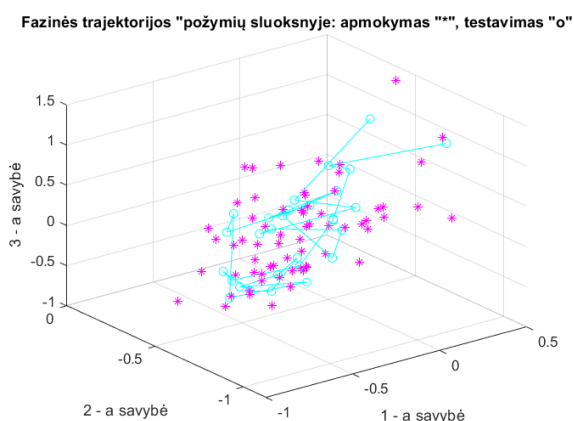
Vidutinė absoliutinė procentinė paklaida (*angl. MAPE*) – tai paklaida, kuri nusako santykinę prognozavimo tikslumą ir kuria galima palyginti skirtingų rodiklių prognozes. Prognozavimas labai tikslus, kai  $MAPE < 10\%$  ir tikslus, kai  $10\% < MAPE < 20\%$ . Kadangi kiekvieno kintamojo vidutinė absoliutinė procentinė paklaida neviršija  $10\%$ , galime daryti išvadą, kad tinklas apmokytas labai tiksliai [31].

Apmokymo duomenų savybės – trajektorijos požymių sluoksnyje:



**36 pav.** AANT apmokymo duomenų savybės („\*“ – kiekvieno kintamojo apmokymo rezultatai, „o“ – kiekvieno kintamojo testavimo rezultatai)

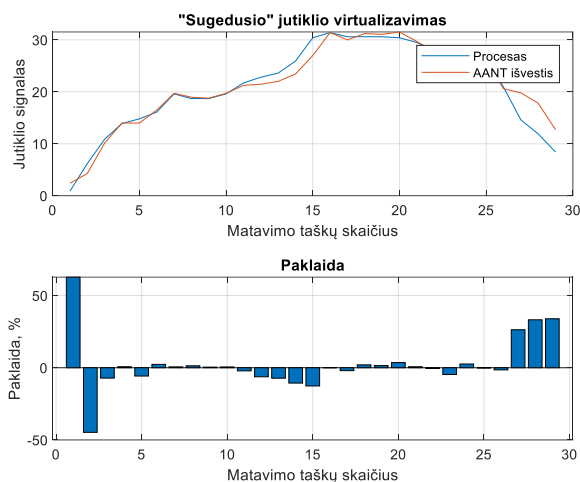
Apmokymo duomenų fazinės trajektorijos požymių sluoksnyje:



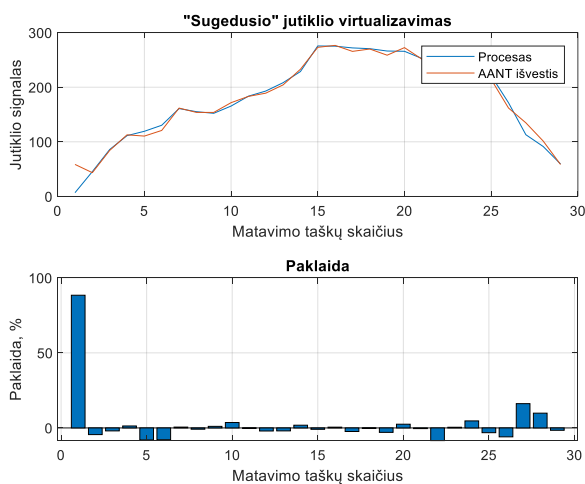
**37 pav.** AANT fazinės trajektorijos požymių sluoksnyje („\*“ – apmokymas, „o“ – testavimas)

### 3.1.7. „Sugedusio“ jutiklio virtualizavimo naudojant apmokytą AANT rezultatai

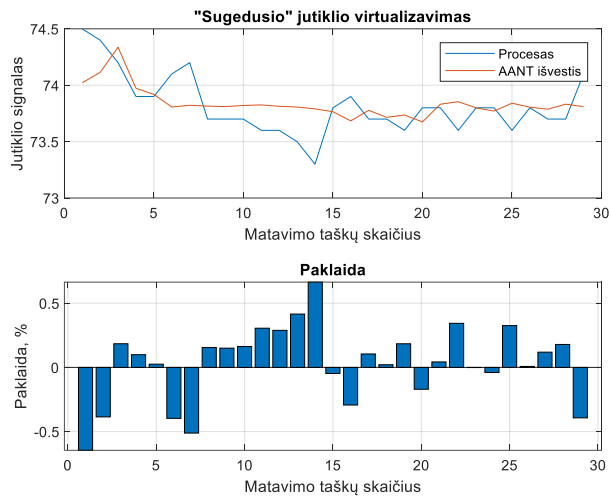
Apmokytas AANT toliau bandomas, tiriant „sugedusio“ jutiklio atkūrimą – virtualizavimą. Jutiklio gedimas imituojamas jį atjungiant, atkūrimas vykdomas naudojant „fminsearch“ MATLAB funkciją. Grafikuose pateikiamas kiekvienos tinklo išvesties atkūrimas, lyginant išvestį su jėgainės matavimo duomenimis (procesas). Šalia pateikiamos gautų rezultatų santykinės procentinės paklaidos kiekviename matavimo taške.



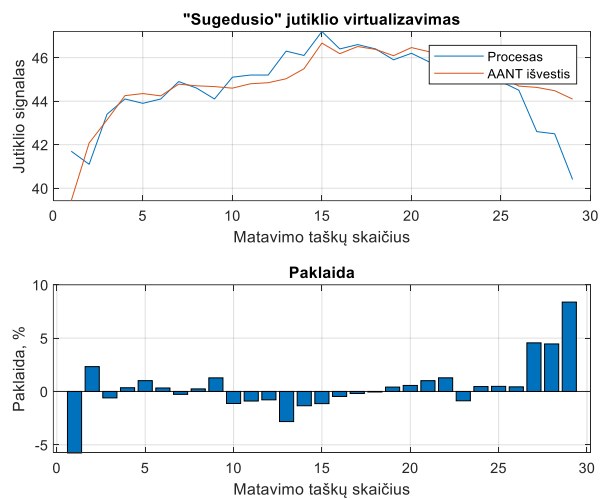
38 pav. „Sugedusio“ jutiklio Nr. 1, 10NDG20CQ901XQ01, MW – aušyklės momentinė šiluminė galia, vertės atkūrimas – virtualizavimas



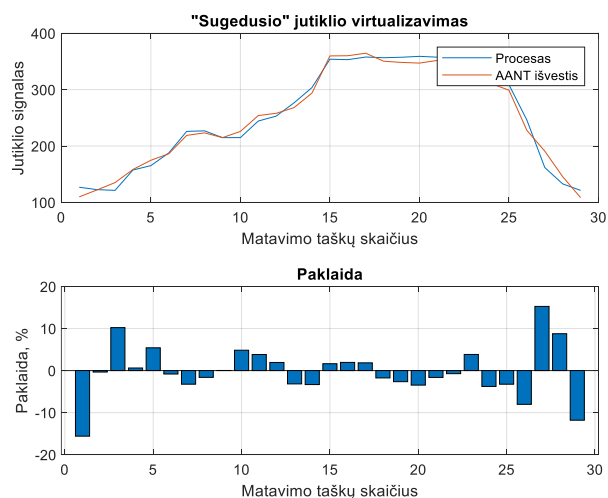
39 pav. „Sugedusio“ jutiklio Nr. 2, 10NDG20CF901XQ01, kg/s – termofikacinio vandens srautas per šilumokaitį, vertės atkūrimas – virtualizavimas



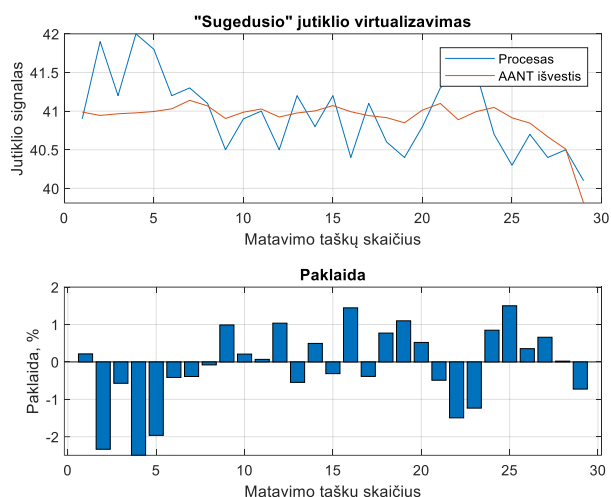
**40 pav.** „Sugedusio“ jutiklio Nr. 3, 10NDA30CT901XQ01, °C – termofikacinio vandens temperatūra prieš šilumokaitį, vertės atkūrimas – virtualizavimas



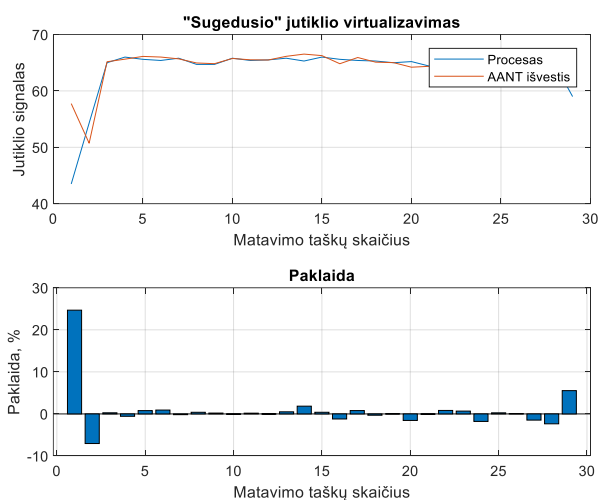
**41 pav.** „Sugedusio“ jutiklio Nr. 4, 10NDG20CT001XQ01, °C – termofikacinio vandens temperatūra už šilumokaičio, vertės atkūrimas – virtualizavimas



**42 pav.** „Sugedusio“ jutiklio Nr. 5, 10PGD20CF901XQ01, kg/s – glikolio vandens srautas per šilumokaitį, vertės atkūrimas – virtualizavimas



**43 pav.** „Sugedusio“ jutiklio Nr. 6, 10PGD50CT002XQ01, °C – glikolio vandens temperatūra už aušyklės, vertės atkūrimas – virtualizavimas



**44 pav.** „Sugedusio“ jutiklio Nr. 7, 10PGD30CT002XQ01, °C – glikolio vandens temperatūra prieš aušyklę, vertės atkūrimas – virtualizavimas

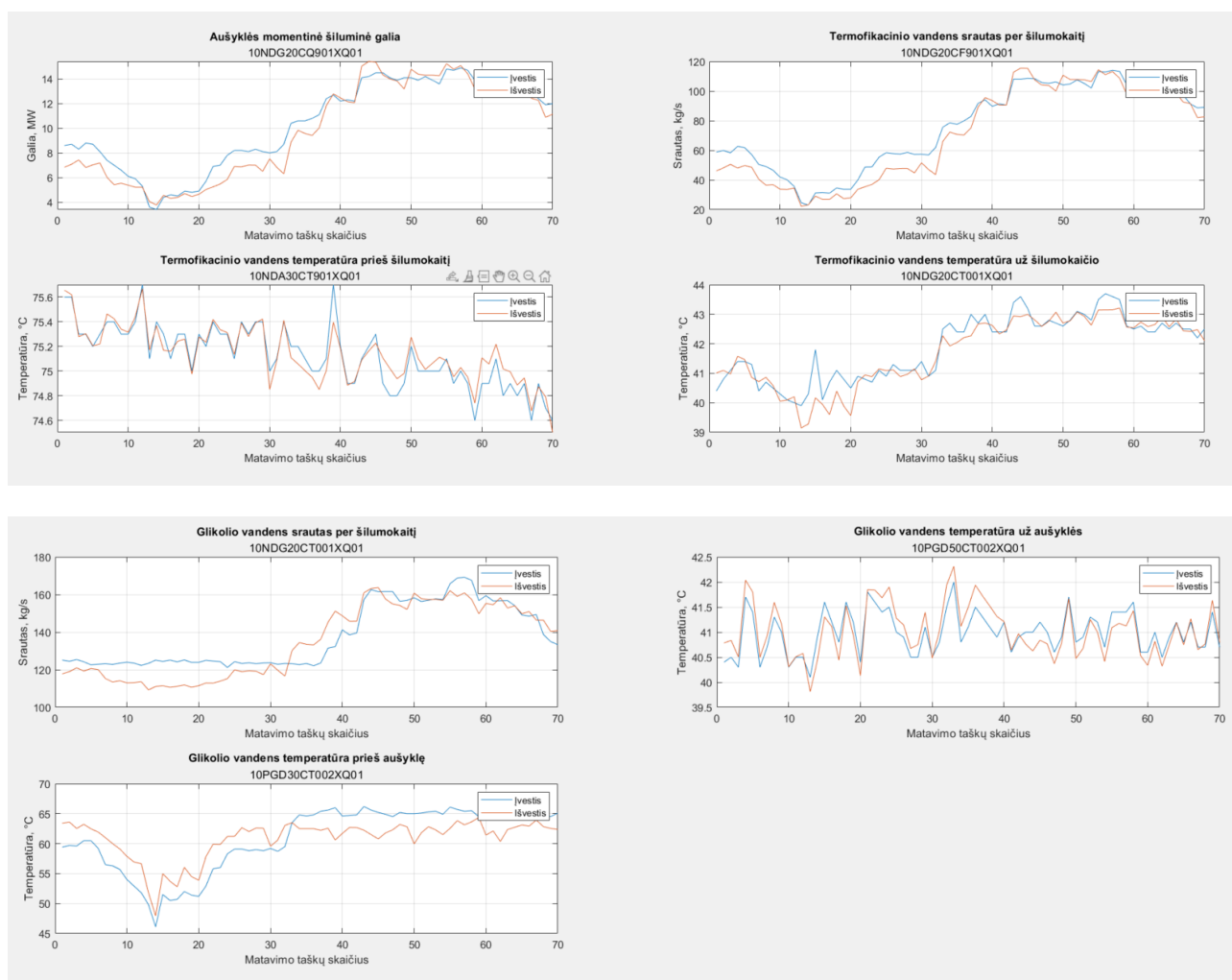
**2 lentelė.** AANT virtualizuotų matavimų, vidutinės absoliutinės procentinės paklaidos

KKS kodas	Signalų pavadinimas	Vidutinė absoliutinė procentinė paklaida, % (MAPE)
10NDG20CQ901XQ01	Aušyklės momentinė šiluminė galia	6,0475 %
10NDG20CF901XQ01	Termofikacinio vandens srautas per šilumokaitį	5,7115 %
10NDA30CT901XQ01	Termofikacinio vandens temperatūra prieš šilumokaitį	0,2013 %
10NDG20CT001XQ01	Termofikacinio vandens temperatūra už šilumokaičio	0,8232 %
10PGD20CF901XQ01	Glikolio vandens srautas per šilumokaitį	4,7361 %
10PGD50CT002XQ01	Glikolio vandens temperatūra už aušyklės	1,0452 %
10PGD30CT002XQ01	Glikolio vandens temperatūra prieš aušyklę	1,0394 %

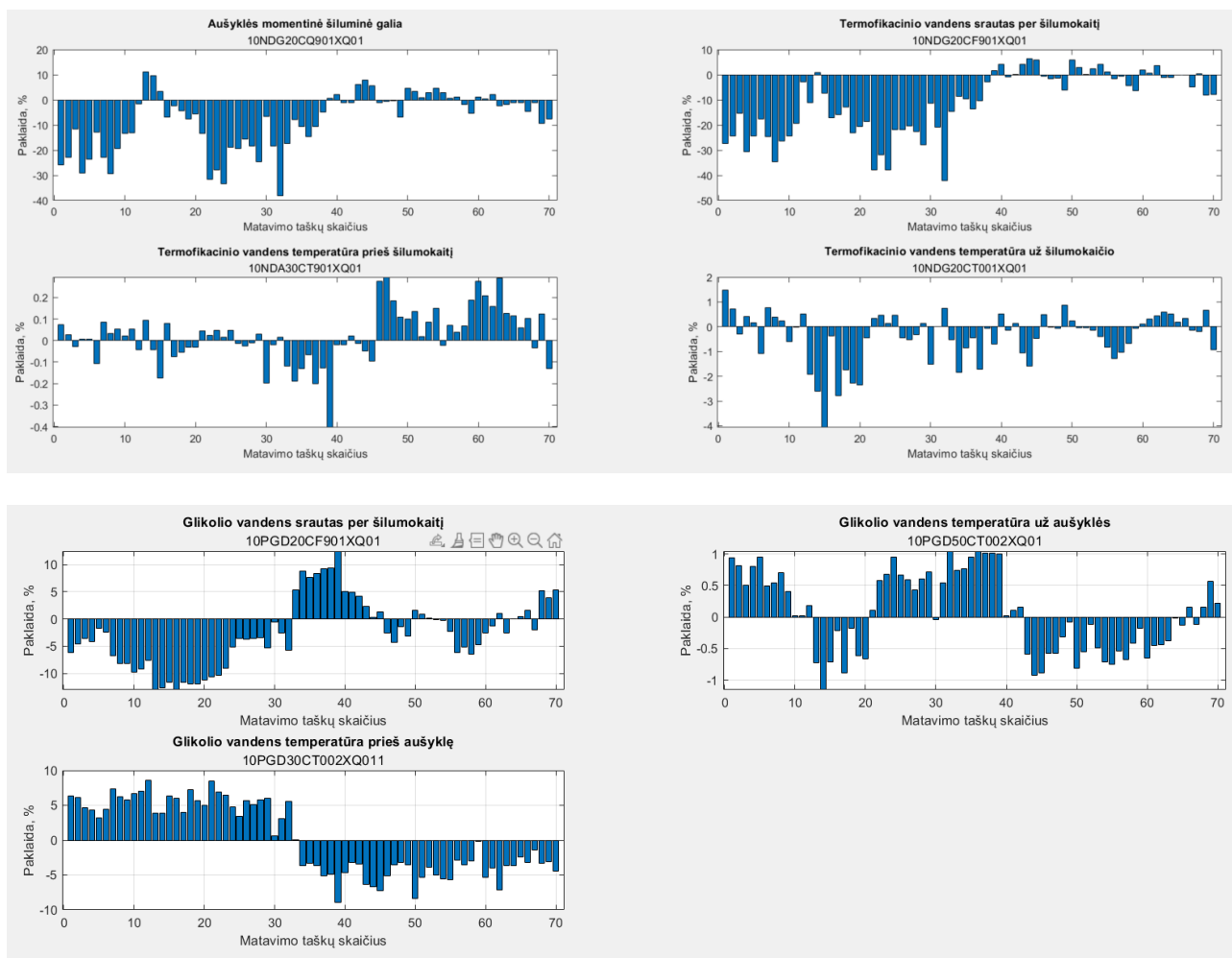
Didžiausios vidutinės absoliutinės procentinės paklaidos vykdant tinklo apmokymą buvo aušyklės momentinės šilumos galios, termofikacinio vandens srauto per šilumokaitį ir glikolio vandens srauto per šilumokaitį. Šios paklaidos taip pat atsispindėjo tiriant AANT veikimą atkuriant šių matavimų vertes, atjungus realų matavimą.

### 3.1.8. Apmokyto AANT tyrimo įvairiuose aušyklės režimuose rezultatai

Norint įsitikinti jau apmokyto AANT adekvatiškumu, iš kietu kuru kūrenamos kogeneracinės jėgainės, paskirstytos valdymo sistemos, istorinių proceso matavimo duomenų, atsitiktine tvarka buvo paimti duomenys iš įvairių aušyklės darbo režimų. 50 paveiksle parodytas atsitiktinių duomenų rinkinys tinklo įvestyje ir išvestyje. Iš grafikų matome, kad apmokytas tinklas sugeba atkartoti kiekvienos įvesties duomenis tinklo išvestyje. Tinklo tikslumui įvertinti toliau pateikiamos AANT apmokymo rezultatų santykinės procentinės paklaidos kiekviename apmokymo taške, lyginant duomenis tinklo įėjime su duomenimis tinklo išėjime.



45 pav. Atsitiktiniai matavimo duomenys AANT bandymui



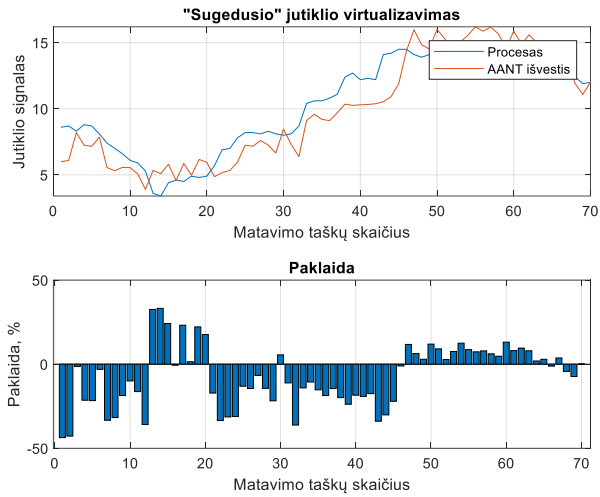
46 pav. AANT rezultatų santykinės procentinės paklaidos

3 lentelė. AANT rezultatų vidutinės absoliutinės procentinės paklaidos

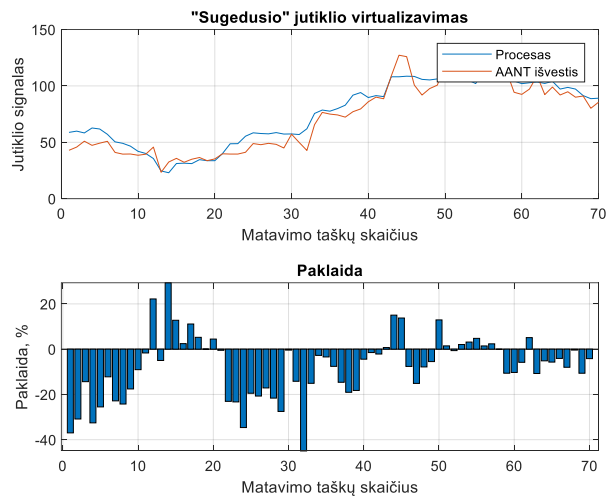
KKS kodas	Signalų pavadinimas	Vidutinė absoliutinė procentinė paklaida, % (MAPE)
10NDG20CQ901XQ01	Aušyklės momentinė šiluminė galia	10,0659 %
10NDG20CF901XQ01	Termofikacinio vandens srautas per šilumokaitį	12,1296 %
10NDA30CT901XQ01	Termofikacinio vandens temperatūra prieš šilumokaitį	0,0919 %
10NDG20CT001XQ01	Termofikacinio vandens temperatūra už šilumokaičio	0,7304 %
10PGD20CF901XQ01	Glikolio vandens srautas per šilumokaitį	5,3954 %
10PGD50CT002XQ01	Glikolio vandens temperatūra už aušyklės	0,54 %
10PGD30CT002XQ01	Glikolio vandens temperatūra prieš aušyklę	4,7978 %

Iš gautų rezultatų matyti, kad tiriant AANT su atsitiktiniais jėgainės jutiklių matavimo duomenimis, gaunamos paklaidos yra didesnės. Taip yra todėl, kad naudojamuose duomenyse yra matavimo taškų, kuriuose tinklas nėra tiksliai apmokytas. Kaip ir pirmuoju atveju AANT toliau bandomas, tiriant „sugedusio“ jutiklio atkūrimą – virtualizavimą. Jutiklio gedimas imituojamas jį atjungiant. Grafikuose pateikiamas kiekvienos tinklo išvesties atkūrimas, lyginant išvestį su jėgainės matavimo duomenimis (procesas). Šalia pateikiamos gautų rezultatų santykinės procentinės paklaidos kiekviename matavimo taške.

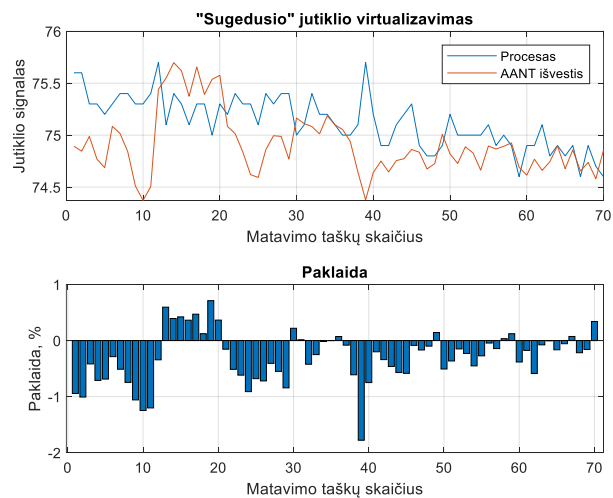




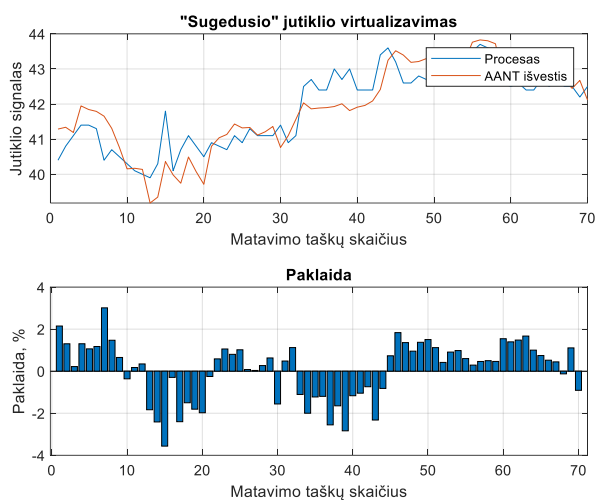
47 pav. „Sugedusio“ jutiklio Nr. 1, 10NDG20CQ901XQ01, MW – aušyklės momentinė šiluminė galia, vertės atkūrimas – virtualizavimas



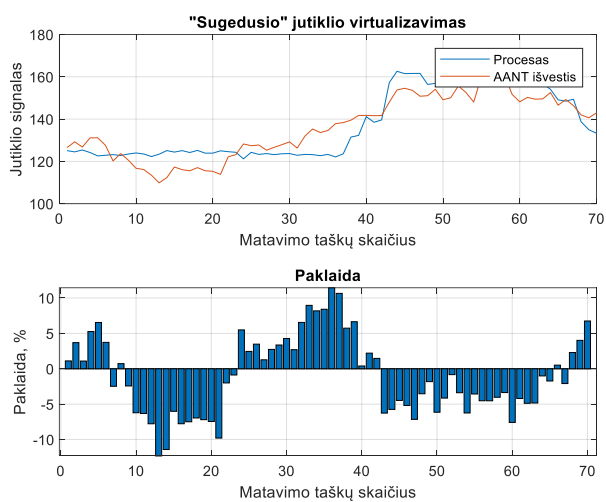
48 pav. „Sugedusio“ jutiklio Nr. 2, 10NDG20CF901XQ01, kg/s – termofikacinio vandens srautas per šilumokaitį, vertės atkūrimas – virtualizavimas



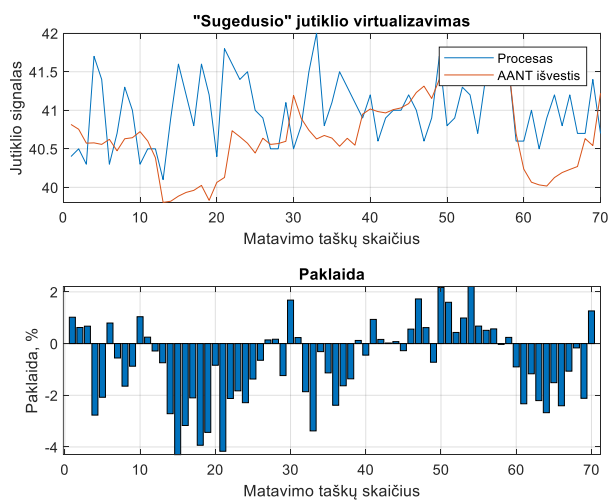
49 pav. „Sugedusio“ jutiklio Nr. 3, 10NDA30CT901XQ01, °C – termofikacinio vandens temperatūra prieš šilumokaitį, vertės atkūrimas – virtualizavimas



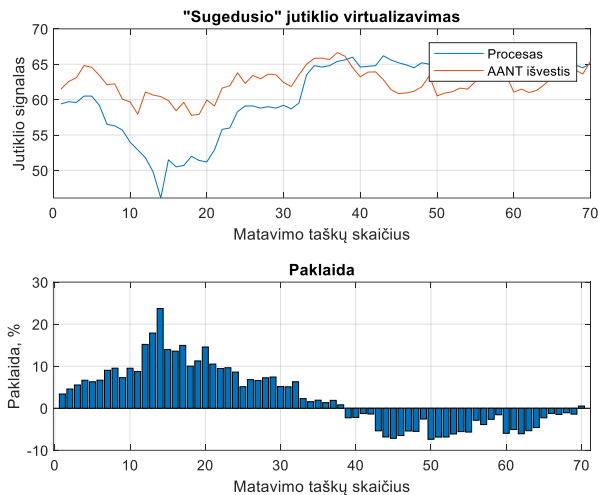
50 pav. „Sugedusio“ jutiklio Nr. 4, 10NDG20CT001XQ01, °C – termofikacinio vandens temperatūra už šilumokaičio, vertės atkūrimas – virtualizavimas



51 pav. „Sugedusio“ jutiklio Nr. 5, 10PGD20CF901XQ01, kg/s – glikolio vandens srautas per šilumokaitį, vertės atkūrimas – virtualizavimas



52 pav. „Sugedusio“ jutiklio Nr. 6, 10PGD50CT002XQ01, °C – glikolio vandens temperatūra už aušyklės, vertės atkūrimas – virtualizavimas



**53 pav.** „Sugedusio“ jutiklio Nr. 7, 10PGD30CT002XQ01, °C – glikolio vandens temperatūra prieš aušyklę, vertės atkūrimas – virtualizavimas

**4 lentelė.** AANT virtualizuotų matavimų, vidutinės absoliutinės procentinės paklaidos

KKS kodas	Signalų pavadinimas	Vidutinė absoliutinė procentinė paklaida, % (MAPE)
10NDG20CQ901XQ01	Aušyklės momentinė šiluminė galia	15,5677 %
10NDG20CF901XQ01	Termofikacinio vandens srautas per šilumokaitį	11,8854 %
10NDA30CT901XQ01	Termofikacinio vandens temperatūra prieš šilumokaitį	0,4206 %
10NDG20CT001XQ01	Termofikacinio vandens temperatūra už šilumokaitį	1,1433 %
10PGD20CF901XQ01	Glikolio vandens srautas per šilumokaitį	4,8259 %
10PGD50CT002XQ01	Glikolio vandens temperatūra už aušyklę	1,3527 %
10PGD30CT002XQ01	Glikolio vandens temperatūra prieš aušyklę	6,2975 %

Taškuose, kuriuose tinklo apmokymo metu buvo gautos didesnės paklaidos (aušyklės momentinė šiluminė galia, termofikacinio vandens srautas per šilumokaitį, glikolio vandens srautas per šilumokaitį, glikolio vandens temperatūra prieš aušyklę) atsispindėjo ir vykdant matavimo virtualizavimą.

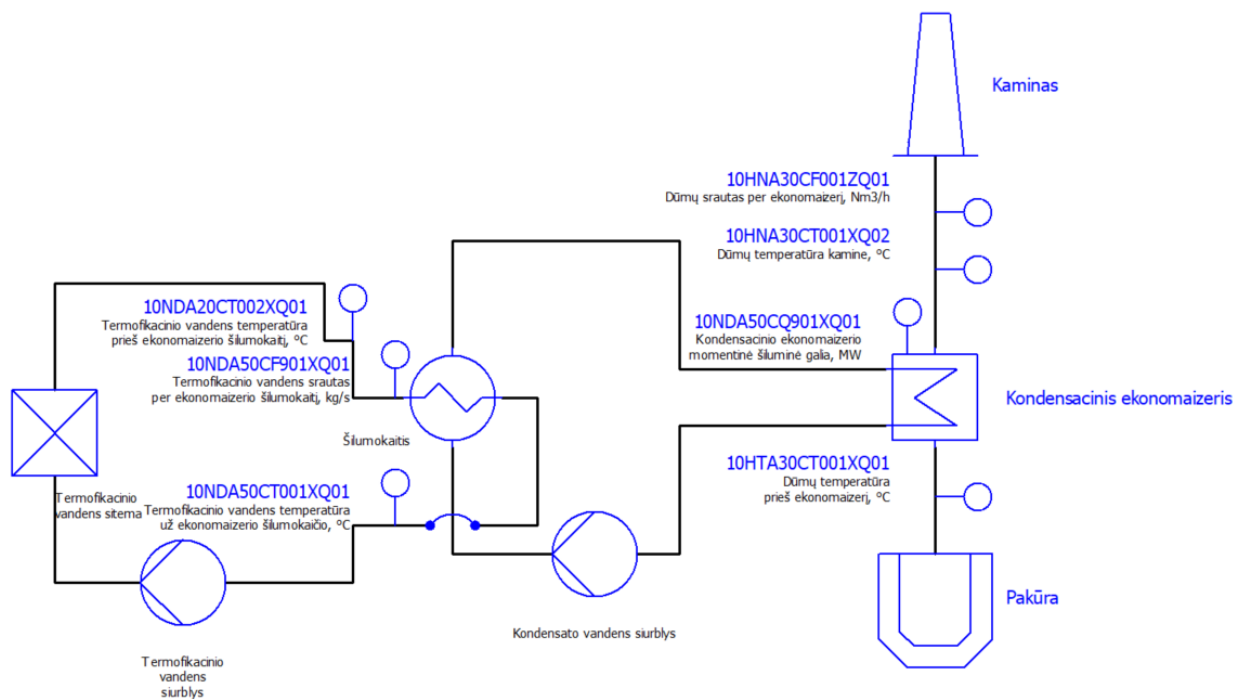
### 3.2. Kietu kuru kūrenamos kogeneracinės jėgainės, dūmų kondensacinio ekonomizerio AANT tyrimas

Siekiant išsiaiškinti ar autoasociatyvinis neuroninis tinklas gali būti taikomas ir kituose kietu kuru kūrenamos kogeneracines jėgaines procesuose, buvo pasirinkta jėgainės kondensacinio ekonomizerio sistema.

#### 3.2.1. Kogeneracinės jėgainės, dūmų kondensacinio ekonomizerio veikimo principas

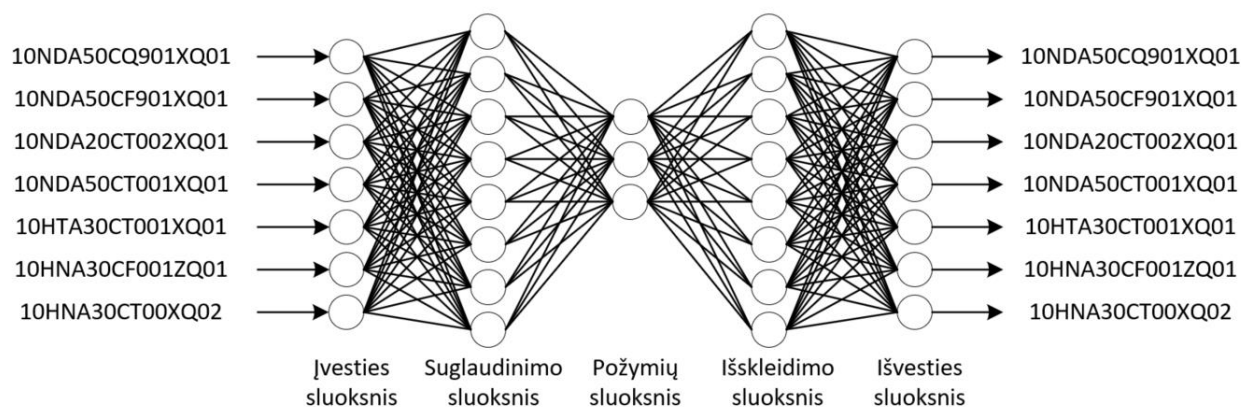
Kogeneracinėje jėgainėje yra sumontuotas kondensacinis dūmų ekonomizeris skirtas nuimti dalį šilumos nuo iš katilo išmetamų dūmų, tokiu būdu pašildant termofikacinį vandenį ir tuo pačiu padidinant jėgainės naudingumo koeficientą. Kadangi katilo išmetami dūmai yra drėgni jie paduodami į kondensacinį ekonomizerį, jame kondensuojasi ir susidariusi šiluma per ekonomizerio šilumokaitį atiduodama termofikaciniam vandeniui. Jėgainės valdymo sistemos, kondensacinio ekonomizerio parametrų, matavimų duomenys bus naudojami apmokyti ir testuoti AANT. 59

paveiksle parodyta supaprastinta kondensacinio ekonomaizerio vamzdynų ir prietaisų schema, kurioje proceso įranga kartu su prietaisų ir valdymo įtaisais.



54 pav. Kondensacinio ekonomaizerio P&ID schema

Parenkama septynių įvesčių, trijų paslėptų sluoksnių ir septynių išvesčių AANT architektūra. Suglaudavimo sluoksnyje parenkami aštuoni neuronai, požymių sluoksnį sudaro trys neuronai, o išskleidimo sluoksnyje modeliuojami aštuoni neuronai. Toliau pavaizduota kondensacinio ekonomaizerio struktūrinė AANT architektūros schema (žr. 60 pav.).



55 pav. Kondensacinio ekonomaizerio AANT modelis ir architektūra

Kogeneracinės jėgainės kondensacinio ekonomaizerio AANT įvestys (angl. *inputs*):

1. kondensacinio ekonomaizerio momentinė šiluminė galia: 10NDA50CQ901XQ01, MW.
2. termofikacinio vandens srautas per ekonomaizerio šilumokaitį: 10NDA50CF901XQ01, kg/s.
3. termofikacinio vandens temperatūra prieš ekonomaizerio šilumokaitį: 10NDA20CT002XQ01, °C.
4. termofikacinio vandens temperatūra už ekonomaizerio šilumokaičio: 10NDA50CT001XQ01, °C.

5. dūmų temperatūra prieš kondensacinį ekonomaizerį: 10HTA30CT001XQ01, °C.
6. dūmų srautas per kondensacinį ekonomaizerį: 10HNA30CF001ZQ01, Nm<sup>3</sup>/h.
7. dūmų temperatūra kamine: 10HNA30CT001XQ02, °C.

Kogeneracinės jėgainės kondensacinio ekonomaizerio AANT išvestys (angl. *outputs*):

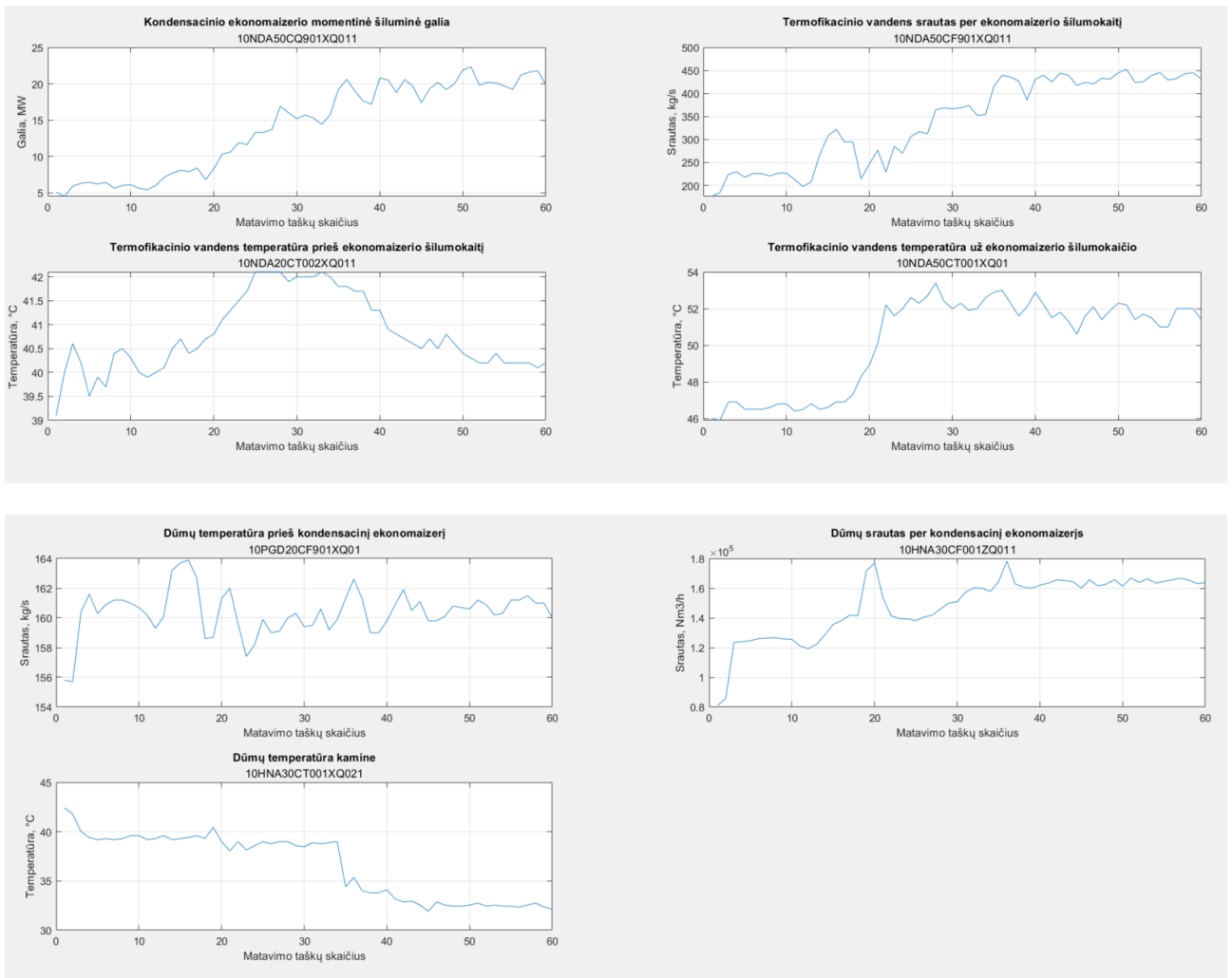
1. kondensacinio ekonomaizerio momentinė šiluminė galia: 10NDA50CQ901XQ01, MW.
2. termofikacinio vandens srautas per ekonomaizerio šilumokaitį: 10NDA50CF901XQ01, kg/s.
3. termofikacinio vandens temperatūra prieš ekonomaizerio šilumokaitį: 10NDA20CT002XQ01, °C.
4. termofikacinio vandens temperatūra už ekonomaizerio šilumokaičio: 10NDA50CT001XQ01, °C.
5. dūmų temperatūra prieš kondensacinį ekonomaizerį: 10HTA30CT001XQ01, °C.
6. dūmų srautas per kondensacinį ekonomaizerį: 10HNA30CF001ZQ01, Nm<sup>3</sup>/h.
7. dūmų temperatūra kamine: 10HNA30CT001XQ02, °C.

### 3.2.2. Kondensacinio ekonomaizerio matavimo duomenys

Iš kogeneracinės jėgainės, paskirstytos valdymo sistemos, istorinių proceso matavimo duomenų buvo paimti 200 matavimo taškai. Atsitiktine tvarka buvo atrinkta 60 matavimo taškų, o likusieji matavimo taškai buvo naudojami kaip apmokymo rinkinys. 61 paveiksle parodytas apmokymo rinkinys, o 62 paveiksle parodytas testavimo duomenų rinkinys.



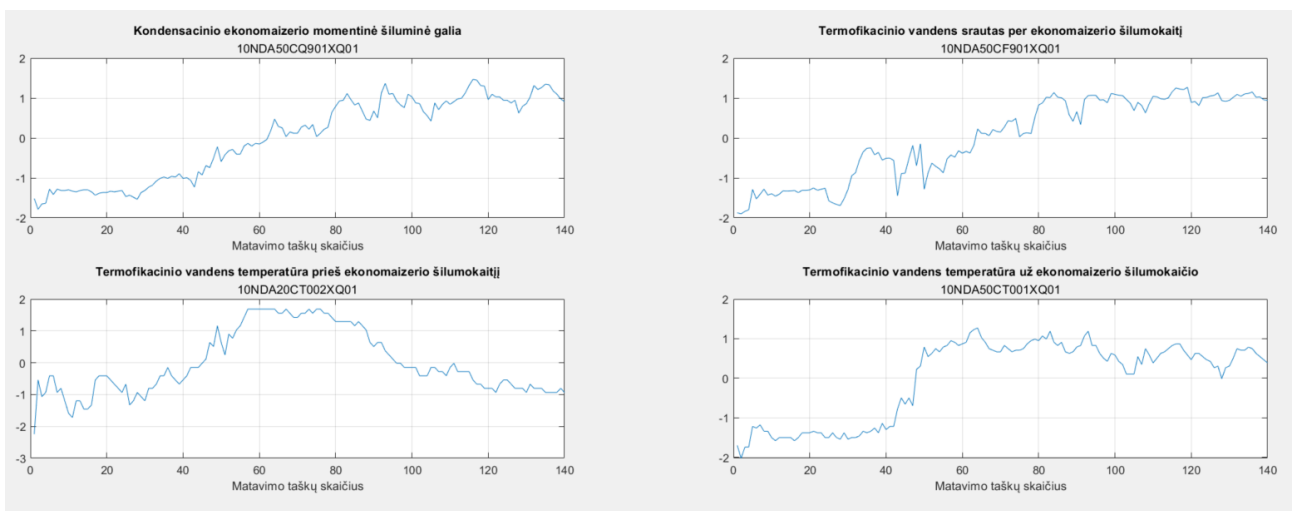
56 pav. AANT apmokymo duomenų rinkinys

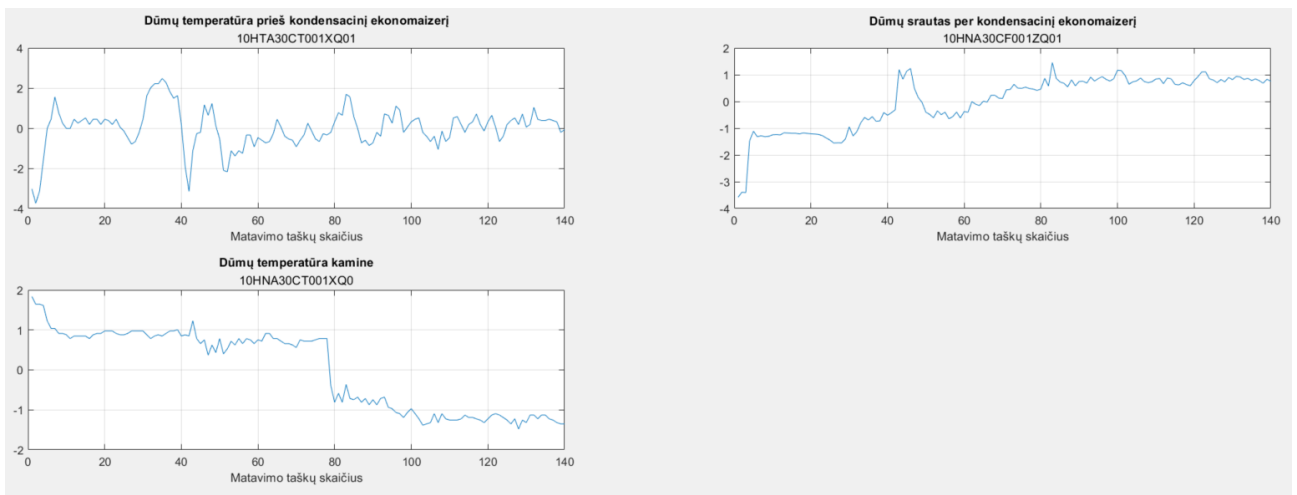


57 pav. AANT testavimo duomenų rinkinys

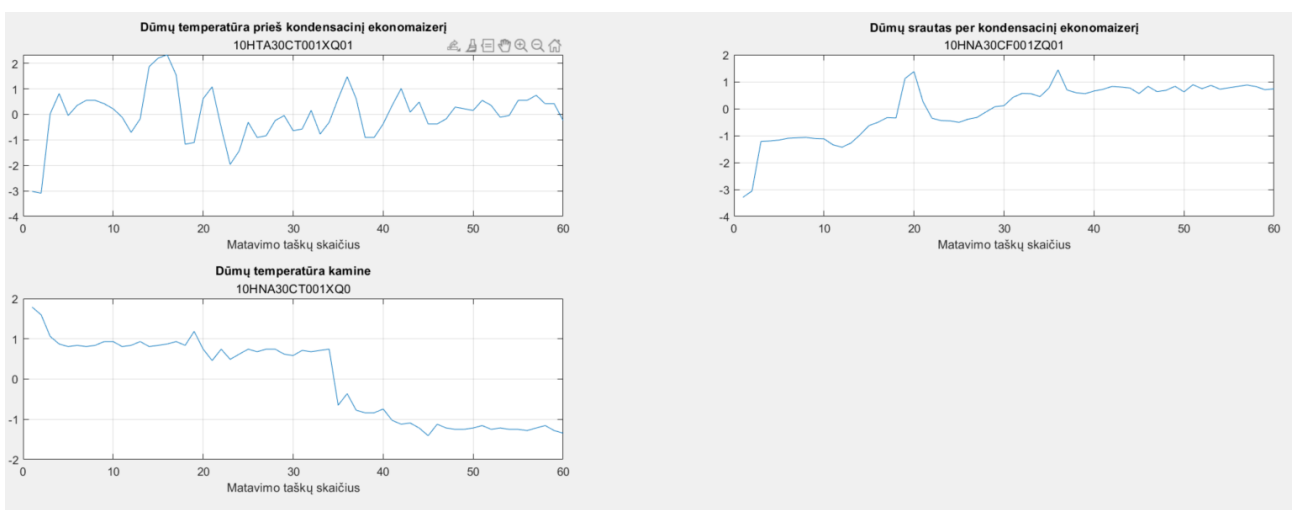
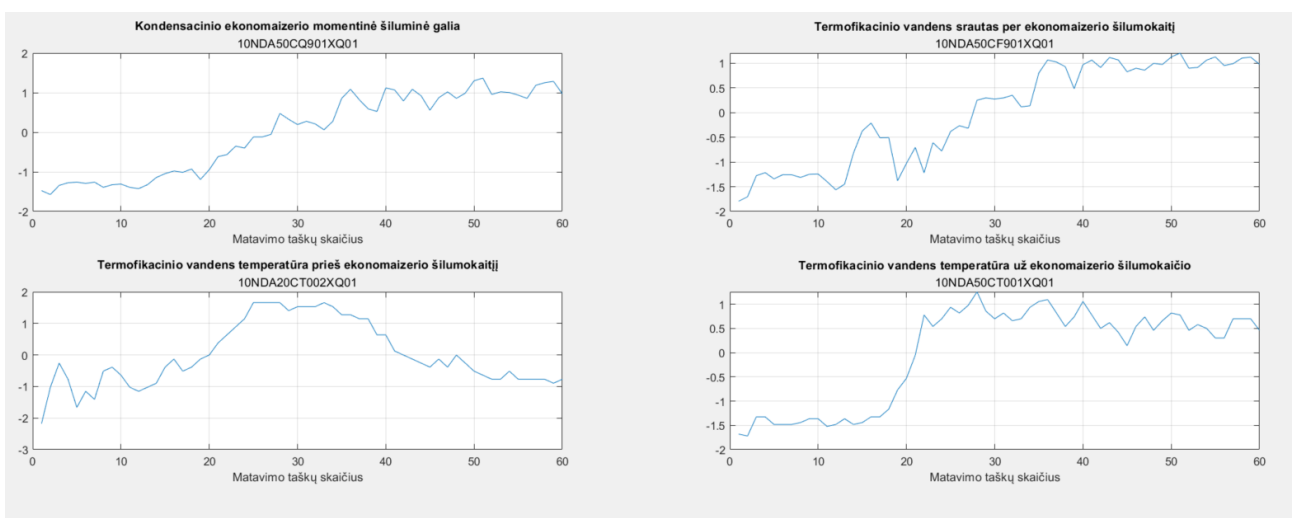
### 3.2.3. Matavimo duomenų normalizavimas

Kaip ir prieš tai atliktų bandymų metu, taip ir šiuo atveju, AANT apmokymui skirti duomenys yra normalizuojami. Toliau pateikiami normalizuotų apmokymo duomenų (žr. 63 pav.) ir normalizuotų AANT testavimo duomenų (žr. pav. 64) rinkiniai.





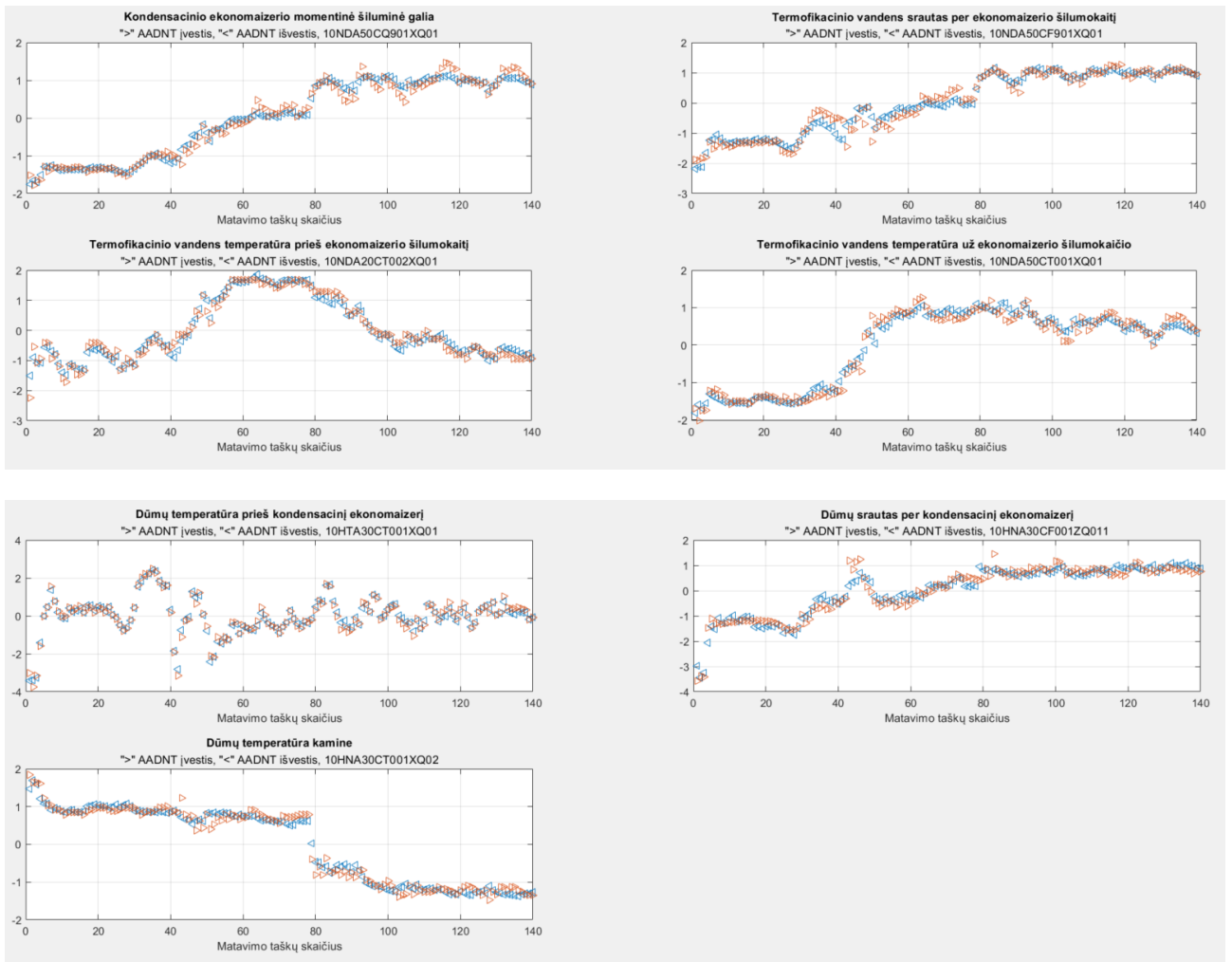
58 pav. Normalizuoti AANT apmokymo duomenų rinkinys



59 pav. Normalizuoti AANT testavimo duomenų rinkinys

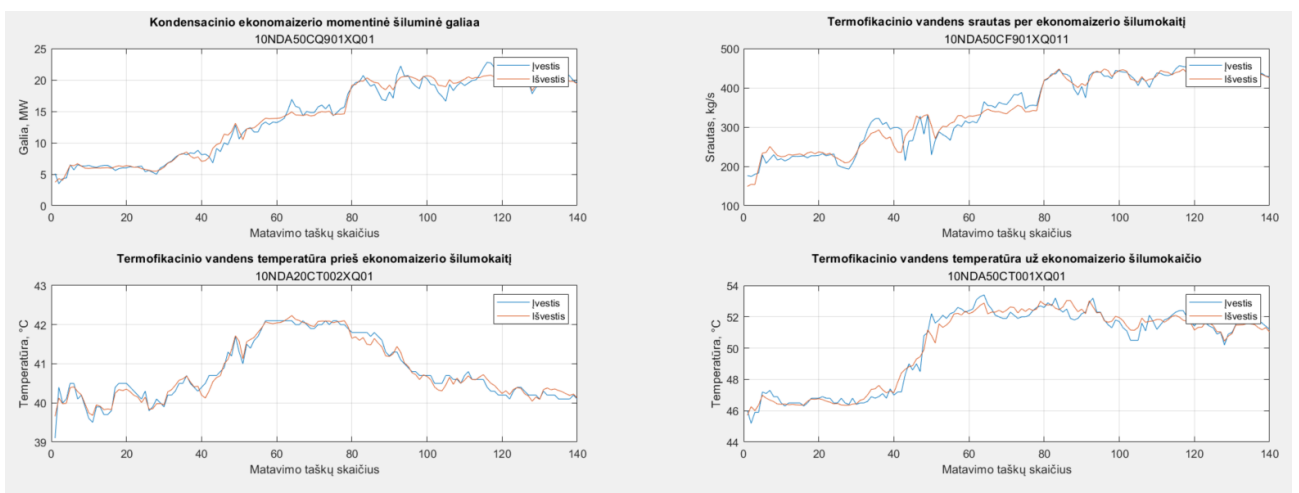
### 3.2.4. Apmokyto AANT rezultatai

Parinktos AANT architektūros (7-8-3-8-7) apmokymas su pateiktu kondensacinio ekonomizerio matavimo duomenų rinkiniu vyksta 1429 sekundes. Toliau pateikiami normalizuoti AANT apmokymo duomenų rezultatai (žr. 65 pav.).

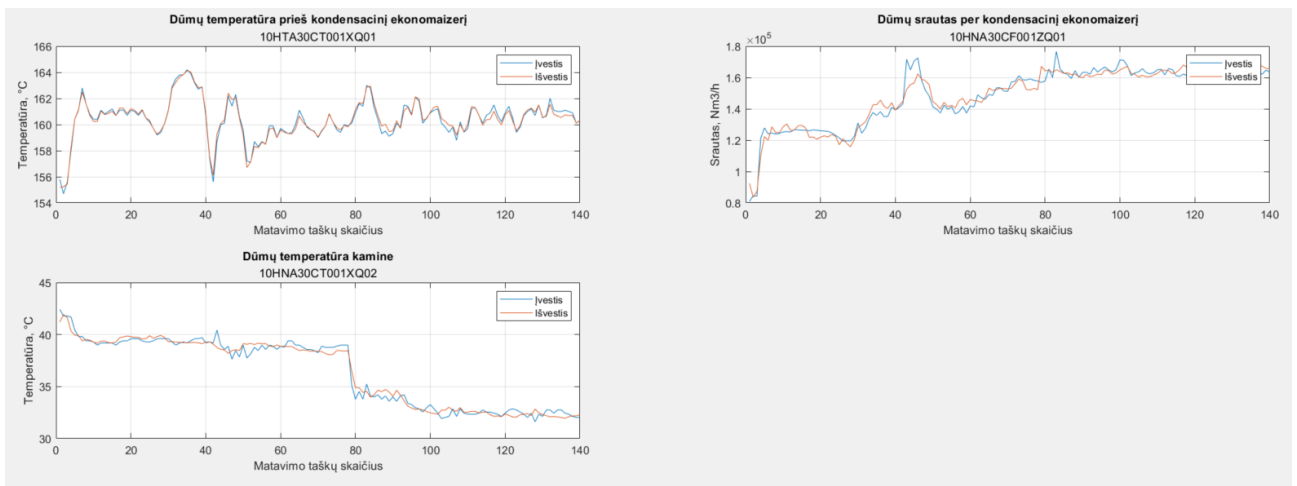


**60 pav.** Apmokyto AANT normalizuoti apmokymo duomenys tinklo įvestyje („>“) ir išvestyje („<“)

Toliau pateikiami AANT apmokymo rezultatų grafikai fizikiniais dydžiais, tinklo įvestis ir tinklo išvestis.

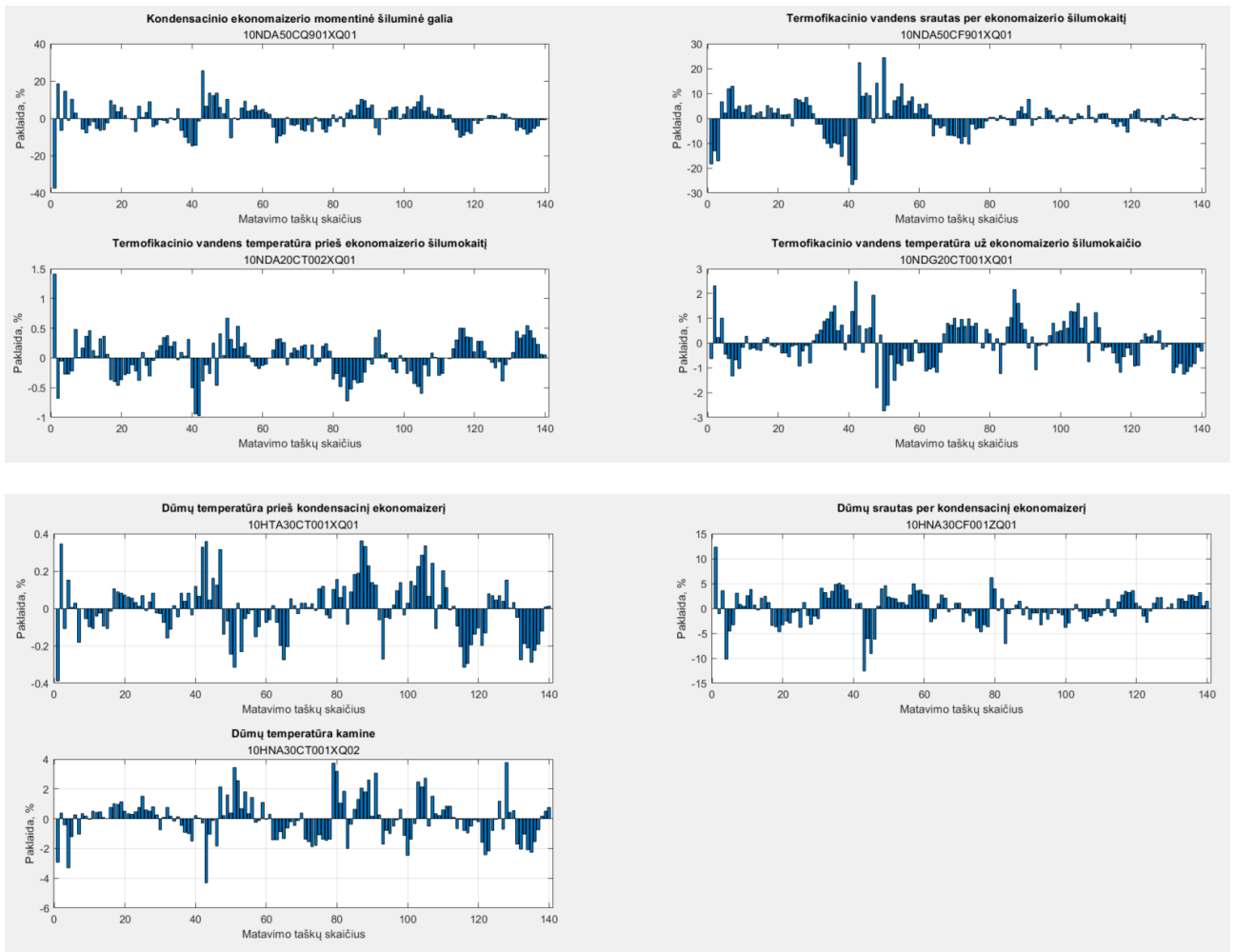






61 pav. AANT apmokymo rezultatai fizikiniais dydžiais

Tinklo tikslumui įvertinti, apskaičiuojamos AANT apmokymo rezultatų santykinės procentinės paklaidos kiekviename matavimo taške, lyginant duomenis tinklo įėjime su duomenimis tinklo išėjime.



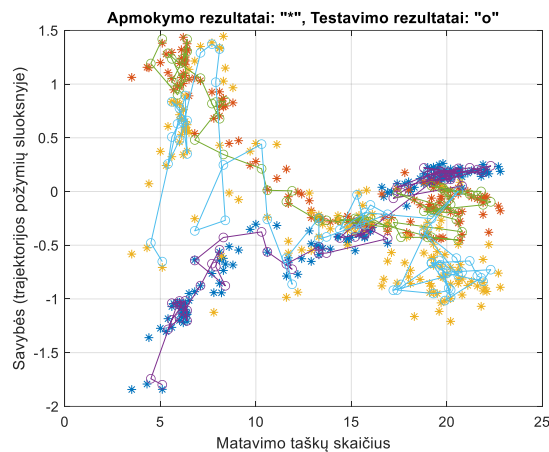
62 pav. AANT rezultatų santykinės procentinės paklaidos

5 lentelė. AANT rezultatų vidutinės absoliutinės procentinės paklaidos

KKS kodas	Signalų pavadinimas	Vidutinė absoliutinė procentinė paklaida, % (MAPE)
10NDA50CQ901XQ01	Kondensacinio ekonomaižerio momentinė šiluminė galia	5,4255 %
10NDA50CF901XQ01	Termofikacinio vandens srautas per ekonomaižerio šilumokaitį	4,7985 %
10NDA20CT002XQ01	Termofikacinio vandens temperatūra prieš ekonomaižerio šilumokaitį	0,2594 %
10NDG20CT001XQ01	Termofikacinio vandens temperatūra už ekonomaižerio šilumokaičio	0,6782 %
10HTA30CT001XQ01	Dūmų temperatūra prieš kondensacinį ekonomaižerį	0,1143 %
10HNA30CF001ZQ01	Dūmų srautas per kondensacinį ekonomaižerį	2,4014 %
10HNA30CT001XQ02	Dūmų temperatūra kamine	1,046 %

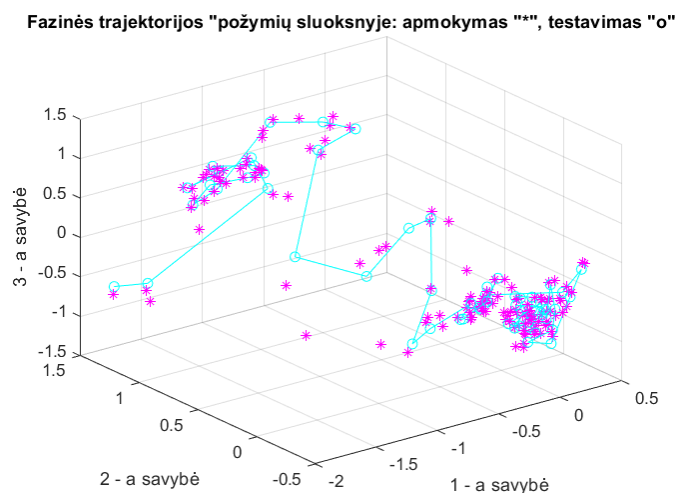
Vidutinė absoliutinė procentinė paklaida neviršija 10 %, todėl galime daryti išvadą, kad AANT su pateiktais matavimo duomenimis yra apmokytas labai tiksliai [31].

Apmokymo duomenų savybės – trajektorijos požymių sluoksnyje:



63 pav. AANT apmokymo duomenų savybės („\*“ – kiekvieno kintamojo apmokymo rezultatai, „o“ – kiekvieno kintamojo testavimo rezultatai)

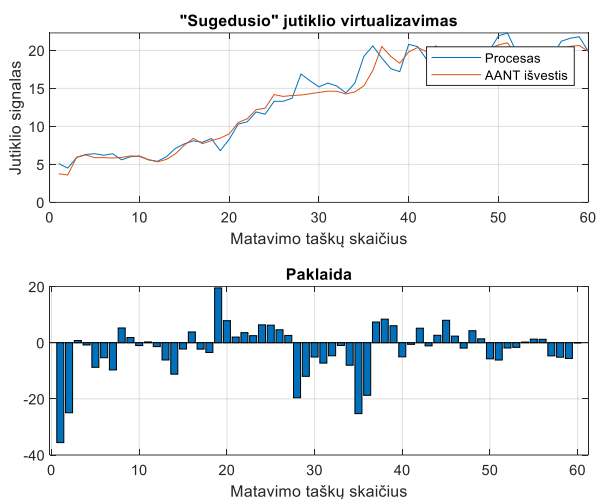
Apmokymo duomenų fazinės trajektorijos požymių sluoksnyje:



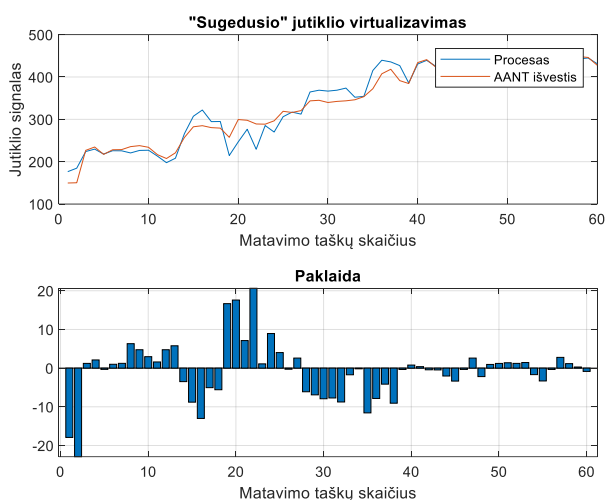
64 pav. AANT fazinės trajektorijos požymių sluoksnyje („\*“ – apmokymas, „o“ – testavimas)

### 3.2.5. „Sugedusio“ jutiklio virtualizavimo naudojant apmokytą AANT rezultatai

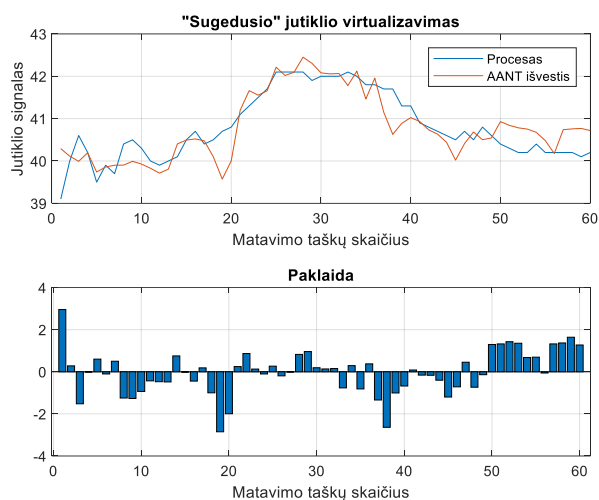
Apmokytas AANT išbandomas, tiriant „sugedusio“ jutiklio atkūrimą – virtualizavimą. Kaip ir prieš tai atliktais bandymais, jutiklio gedimas imituojamas jį atjungiant. Grafikuose pateikiamas kiekvienos tinklo išvesties atkūrimas, lyginant išvestį su jėgainės matavimo duomenimis (procesas). Šalia pateikiamos gautų rezultatų santykinės procentinės paklaidos kiekviename matavimo taške.



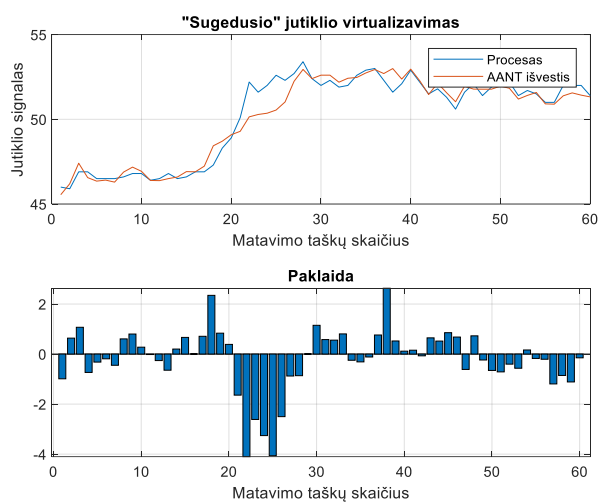
65 pav. „Sugedusio“ jutiklio Nr. 1, 10NDA50CQ901XQ01, MW – kondensacinio ekonomaizerio momentinė šiluminė galia, vertės atkūrimas – virtualizavimas



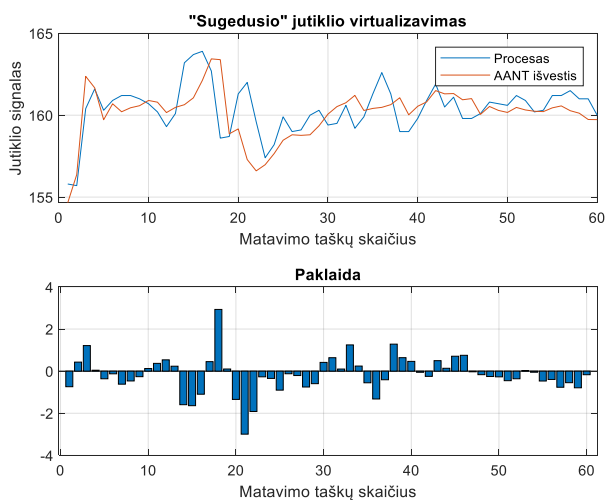
66 pav. „Sugedusio“ jutiklio Nr. 2, 10NDA50CF901XQ01, kg/s – termofikacinio vandens srautas per ekonomaizerio šilumokaitį, vertės atkūrimas – virtualizavimas



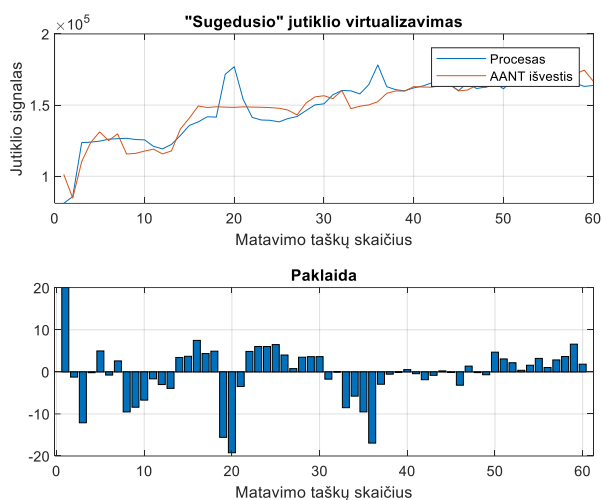
67 pav. „Sugedusio“ jutiklio Nr. 3, 10NDA20CT002XQ01, °C – termofikacinio vandens temperatūra prieš ekonomizaizerio šilumokaitį, vertės atkūrimas – virtualizavimas



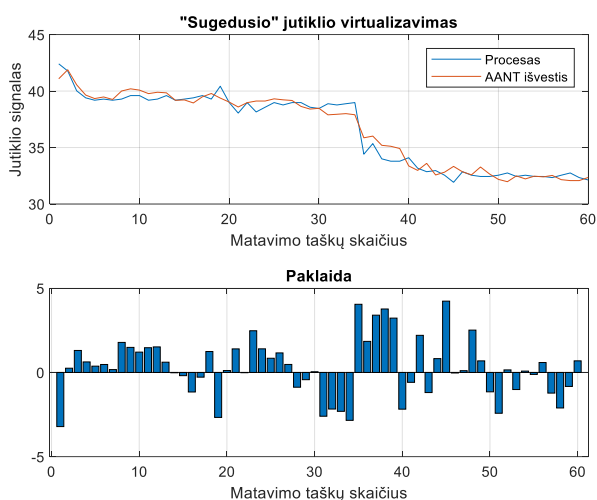
68 pav. „Sugedusio“ jutiklio Nr. 4, 10NDA50CT001XQ01, °C – termofikacinio vandens temperatūra už ekonomizaizerio šilumokaitio, vertės atkūrimas – virtualizavimas



69 pav. „Sugedusio“ jutiklio Nr. 5, 10HTA30CT001XQ01, °C – dūmų temperatūra prieš kondensacinį ekonomizaizerį, vertės atkūrimas – virtualizavimas



**70 pav.** „Sugedusio“ jutiklio Nr. 6, 10HNA30CF001ZQ01, Nm<sup>3</sup>/h – dūmų srautas per kondensacinį ekonomaizerį, vertės atkūrimas – virtualizavimas



**71 pav.** „Sugedusio“ jutiklio Nr. 7, 10HNA30CT001XQ02, °C – dūmų temperatūra kamine, vertės atkūrimas – virtualizavimas

**6 lentelė.** AANT virtualizuotų matavimų, vidutinės absoliutinės procentinės paklaidos

KKS kodas	Signalų pavadinimas	Vidutinė absoliutinė procentinė paklaida, % (MAPE)
10NDA50CQ901XQ01	Kondensacinio ekonomaizerio momentinė šiluminė galia	6,1625 %
10NDA50CF901XQ01	Termofikacinio vandens srautas per ekonomaizerio šilumokaitį	4,8213 %
10NDA20CT002XQ01	Termofikacinio vandens temperatūra prieš ekonomaizerio šilumokaitį	0,7755 %
10NDG20CT001XQ01	Termofikacinio vandens temperatūra už ekonomaizerio šilumokaičio	0,8429 %
10HTA30CT001XQ01	Dūmų temperatūra prieš kondensacinį ekonomaizerį	0,6208 %
10HNA30CF001ZQ01	Dūmų srautas per kondensacinį ekonomaizerį	4,3783 %
10HNA30CT001XQ02	Dūmų temperatūra kamine	1,3403 %

Iš gautų rezultatų ir vidutinių absoliutinių procentinių paklaidų matome, kad tinklas atkuria sugedusį jutiklį, o atkūrimo paklaidos neviršija 10 %, kas rodo, kad jutikliai virtualizuojami labai tiksliai.

## Rezultatai ir išvados

1. Sumodeliuotas autoasociatyvinis neuroninis tinklas apmokomas Levenbergo – Marquardo algoritmu. Apmokymo algoritmo procedūra kartojama tol, kol išėjimo vektorius tampa pakankamai panašus į įėjimo vektorius (užduotos paklaidos ribose), arba kol pasiekiamas nustatytas iteracijų skaičius. Darbe visais atvejais AANT tinklas apmokomas mažiau nei per 10 iteracijų.
2. Atliekant bandymus su įvairiais duomenimis pastebėta, kad tinklo paklaidos didėja srityse, kuriose tinklas nėra apmokytas.
3. Srityse, kuriose tinklas yra apmokytas, vidutinės absoliutinės procentinės paklaidos tarp apmokyto tinklo įvesčių ir išvesčių neviršija 10 %, kas leidžia daryti prielaidą, kad tinklas yra apmokytas labai tiksliai.
4. Atlikus tyrimą iš apskaičiuotų paklaidų rezultatų galima daryti išvadą, kad tokio tikslumo neuroninio tinklo pakaktų, nustatyti jutiklio gedimą, tačiau tikslumas būtų per mažas tinklo išvestį naudoti proceso valdyme, reguliavime.
5. Jutiklių virtualizavimo sistema sugeba atkurti kiekvienos tinklo įvesties jutiklio vertes, kurios gali būti naudojamos, kaip dubliuojanti sistema jutiklių matavimo monitoringui, jutiklių gedimo identifikavimui ar kaip pagalbinė priemonė operatyviam personalui, esant matavimo prietaiso gedimui ir neturint tiesioginio proceso matavimo prietaiso.
6. Įrodyta, kad AANT gali būti naudojamas jutiklių diagnostikoje, taip pat matavimo prietaisų parodymams virtualizuoti. Pateikti pavyzdžiai iliustruoja šios sistemos efektyvumą.
7. Esant dideliame duomenų diskretizavimo dažniui, kas 1 sekundę, AANT veikimas blogėja, todėl buvo pasirinktas mažesnis diskretizavimo dažnis, kas 30 minučių, tuo pačiu stengiantis neprarasti informacijos apie proceso dinamines savybes.
8. Pastebėta, kad esant dideliame duomenų diskretizavimo dažniui, kas 1 sekundę, apmokymo duomenyse, tuo pačiu į tinklą paduodamas triukšmas, dėl to atsiranda paklaidos atsiranda tinklo svoriuose, dėl ko tinklas pradeda dirbti netiksliai – didėja paklaidos tarp tinklo įvesties ir išvesties.
9. Siekiant patikrinti, ar AANT gali būti realizuotas įvairiuose jėgainės sistemose, buvo atliekamas tinklo tyrimas kitoje kogeneracinės jėgainės sistemoje. Tinklas pritaikytas jėgainės kondensacinio ekonomizerio sistemai.

## Literatūros sąrašas

1. KRAMER, M.A. Autoassociative Neural Networks. *Computers & Chemical Engineering* [Interaktyvus]. Laboratory for Intelligent Systems in Process Engineering, Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A, 1992, **vol. 16** (4), 313-328 [žiūrėta 2021-05-09]. doi: 10.1016/0098-1354(92)80051-A.
2. KRAMER, M.A. Nonlinear Principal Component Analysis using Autoassociative Neural Networks. *AIChE Journal* [interaktyvus]. Laboratory for Intelligent Systems in Process Engineering, Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A, 1991, **vol. 37** (2), 233-243 [žiūrėta 2021-05-09]. doi: 10.1002/aic.690370209.
3. SHAJITH IKBAL M., MISRA H. and YEGNANARAYANA B. Analysis of Autoassociative Mapping Neural Networks. Iš: IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No.99CH36339), July 10-16, 1999, Washington, DC, USA, 1999. pp. 3025-3029.
4. HINES J. W., UHRIG R. E. and WREST D. J. Use of Autoassociative Neural Networks for Signal Validation. *Journal of Intelligent and Robotic Systems* [interaktyvus]. Kluwer Academic Publishers. Printed in the Netherlands, 1998, **vol. 21** (2), 143-154 [žiūrėta 2021-05-09].
5. HINES, J.W., WREST D. J. and UHRIG R. E. Plant Wide Sensor Calibration Monitoring. Iš: Proceedings of the 1996 IEEE International Symposium on Intelligent Control, 15-18 Sept. 1996, Dearborn, MI, USA, 1996. pp. 378-383.
6. XU X., HINES J. W. and UHRIG R. E. Sensor Validation and Fault Detection using Neural Networks. Iš: Proc. Maintenance and Reliability Conference (MARCON 99), 1999. pp. 10-12.
7. HEYDARZADEH M. and NOURANI M. A Two-Stage Fault Detection and Isolation Platform for Industrial Systems using Residual Evaluation. *IEEE Transactions on Instrumentation and Measurement* [interaktyvus]. Tehran, Iran, 2016, **vol. 65** (10), 2424-2432 [žiūrėta 2021-05-09]. doi: 10.1109/TIM.2016.2575179.
8. NAJAFABADI T. A., SALMASI F. R. and JABEHDAR-MARALANI P. Detection and Isolation of Speed-, DC-Link Voltage-, and Current-Sensor Faults Based on an Adaptive Observer in Induction-Motor Drives. *IEEE Transactions on Industrial Electronics* [interaktyvus]. Esfahan, Iran, 2011, **vol. 58** (5), 1662-1672 [žiūrėta 2021-05-09]. doi: 10.1109/TIE.2010.2055775.
9. SOTOMAYOR O. A. Z. and ODLOAK D. Observer-Based Fault Diagnosis in Chemical Plants. *Chemical Engineering Journal* [interaktyvus]. University of Sao Paulo, Department of Chemical Engineering, Av. Prof. Luciano Gualberto, trav. 3, n. 380, 05508-900 Sao Paulo SP, Brazil, 2005, **vol. 112** (1), 93-108 [žiūrėta 2021-05-09]. doi: 10.1016/j.cej.2005.07.001.
10. LI B., CHOW M., TIPSUVAN Y. and HUNG J. C. Neural-Network-Based Motor Rolling Bearing Fault Diagnosis. *IEEE Transactions on Industrial Electronics* [interaktyvus]. Beijing, China, 2000, **vol. 47** (5), 1060-1069 [žiūrėta 2021-05-09]. doi: 10.1109/41.873214.
11. GAO X. Z. and OVASKA S. J. Soft Computing Methods in Motor Fault Diagnosis. *Applied Soft Computing* [interaktyvus]. Institute of Intelligent Power Electronics, Helsinki University of Technology, Otakaari 5 A, FIN-02150 Espoo, Finland, 2001, **vol. 1** (1), 73-81 [žiūrėta 2021-05-09]. doi: 10.1016/S1568-4946(01)00008-4.

12. BENBOUZID M. E. H. and NEJJARI H. A Simple Fuzzy Logic Approach for Induction Motors Stator Condition Monitoring. Iš: IEMDC 2001. IEEE International Electric Machines and Drives Conference (Cat. No.01EX485), August 07, 2002, Cambridge, MA, USA, 2001. pp. 634-639.
13. VENKATASUBRAMANIAN V., RENGASWAMY R. and KAVURI SURYA N. A Review of Process Fault Detection and Diagnosis: Part II: Qualitative Models and Search Strategies. *Computers & Chemical Engineering* [interaktyvus]. Laboratory for Intelligent Process Systems, School of Chemical Engineering, Purdue University, West Lafayette, IN 47907, USA, Department of Chemical Engineering, Clarkson University, Potsdam, NY 13699-5705, USA, BP, Houston, TX, USA, 2003, **vol. 27** (3), 313-326 [žiūrėta 2021-05-09]. doi: 10.1016/S0098-1354(02)00161-8.
14. VENKATASUBRAMANIAN, V., RENGASWAMY, R., YIN K. and KAVURI S. N. A Review of Process Fault Detection and Diagnosis: Part I: Quantitative Model-Based Methods. *Computers & Chemical Engineering* [interaktyvus]. Laboratory for Intelligent Process Systems, School of Chemical Engineering, Purdue University, West Lafayette, IN 47907, USA, Department of Chemical Engineering, Clarkson University, Potsdam, NY 13699-5705, USA, Department of Wood and Paper Science, University of Minnesota, St. Paul, MN 55108, USA, BP, Houston, TX, USA, 2003, **vol. 27** (3), 293-311 [žiūrėta 2021-05-09]. doi: 10.1016/S0098-1354(02)00160-6.
15. VENKATASUBRAMANIAN V., RENGASWAMY R., KAVURI SURYA N. and YIN K. A Review of Process Fault Detection and Diagnosis: Part III: Process History Based Methods. *Computers & Chemical Engineering* [interaktyvus]. Laboratory for Intelligent Process Systems, School of Chemical Engineering, Purdue University, West Lafayette, IN 47907, USA, Department of Chemical Engineering, Clarkson University, Potsdam, NY 13699-5705, USA, BP, Houston, TX, USA, Department of Wood and Paper Science, University of Minnesota, St. Paul, MN 55108, USA, 2003, **vol. 27** (3), 327-346 [žiūrėta 2021-05-09]. doi: 10.1016/S0098-1354(02)00162-X.
16. TAN WOEI WAN and HUO HONG. An on-Line Neurofuzzy Approach for Detecting Faults in Induction Motors. Iš: IEMDC 2001. IEEE International Electric Machines and Drives Conference (Cat. No.01EX485), 17-20 June 2001, Cambridge, MA, USA, 2001. pp. 878-883.
17. ROSSI T. M. and BRAUN J. E. A Statistical, Rule-Based Fault Detection and Diagnostic Method for Vapor Compression Air Conditioners. *Hvac&R Research* [interaktyvus]. National Institute of Standards and Technology, Johnson Controls, Inc., and Purdue University, 1997, **vol. 3** (1), 19-37 [žiūrėta 2021-05-09]. doi: 10.1080/10789669.1997.10391359.
18. WANG S. and WANG J. Law-Based Sensor Fault Diagnosis and Validation for Building Air-Conditioning Systems. *HVAC&R Research* [interaktyvus]. University Research Grant, 1999, **vol. 5** (4), 353-380 [žiūrėta 2021-05-09]. doi: 10.1080/10789669.1999.10391243.
19. M. Elnour, N. Meskin and M. Al-Naemi. Sensor Fault Diagnosis of Multi-Zone HVAC Systems using Auto-Associative Neural Network. Iš: 2019 IEEE Conference on Control Technology and Applications (CCTA), 19-21 Aug. 2019, Hong Kong, China, 2019. pp. 118-123.
20. KAVAZ A. G. and BARUTCU B. Fault Detection of Wind Turbine Sensors using Artificial Neural Networks. *Journal of Sensors* [interaktyvus]. Energy Institute, Istanbul Technical University, 34469 Istanbul, Turkey, 2018, **vol. 2018**, 5628429 [žiūrėta 2021-05-09]. doi: 10.1155/2018/5628429.
21. SHAHERYAR A., et al. A Denoising Based Autoassociative Model for Robust Sensor Monitoring in Nuclear Power Plants. *Science and Technology of Nuclear Installations*



[interaktyvus]. University of Science and Technology Beijing (USTB), 30 Xueyuan Road, Haidian District, Beijing 100083, China, Institute of Automation, Chinese Academy of Sciences, University of Science and Technology Beijing, 95 Zhongguancun East Road, Beijing 100190, China, Department of Electrical Engineering, COMSATS Institute of Information Technology, University Road, Abbottabad 22060, Pakistan, COMSATS Institute of Information Technology near Officers Colony, Kamra Road, Attock 43600, Pakistan, 2016, **vol. 2016** [žiūrėta 2021-05-09]. doi: 10.1155/2016/9746948.

22. FROMAIGÉAT L. and NICCHIOTTI G. Case Study: Vibration Trip and Post-Event Analysis with Auto-Associative Neural Networks on a Large Steam Turbine. Iš: PHM Society European Conference, July 2016, 2016.

23. LEMA T. A. and HASHIM F. M. Wavelet Analysis and Auto-Associative Neural Network Based Fault Detection and Diagnosis in an Industrial Gas Turbine. Iš: 2012 IEEE Business, Engineering Industrial Applications Colloquium (BEIAC), 7-8 April 2012, Kuala Lumpur, Malaysia, 2012. pp. 103-108.

24. PALMÉ, T., FAST M. and THERN M. Gas Turbine Sensor Validation through Classification with Artificial Neural Networks. *Applied Energy* [interaktyvus]. Department of Mechanical and Structural Engineering and Materials Science, University of Stavanger, N-4036 Stavanger, Norway, Department of Energy Sciences, Lund University, P.O. Box 118, SE 22100, Sweden, 2011, **vol. 88** (11), 3898-3904 [žiūrėta 2021-05-09]. doi: 10.1016/j.apenergy.2011.03.047.

25. SHAH B., SARVAJITH M., SANKAR B. and THENNAVARAJAN S. Multi-Auto Associative Neural Network Based Sensor Validation and Estimation for Aero-Engine. Iš: 2013 IEEE AUTOTESTCON, 16-19 Sept. 2013, Schaumburg, IL, USA, 2013. pp. 1-7.

26. WILAMOWSKI B. M. Neural Network Architectures and Learning. Iš: IEEE International Conference on Industrial Technology, 2003, 10-12 Dec. 2003, Maribor, Slovenia, 2003. pp. TU1-T12.

27. LEVENBERG K. A Method for the Solution of Certain Non-Linear Problems in Least Squares. *Quarterly of Applied Mathematics* [interaktyvus]. Frankford Arsenal, 1944, **vol. 2** (2), 164-168 [žiūrėta 2021-05-09]. doi: 10.1090/QAM/10666.

28. MARQUARDT D. W. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics* [interaktyvus]. U.S.A, 1963, **vol. 11** (2), 431-441 [žiūrėta 2021-05-09]. doi: 10.1137/0111030.

29. RANGANATHAN A. The Levenberg - Marquardt Algorithm. *Tutorial on LM Algorithm* [interaktyvus]. , 2004, **vol. 11** (1), 101-110 [žiūrėta 2021-05-09]. doi: 10.1.1.135.865&rep=rep1.

30. NORGAARD M. Nnsysid. [žiūrėta 2021-05-09]. Prieiga per: <https://www.mathworks.com/matlabcentral/fileexchange/87-nnsysid>.

31. STABINGIENĖ L. Prognozavimo Uždavinių Samprata. [žiūrėta 2021-05-09]. Prieiga per: [http://www.ilab.lt/stabingiene/sk8\\_5.html](http://www.ilab.lt/stabingiene/sk8_5.html).

## Priedai

### 1 priedas. AANT vykdymo programa MATLAB aplinkoje

```
% Jutikliu matavimo duomenų tyrimas naudojant AADNT %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Jėgaines ausykles, jutikliu matavimo duomenų tyrimas naudojant AADNT %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Naudojamos paprogramės: %
% DSCALE.m, MARQX2NN.m, NNEVALE4E.m, NNEVAL4ERR.m, NNEVAL4NN.m, PMNTANH.m %
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function AADNT_modelis_apmokymas

clc, clear all, close all, warning off

fprintf('    AADNT_modelis.m:\n');
fprintf('    AADNT taikymas termofikacinio\n');
fprintf('    vandens aušyklės jutiklių matavimų stebėjimui\n');
fprintf('    Skaičiavimai atlikti %s
%s\n', date, datestr(rem(now,1)), fprintf('\n'));

training_RMSE=1%e7; % Pradinė tikslumo vertė, RMSE - Root mean square Error
Ypseudo3=[]; % Masyvas rekursiniam atnaujinimui

% Ikeliami AADNT apmokymui skirti matavimų duomenys
% Visi duomenys

load('R1100.mat');

% Apmokymo duomenys

load('R1.mat');

% Testavimo duomenys

load('R2.mat');

% Duomenų bylu struktūra:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DH AUX COOL MW ,DH AUX COOL DH F,DH WATER TO NET T,DH AFTER AUX COOL T,%
% 10NDG20CQ901XQ01,10NDG20CF901XQ01,10NDA30CT901XQ01 ,10NDG20CT001XQ01 ,%
% 1, 2, 3, 4, %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DH AUX CW F ,DH AUX CW TO PUMP T,DH AUX CW TO AIR COOL T,%
% 10PGD20CF901XQ01,10PGD50CT002XQ01 ,10PGD30CT002XQ01 ,%
% 5, 6, 7, %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Matavimo duomenys AADNT apmokymui, duomenų rinkiniai:

T_AUX_cooler=[R1]';

% Matavimo duomenys AADNT testavimui:

V_AUX_cooler=[R2]';

% Matavimo duomenų normavimas (skaliavimas):

[T_AUX_cooler_sc,sclt]=DSCALE(T_AUX_cooler);
[V_AUX_cooler_sc,sclv]=DSCALE(V_AUX_cooler);
```

```
% Bendras matavimo duomenų rinkinys 98 taskai (fizikiniais dydžiais):
```

```
figure(600)
subplot(3,2,1);
plot(R1100(:,1)),hold on
grid on
title('Auðyklės momentinė šiluminė galia')
ylabel('Galija, MW');
subplot('10NDG20CQ901XQ01');
xlabel('Matavimo taðkø skaièius');

subplot(3,2,2);
plot(R1100(:,2)),hold on
grid on
title('Termofikacinio vandens srautas per šilumokaitá')
subplot('10NDG20CF901XQ01')
ylabel('Srautas, kg/s');
xlabel('Matavimo taðkø skaièius');

subplot(3,2,3);
plot(R1100(:,3)),hold on
grid on
title('Termofikacinio vandens temperatūra prieš šilumokaitá')
subplot('10NDA30CT901XQ01')
ylabel('Temperatūra, °C');
xlabel('Matavimo taðkø skaièius');

subplot(3,2,4);
plot(R1100(:,4)),hold on
grid on
title('Termofikacinio vandens temperatūra up šilumokaitai')
subplot('10NDG20CT001XQ01')
ylabel('Temperatūra, °C');
xlabel('Matavimo taðkø skaièius');

figure(601)
subplot(3,2,1);
plot(R1100(:,5)),hold on
grid on
title('Glikolio vandens srautas per šilumokaitá')
subplot('10PGD20CF901XQ01')
ylabel('Srautas, kg/s');
xlabel('Matavimo taðkø skaièius');

subplot(3,2,2);
plot(R1100(:,6)),hold on
grid on
title('Glikolio vandens temperatūra up auðyklės')
subplot('10PGD50CT002XQ01')
ylabel('Temperatūra, °C');
xlabel('Matavimo taðkø skaièius');

subplot(3,2,3);
plot(R1100(:,7)),hold on
grid on
title('Glikolio vandens temperatūra prieš auðyklę')
subplot('10PGD30CT002XQ01')
ylabel('Temperatūra, °C');
xlabel('Matavimo taðkø skaièius');
```

```
% Apmokymo duomenų rinkinys (fizikiniais dydžiais):
```

```
figure(300)
subplot(3,2,1);
```

```

plot(R1(:,1)),hold on
grid on
title('Auðyklės momentinė šiluminė galia')
ylabel('Galija, MW');
subtitle('10NDG20CQ901XQ01');
xlabel('Matavimo taškų skaičius');

subplot(3,2,2);
plot(R1(:,2)),hold on
grid on
title('Termofikacinio vandens srautas per šilumokaitą')
subtitle('10NDG20CF901XQ01')
ylabel('Srautas, kg/s');
xlabel('Matavimo taškų skaičius');

subplot(3,2,3);
plot(R1(:,3)),hold on
grid on
title('Termofikacinio vandens temperatūra prieš šilumokaitą')
subtitle('10NDA30CT901XQ01')
ylabel('Temperatūra, °C');
xlabel('Matavimo taškų skaičius');

subplot(3,2,4);
plot(R1(:,4)),hold on
grid on
title('Termofikacinio vandens temperatūra už šilumokaitę')
subtitle('10NDG20CT001XQ01')
ylabel('Temperatūra, °C');
xlabel('Matavimo taškų skaičius');

figure(301)
subplot(3,2,1);
plot(R1(:,5)),hold on
grid on
title('Glikolio vandens srautas per šilumokaitą')
subtitle('10PGD20CF901XQ01')
ylabel('Srautas, kg/s');
xlabel('Matavimo taškų skaičius');

subplot(3,2,2);
plot(R1(:,6)),hold on
grid on
title('Glikolio vandens temperatūra už auðyklę')
subtitle('10PGD50CT002XQ01')
ylabel('Temperatūra, °C');
xlabel('Matavimo taškų skaičius');

subplot(3,2,3);
plot(R1(:,7)),hold on
grid on
title('Glikolio vandens temperatūra prieš auðyklę')
subtitle('10PGD30CT002XQ01')
ylabel('Temperatūra, °C');
xlabel('Matavimo taškų skaičius');

% Testavimo duomenų rinkinys (fizikiniai dydžiai):

figure(500)
subplot(3,2,1);
plot(R2(:,1)),hold on
grid on
title('Auðyklės momentinė šiluminė galia')
subtitle('10NDG20CQ901XQ01');

```

```

ylabel('Galia, MW');
xlabel('Matavimo taškų skaičius');

subplot(3,2,2);
plot(R2(:,2)),hold on
grid on
title('Termofikacinio vandens srautas per šilumokaitą')
subtitle('10NDG20CF901XQ01')
ylabel('Srautas, kg/s');
xlabel('Matavimo taškų skaičius');

subplot(3,2,3);
plot(R2(:,3)),hold on
grid on
title('Termofikacinio vandens temperatūra prieš šilumokaitą')
subtitle('10NDA30CT901XQ01')
ylabel('Temperatūra, °C');
xlabel('Matavimo taškų skaičius');

subplot(3,2,4);
plot(R2(:,4)),hold on
grid on
title('Termofikacinio vandens temperatūra už šilumokaitę')
subtitle('10NDG20CT001XQ01')
ylabel('Temperatūra, °C');
xlabel('Matavimo taškų skaičius');

figure(501)
subplot(3,2,1);
plot(R2(:,5)),hold on
grid on
title('Glikolio vandens srautas per šilumokaitą')
subtitle('10PGD20CF901XQ01')
ylabel('Srautas, kg/s');
xlabel('Matavimo taškų skaičius');

subplot(3,2,2);
plot(R2(:,6)),hold on
grid on
title('Glikolio vandens temperatūra už aušyklės')
subtitle('10PGD50CT002XQ01')
ylabel('Temperatūra, °C');
xlabel('Matavimo taškų skaičius');

subplot(3,2,3);
plot(R2(:,7)),hold on
grid on
title('Glikolio vandens temperatūra prieš aušyklę')
subtitle('10PGD30CT002XQ01')
ylabel('Temperatūra, °C');
xlabel('Matavimo taškų skaičius');

% Apmokymo duomenų rinkinys (skaliuoti)

figure(100)
subplot(3,2,1);
plot(T_AUX_cooler_sc(1,:)),hold on
grid on
title('Aušyklės momentinė šiluminė galia')
subtitle('10NDG20CQ901XQ01');
xlabel('Matavimo taškų skaičius');

subplot(3,2,2);
plot(T_AUX_cooler_sc(2,:)),hold on

```

```

grid on
title('Termofikacinio vandens srautas per õilumokaitá')
subtitle('10NDG20CF901XQ01')
xlabel('Matavimo taðkø skaièius');

subplot(3,2,3);
plot(T_AUX_cooler_sc(3,:),hold on
grid on
title('Termofikacinio vandens temperatûra prieð õilumokaitá')
subtitle('10NDA30CT901XQ01')
xlabel('Matavimo taðkø skaièius');

subplot(3,2,4);
plot(T_AUX_cooler_sc(4,:),hold on
grid on
title('Termofikacinio vandens temperatûra up õilumokaièio')
subtitle('10NDG20CT001XQ01')
xlabel('Matavimo taðkø skaièius');

figure(101)
subplot(3,2,1);
plot(T_AUX_cooler_sc(5,:),hold on
grid on
title('Glikolio vandens srautas per õilumokaitá')
subtitle('10PGD20CF901XQ01')
xlabel('Matavimo taðkø skaièius');

subplot(3,2,2);
plot(T_AUX_cooler_sc(6,:),hold on
grid on
title('Glikolio vandens temperatûra up auðyklës')
subtitle('10PGD50CT002XQ01')
xlabel('Matavimo taðkø skaièius');

subplot(3,2,3);
plot(T_AUX_cooler_sc(7,:),hold on
grid on
title('Glikolio vandens temperatûra prieð auðyklæ')
subtitle('10PGD30CT002XQ01')
xlabel('Matavimo taðkø skaièius');

% Testavimo duomenų rinkinys (skaliuoti)

figure(200)
subplot(3,2,1);
plot(V_AUX_cooler_sc(1,:),hold on
grid on
title('Auðyklës momentinë õiluminë galia')
subtitle('10NDG20CQ901XQ01');
xlabel('Matavimo taðkø skaièius');

subplot(3,2,2);
plot(V_AUX_cooler_sc(2,:),hold on
grid on
title('Termofikacinio vandens srautas per õilumokaitá')
subtitle('10NDG20CF901XQ01')
xlabel('Matavimo taðkø skaièius');

subplot(3,2,3);
plot(V_AUX_cooler_sc(3,:),hold on
grid on
title('Termofikacinio vandens temperatûra prieð õilumokaitá')
subtitle('10NDA30CT901XQ01')
xlabel('Matavimo taðkø skaièius');

```

```

subplot(3,2,4);
plot(V_AUX_cooler_sc(4,:),hold on
grid on
title('Termofikacinio vandens temperatūra up̄ ðilumokaièio')
subtitle('10NDG20CT001XQ01')
xlabel('Matavimo taðkø skaièius');

figure(201)
subplot(3,2,1);
plot(V_AUX_cooler_sc(5,:),hold on
grid on
title('Glikolio vandens srautas per ðilumokaitá')
subtitle('10PGD20CF901XQ01')
xlabel('Matavimo taðkø skaièius');

subplot(3,2,2);
plot(V_AUX_cooler_sc(6,:),hold on
grid on
title('Glikolio vandens temperatūra up̄ auðyklès')
subtitle('10PGD50CT002XQ01')
xlabel('Matavimo taðkø skaièius');

subplot(3,2,3);
plot(V_AUX_cooler_sc(7,:),hold on
grid on
title('Glikolio vandens temperatūra prieð auðyklæ')
subtitle('10PGD30CT002XQ01')
xlabel('Matavimo taðkø skaièius');

% Matavimo duomenø struktura:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 10NDG20CQ901XQ01,10NDG20CF901XQ01,10NDA30CT901XQ01 ,c ,%
% 1, 2, 3, 4, %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 10PGD20CF901XQ01,10PGD50CT002XQ01 ,10PGD30CT002XQ01 ,%
% 5, 6, 7, %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Ávesties eksperimentiniø duomenø rinkinio formavimas:

input_No=[1 2 3 4 5 6 7];

% AADNT strukturos apibrèzimas:
% Mapping, Bottleneck, De-Mapping, Output,
%
NetDef=['HHHHHHHH--';'LLL-----';'HHHHHHHH--';'LLLLLLL---'];
NetDefC=XNETDEF(NetDef); % Konvertavimas is simboliu i skaičius Levenberg-
Marquardt paprogrameje;

% Jeigu apibrèztos 2 savybès - papildoma fazes trajektorija:

if length(find(NetDefC(:,2)==0))==2 | length(find(NetDefC(:,2)==1))==2,
detail=1; else detail=0; end,

% "Sugedusio" jutiklio numeris (id) ivesties duomenø rinkinyje atsizvelgiant i
input_No elementus:

sens_No=1;
isens=input_No(sens_No);

% Kiekvieno sluoksnio mazgu apibrèzimas:

mapp=length((find(NetDef(1,:)=='L' | NetDef(1,:)=='H'))');
bottleneck=length((find(NetDef(2,:)=='L' | NetDef(2,:)=='H'))');

```

```

demapp=length((find(NetDef(3,:)=='L' | NetDef(3,:)=='H'))');
outputs=length((find(NetDef(4,:)=='L' | NetDef(4,:)=='H'))');
inputs=outputs; % Ivesties kintamuju skaicius = Isvesties kintamuju skaiciui;
IN_tr = T_AUX_cooler_sc(input_No(1:inputs),:); % Apmokymo duomenys;
OUT_tr = T_AUX_cooler_sc(input_No(1:inputs),:); % Apmokymo duomenys;

% Levenberg-Marquardt paprogrames nustatymai

M_options= [10 0 1 0];

% Iteracijų skaicius:

niter=20;

% Atsitiktiniu mokymosi ciklu skaicius, priklausantis nuo svoriu skaiciaus
tinkle:
nrnd=10*(1+fix((outputs*(demapp+1)+demapp*bottleneck+bottleneck*(mapp+1)+mapp*(i
nputs+1))/5.));

fprintf('\n Vykdomas AADNT apmokymas...\n');
train_time=cputime;

% Pagrindinis apmokymo ciklas
% Isorine kilpa - atsitiktinės pradines tinklo svorio salygos.
% Tai atliekama kelis kartus, norint gauti apmokymo rezultatu statistika, is
kurios paimami geriausi rezultatai.

for irnd=1:nrnd

    W4=(rand(outputs,demapp+1)-0.5);
    W3=(rand(demapp,bottleneck)-0.5);
    W2=(rand(bottleneck,mapp+1)-0.5);
    W1=(rand(mapp,inputs+1)-0.5);

    W1A=W1; W2A=W2; % Pradiniai svoriai - visada pradedama nuo to paties;
    W3A=W3; W4A=W4; % Pradiniai svoriai - visada pradedama nuo to paties;

% Vidinė kilpa - tinklo apmokymas su duotais pradiniais svoriais

    for iter=1:niter
        [W1A,W2A,W3A,W4A,PI,it,lambda]=
MARQX2NN(NetDefC,W1A,W2A,W3A,W4A,IN_tr,OUT_tr,M_options);
        [Ytr,Ypseudo,RMStr(iter)] =
NNEVAL4E(NetDefC,W1A,W2A,W3A,W4A,IN_tr,OUT_tr);
        if lambda>1.e8, break, end;
    end % iteracijos pabaiga

    [RMSmin2,best_iter2]=min(RMStr');

% Apskaiciuojamas geriausias sprendimas

    if RMSmin2<training_RMSE,

        W1R=W1; W2R=W2; % Pradiniai svoriai - visada pradedama nuo to paties
        W3R=W3; W4R=W4; % Pradiniai svoriai - visada pradedama nuo to paties

        for iter=1:best_iter2
            [W1R,W2R,W3R,W4R,PI,it,lambda]=
MARQX2NN(NetDefC,W1R,W2R,W3R,W4R,IN_tr,OUT_tr,M_options);
            [Ytr,Ypseudo,RMStr(iter)] =
NNEVAL4E(NetDefC,W1R,W2R,W3R,W4R,IN_tr,OUT_tr);
        end % iteracija

        training_RMSE=RMSmin2

```



```

% RMSE apmokymo paklaida
figure(20)
plot(RMStr), hold off
grid on
title('Vidutinė kvadratinė paklaida')
ylabel('Paklaida');
xlabel('Iteracijø skaièius');
end;

end % iteracijos pabaiga

train_time2=cputime-train_time;

% AADNT apmokymo rezultatu (skaliuotø) diagramos:

figure(1)
subplot(3,2,1);
[Ytr,Ypseudo,RMStr(iter)]=NNEVAL4E(NetDefC,W1R,W2R,W3R,W4R,IN_tr,OUT_tr);
plot(Ytr(1,:), '<'),hold on
plot(OUT_tr(1,:), '>'),hold off
grid on
title('Auðyklës momentinë ðiluminë galia')
subtitle('>" AADNT ávestis, "<" AADNT iðvestis, 10NDG20CQ901XQ01');
xlabel('Matavimo taðkø skaièius');

subplot(3,2,2);
[Ytr,Ypseudo,RMStr(iter)]=NNEVAL4E(NetDefC,W1R,W2R,W3R,W4R,IN_tr,OUT_tr);
plot(Ytr(2,:), '<'),hold on
plot(OUT_tr(2,:), '>'),hold off
grid on
title('Termofikacinio vandens srautas per ðilumokaitá')
subtitle('>" AADNT ávestis, "<" AADNT iðvestis, 10NDG20CF901XQ01')
xlabel('Matavimo taðkø skaièius');

subplot(3,2,3);
[Ytr,Ypseudo,RMStr(iter)]=NNEVAL4E(NetDefC,W1R,W2R,W3R,W4R,IN_tr,OUT_tr);
plot(Ytr(3,:), '<'),hold on
plot(OUT_tr(3,:), '>'),hold off
grid on
title('Termofikacinio vandens temperatûra prieš ðilumokaitá')
subtitle('>" AADNT ávestis, "<" AADNT iðvestis, 10NDA30CT901XQ01')
xlabel('Matavimo taðkø skaièius');

subplot(3,2,4);
[Ytr,Ypseudo,RMStr(iter)]=NNEVAL4E(NetDefC,W1R,W2R,W3R,W4R,IN_tr,OUT_tr);
plot(Ytr(4,:), '<'),hold on
plot(OUT_tr(4,:), '>'),hold off
grid on
title('Termofikacinio vandens temperatûra up ðilumokaièio')
subtitle('>" AADNT ávestis, "<" AADNT iðvestis, 10NDG20CT001XQ01')
xlabel('Matavimo taðkø skaièius');

figure(11)
subplot(3,2,1);
[Ytr,Ypseudo,RMStr(iter)]=NNEVAL4E(NetDefC,W1R,W2R,W3R,W4R,IN_tr,OUT_tr);
plot(Ytr(5,:), '<'),hold on
plot(OUT_tr(5,:), '>'),hold off
grid on
title('Glikolio vandens srautas per ðilumokaitá')
subtitle('>" AADNT ávestis, "<" AADNT iðvestis, 10PGD20CF901XQ01')
xlabel('Matavimo taðkø skaièius');

subplot(3,2,2);
[Ytr,Ypseudo,RMStr(iter)]=NNEVAL4E(NetDefC,W1R,W2R,W3R,W4R,IN_tr,OUT_tr);

```

```

plot(Ytr(6,:), '<'), hold on
plot(OUT_tr(6,:), '>'), hold off
grid on
title('Glikolio vandens temperatūra up auðyklës')
subtitle('>" AADNT ávestis, "<" AADNT iðvestis, 10PGD50CT002XQ01')
xlabel('Matavimo taðkø skaièius');

subplot(3,2,3);
[Ytr, Ypseudo, RMStr(iter)] = NNEVAL4E(NetDefC, W1R, W2R, W3R, W4R, IN_tr, OUT_tr);
plot(Ytr(7,:), '<'), hold on
plot(OUT_tr(7,:), '>'), hold off
grid on
title('Glikolio vandens temperatūra prieð auðyklæ')
subtitle('>" AADNT ávestis, "<" AADNT iðvestis, 10PGD30CT002XQ01')
xlabel('Matavimo taðkø skaièius');

% Apmokymo duomeni perskaiciavimas i fizikinius dydzius (tinklo investis, tinklo
isvestis):

T_AUX_cooler_sk = [T_AUX_cooler(input_No(1:inputs), :)];
std_traindata = std(T_AUX_cooler_sk, [], 2);
mean_traindata = mean(T_AUX_cooler_sk, 2);
T_AUX_cooler_fizik_in = (OUT_tr.*std_traindata)+mean_traindata;
T_AUX_cooler_fizik_out = (Ytr.*std_traindata)+mean_traindata;

% AADNT apmokymo rezultatu (fizikiniais dydziais) diagramos:

figure(5)
subplot(3,2,1);
plot(T_AUX_cooler_fizik_in(1,:), hold on
plot(T_AUX_cooler_fizik_out(1,:), hold off
grid on
title('Auðyklës momentinë ðiluminë galia')
subtitle('10NDG20CQ901XQ01');
ylabel('Galija, MW');
xlabel('Matavimo taðkø skaièius');
legend('Ávestis', 'Iðvestis')

subplot(3,2,2);
plot(T_AUX_cooler_fizik_in(2,:), hold on
plot(T_AUX_cooler_fizik_out(2,:), hold off
grid on
title('Termofikacinio vandens srautas per ðilumokaitá')
subtitle('10NDG20CF901XQ01')
ylabel('Srautas, kg/s');
xlabel('Matavimo taðkø skaièius');
legend('Ávestis', 'Iðvestis')

subplot(3,2,3);
plot(T_AUX_cooler_fizik_in(3,:), hold on
plot(T_AUX_cooler_fizik_out(3,:), hold off
grid on
title('Termofikacinio vandens temperatūra prieð ðilumokaitá')
subtitle('10NDA30CT901XQ01')
ylabel('Temperatūra, °C');
xlabel('Matavimo taðkø skaièius');
legend('Ávestis', 'Iðvestis')

subplot(3,2,4);
plot(T_AUX_cooler_fizik_in(4,:), hold on
plot(T_AUX_cooler_fizik_out(4,:), hold off
grid on
title('Termofikacinio vandens temperatūra up ðilumokaièio')
subtitle('10NDG20CT001XQ01')

```

```

ylabel('Temperatūra, °C');
xlabel('Matavimo taškų skaičius');
legend('Ávestis', 'Iðvestis')

figure(55)
subplot(3,2,1);
plot(T_AUX_cooler_fizik_in(5,:),hold on
plot(T_AUX_cooler_fizik_out(5,:),hold off
grid on
title('Glikolio vandens srautas per ðilumokaitá')
subtitle('10NDG20CT001XQ01')
ylabel('Srautas, kg/s');
xlabel('Matavimo taškų skaičius');
legend('Ávestis', 'Iðvestis')

subplot(3,2,2);
plot(T_AUX_cooler_fizik_in(6,:),hold on
plot(T_AUX_cooler_fizik_out(6,:),hold off
grid on
title('Glikolio vandens temperatūra up auðyklës')
subtitle('10PGD50CT002XQ01')
ylabel('Temperatūra, °C');
xlabel('Matavimo taškų skaičius');
legend('Ávestis', 'Iðvestis')

subplot(3,2,3);
plot(T_AUX_cooler_fizik_in(7,:),hold on
plot(T_AUX_cooler_fizik_out(7,:),hold off
grid on
title('Glikolio vandens temperatūra prieš auðyklæ')
subtitle('10PGD30CT002XQ01')
ylabel('Temperatūra, °C');
xlabel('Matavimo taškų skaičius');
legend('Ávestis', 'Iðvestis')

figure(2)
plot(T_AUX_cooler(1,:),Ypseudo,'*'),hold off
grid on
title('Apmokymo rezultatai: "*" , Testavimo rezultatai: "o"');
ylabel('Savybës (trajektorijos popymiø sluoksnyje)');
xlabel('Matavimo taškų skaičius');

if detail==1
figure(4), grid on
if bottleneck==2,
plot(Ypseudo(1,:),Ypseudo(2:),'m*'),hold off
ylabel('2 - a savybë');xlabel('1 - a savybë');

else
if bottleneck>2,
plot3(Ypseudo(1,:),Ypseudo(2:),Ypseudo(3:),'m*'),hold off
ylabel('2 - a savybë');xlabel('1 - a savybë');zlabel('3 - a
savybë');
end
end
title('Fazinës trajektorijos "popymiø sluoksnyje: apmokymas "*" ,
testavimas "o"');
end

% Optimizuotu svoriu issaugojimas:

asave=input ('AADNT apmokymas baigtas.\n\nAr iðsaugoti tinklo parametrus W1...W4
? (y/[n]) ', 's');
train_time=cputime;

```

```

if (asave=='y'),
    fid = fopen('W4.dat','w');
    fprintf(fid,'%12.9f ', W4R);
    fclose(fid);
    fid = fopen('W3.dat','w');
    fprintf(fid,'%12.9f ', W3R);
    fclose(fid);
    fid = fopen('W2.dat','w');
    fprintf(fid,'%12.9f ', W2R);
    fclose(fid);
    fid = fopen('W1.dat','w');
    fprintf(fid,'%12.9f ', W1R);
    fclose(fid);

delete('W1.mat');
delete('W2.mat');
delete('W3.mat');
delete('W4.mat');

save('W4.mat','W4R');
save('W3.mat','W3R');
save('W2.mat','W2R');
save('W1.mat','W1R');

save('D:\OneDrive - Kaunas University of
Technology\Magistrantūra\Magistrinis_projektas\Modeliavimas_Mantas\Modelis_2021_
04\Apmokyto_AADNT_testavimas\W4_done.mat','W4R');
save('D:\OneDrive - Kaunas University of
Technology\Magistrantūra\Magistrinis_projektas\Modeliavimas_Mantas\Modelis_2021_
04\Apmokyto_AADNT_testavimas\W3_done.mat','W3R');
save('D:\OneDrive - Kaunas University of
Technology\Magistrantūra\Magistrinis_projektas\Modeliavimas_Mantas\Modelis_2021_
04\Apmokyto_AADNT_testavimas\W2_done.mat','W2R');
save('D:\OneDrive - Kaunas University of
Technology\Magistrantūra\Magistrinis_projektas\Modeliavimas_Mantas\Modelis_2021_
04\Apmokyto_AADNT_testavimas\W1_done.mat','W1R');
end

% "Sugedusio" jutiklio virtualizavimas
% Perskaiciuojama tiesine transformacija sugedusio jutiklio renormalizavimui:

gain=(V_AUX_cooler(isens,length(V_AUX_cooler_sc))-
V_AUX_cooler(isens,1))/(V_AUX_cooler_sc(isens,length(V_AUX_cooler_sc))-
V_AUX_cooler_sc(isens,1));
bias=V_AUX_cooler(isens,length(V_AUX_cooler_sc))-
V_AUX_cooler_sc(isens,length(V_AUX_cooler_sc))*gain;

for itst=1:length(V_AUX_cooler_sc),
    IN1_tst = V_AUX_cooler_sc(input_No(1:inputs),itst);
    IN2_tst = V_AUX_cooler(input_No(1:inputs),itst);

    if itst==1,
        p0=V_AUX_cooler_sc(isens,itst);
    else
        p0=xsens;
    end
end

% Optimizavimas: jutiklio vertes imitacija tinklo ivestyje:

options1=optimset('MaxFunEvals',1000,'Display','off');
[xsens,J]=fminsearch('NNEVAL4ERR',p0,options1,NetDefC,W1R,W2R,W3R,W4R,IN1_tst,IN
1_tst,sens_No);
IN1_tst(sens_No)=xsens;

```

```

[Ytst, Ypseudo2, RMStst(itst)] =
NNEVAL4E(NetDefC, W1R, W2R, W3R, W4R, IN1_tst, IN1_tst);
IN2_tst(sens_No)=xsens*gain+bias;
Ytst=Ytst*gain+bias;

% Diagramu atnaujinimas testuotais duomenimis:

figure(3)
plot(V_AUX_cooler(1, itst), Ytst(sens_No), '<b'), hold on
plot(V_AUX_cooler(1, itst), IN2_tst(sens_No), '>g'), hold on
plot(V_AUX_cooler(1, itst), V_AUX_cooler(isens, itst), 'or'), hold on
grid on
%xlim([0 7])
%ylim([39 45])
title("Sugedusio jutiklio virtualizavimas: AADNT ávestis ">", iðvestis
"<", procesas "o");
ylabel('Jutiklio signalas'); xlabel('Matavimo taðkø skaièius');
Ypseudo3=[Ypseudo3 Ypseudo2];

end % iteracijos pabaiga

figure(2), hold on
plot(V_AUX_cooler(1, :), Ypseudo3', 'o-'), hold on

if detail==1
figure(4), hold on, grid on
if bottleneck==2,
plot(Ypseudo3(1, :), Ypseudo3(2, :), 'co-'), hold on
else
if bottleneck>2, plot3(Ypseudo3(1, :), Ypseudo3(2, :), Ypseudo3(3, :), 'co-
'), hold on, end
end
end
fprintf('\nBendras skaièiavimø laikas buvo %6.2f [s]\n', cputime-
train_time+train_time2)

% AADNT paklaidos - kokybiniai parametrai
% Santykine procentine paklaida

Target=T_AUX_cooler_fizik_out(1, :);
Prediction=T_AUX_cooler_fizik_in(1, :);
Paklaida_1=errperf(Target, Prediction, 'pe');

Target=T_AUX_cooler_fizik_out(2, :);
Prediction=T_AUX_cooler_fizik_in(2, :);
Paklaida_2=errperf(Target, Prediction, 'pe');

Target=T_AUX_cooler_fizik_in(3, :);
Prediction=T_AUX_cooler_fizik_in(3, :);
Paklaida_3=errperf(Target, Prediction, 'pe');

Target=T_AUX_cooler_fizik_out(4, :);
Prediction=T_AUX_cooler_fizik_out(4, :);
Paklaida_4=errperf(Target, Prediction, 'pe');

Target=T_AUX_cooler_fizik_out(5, :);
Prediction=T_AUX_cooler_fizik_in(5, :);
Paklaida_5=errperf(Target, Prediction, 'pe');

Target=T_AUX_cooler_fizik_out(6, :);
Prediction=T_AUX_cooler_fizik_in(6, :);
Paklaida_6=errperf(Target, Prediction, 'pe');

Target=T_AUX_cooler_fizik_out(7, :);

```

```

Prediction=T_AUX_cooler_fizik_in(7,:);
Paklaida_7=errperf(Target,Prediction,'pe');

figure(6)
subplot(3,2,1);
bar(Paklaida_1)
grid on
title('Auðyklės momentinė šiluminė galia')
subtitle('10NDG20CQ901XQ01')
ylabel('Paklaida, %');
xlabel('Matavimo taškų skaičius');

subplot(3,2,2);
bar(Paklaida_2)
grid on
title('Termofikacinio vandens srautas per šilumokaitą')
subtitle('10NDG20CF901XQ01')
ylabel('Paklaida, %');
xlabel('Matavimo taškų skaičius');

subplot(3,2,3);
bar(Paklaida_3)
grid on
title('Termofikacinio vandens temperatūra prieš šilumokaitą')
subtitle('10NDA30CT901XQ01')
ylabel('Paklaida, %');
xlabel('Matavimo taškų skaičius');

subplot(3,2,4);
bar(Paklaida_4)
grid on
title('Termofikacinio vandens temperatūra už šilumokaitą')
subtitle('10NDG20CT001XQ01')
ylabel('Paklaida, %');
xlabel('Matavimo taškų skaičius');

figure(66)

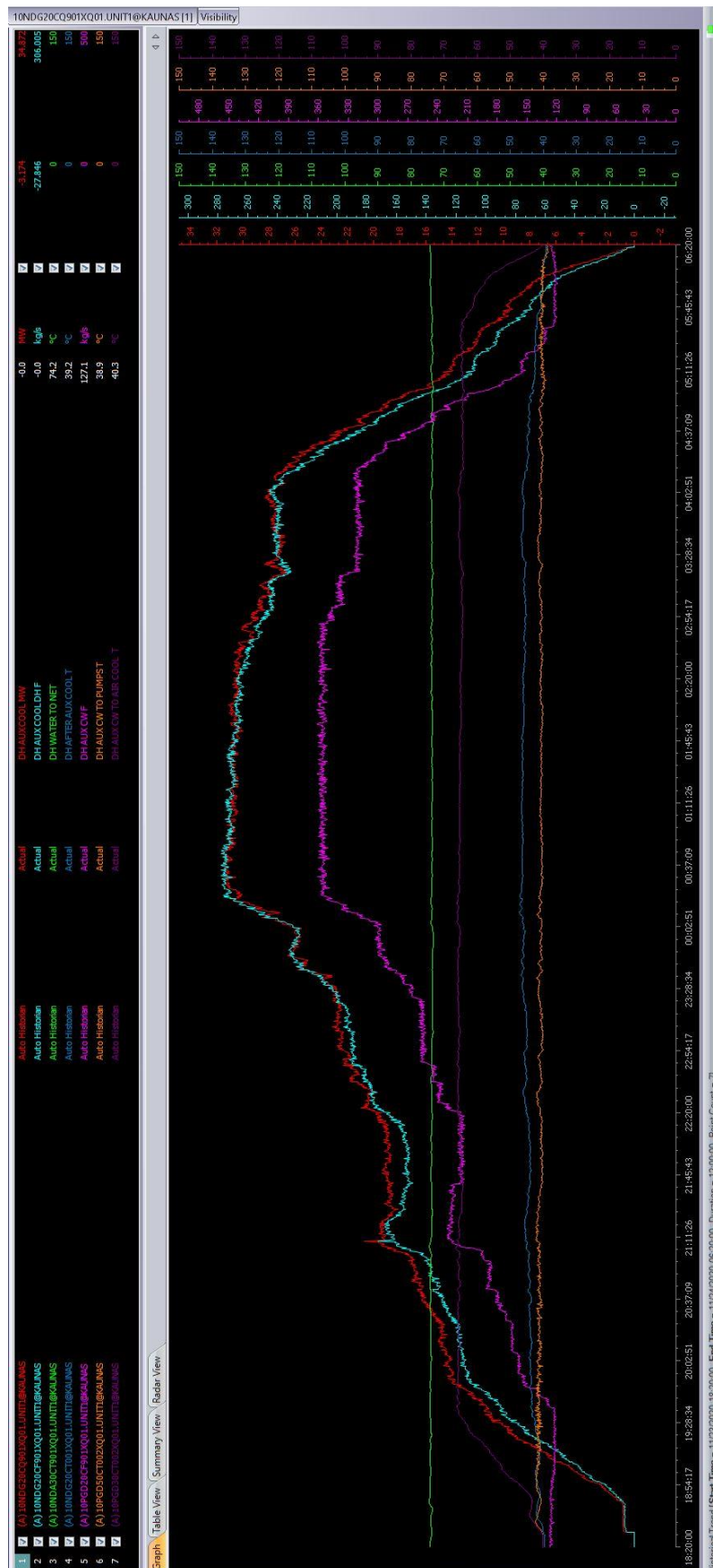
subplot(3,2,1);
bar(Paklaida_5)
grid on
title('Glikolio vandens srautas per šilumokaitą')
subtitle('10PGD20CF901XQ01')
ylabel('Paklaida, %');
xlabel('Matavimo taškų skaičius');

subplot(3,2,2);
bar(Paklaida_6)
grid on
title('Glikolio vandens temperatūra už auðyklės')
subtitle('10PGD50CT002XQ01')
ylabel('Paklaida, %');
xlabel('Matavimo taškų skaičius');

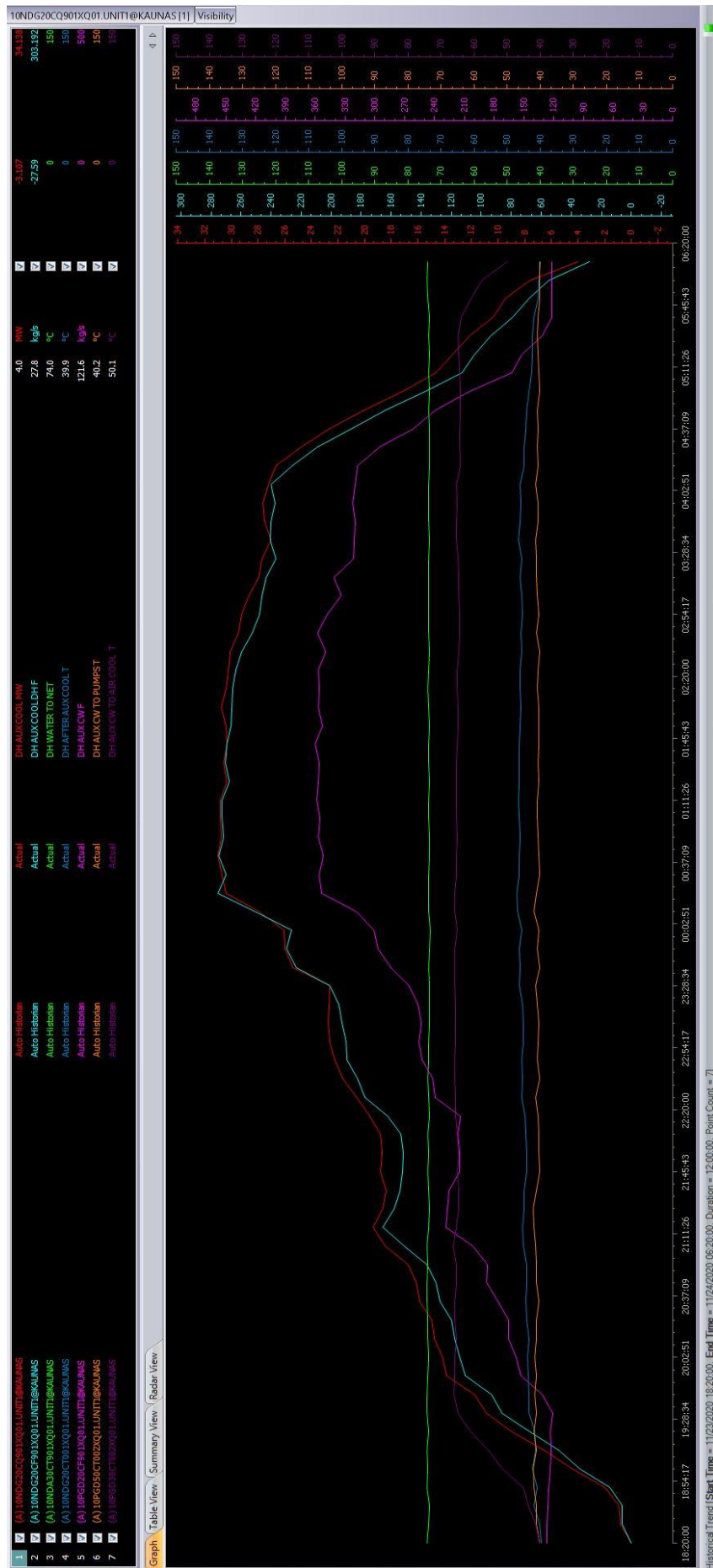
subplot(3,2,3);
bar(Paklaida_7)
grid on
title('Glikolio vandens temperatūra prieš auðyklę')
subtitle('10PGD30CT002XQ011')
ylabel('Paklaida, %');
xlabel('Matavimo taškų skaičius');

```

2 priedas. Termofikacinio vandens aušyklės duomenų rinkinio grafikai iš kietu kuru kūrenamos kogeneracinės jėgainės, paskirstytos valdymo sistemos (DCS)



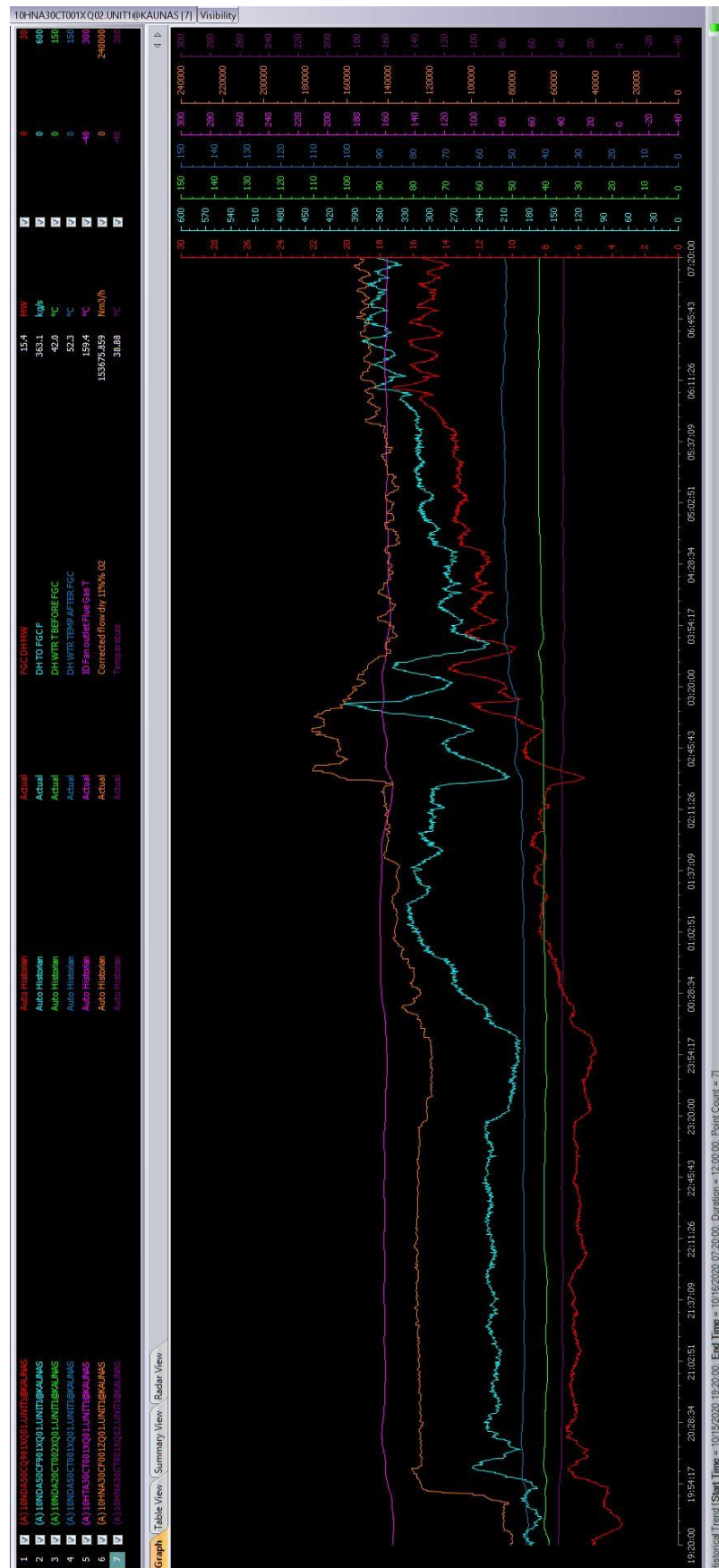
72 pav. Termofikacinio vandens aušyklės duomenų rinkinys (2000 taškų)



73 pav. Termofikacinio vandens aušyklės duomenų rinkinys (70 taškų)

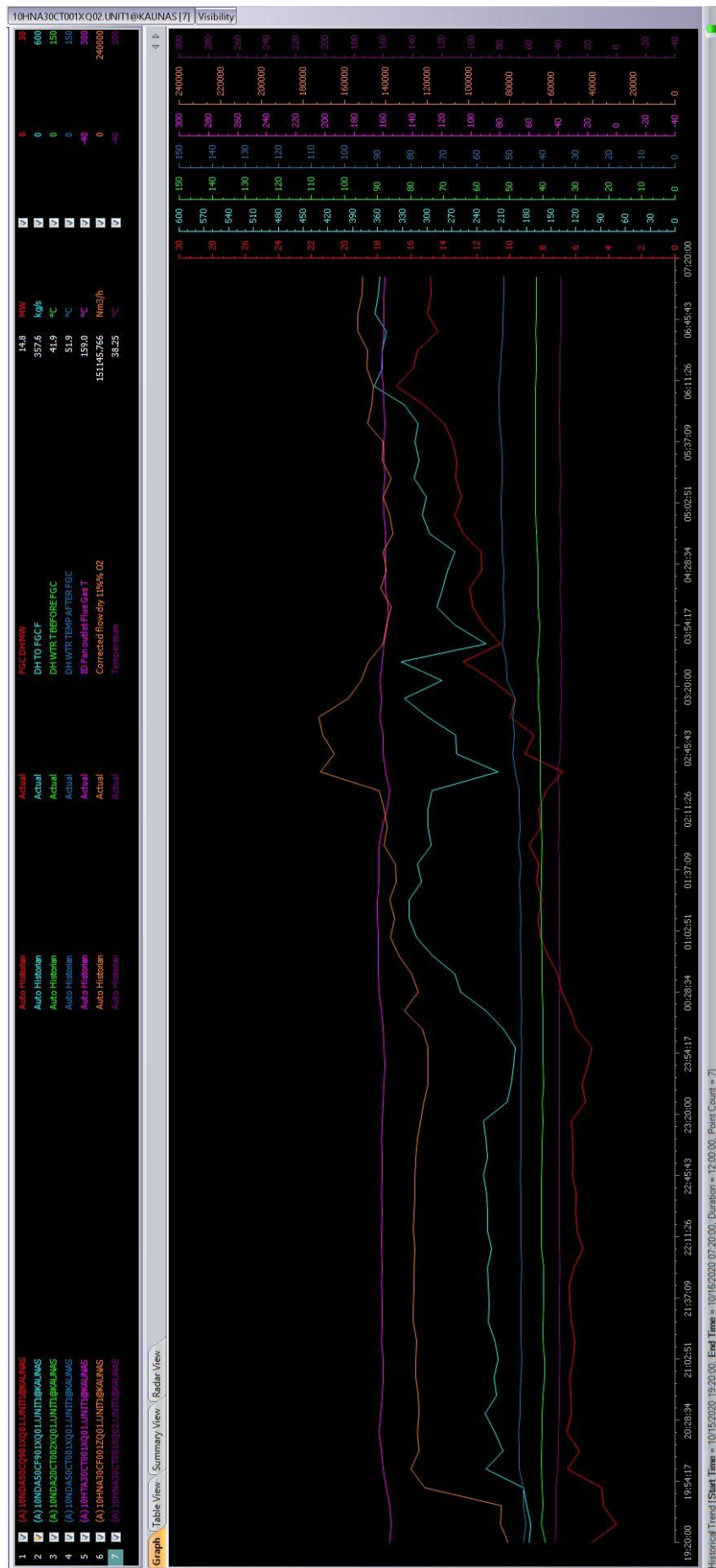


### 3 priedas. Kondensacinio ekonomaizerio duomenų rinkinio grafikai iš kietu kuru kūrenamos kogeneracinės jėgainės, paskirstytos valdymo sistemos (DCS)

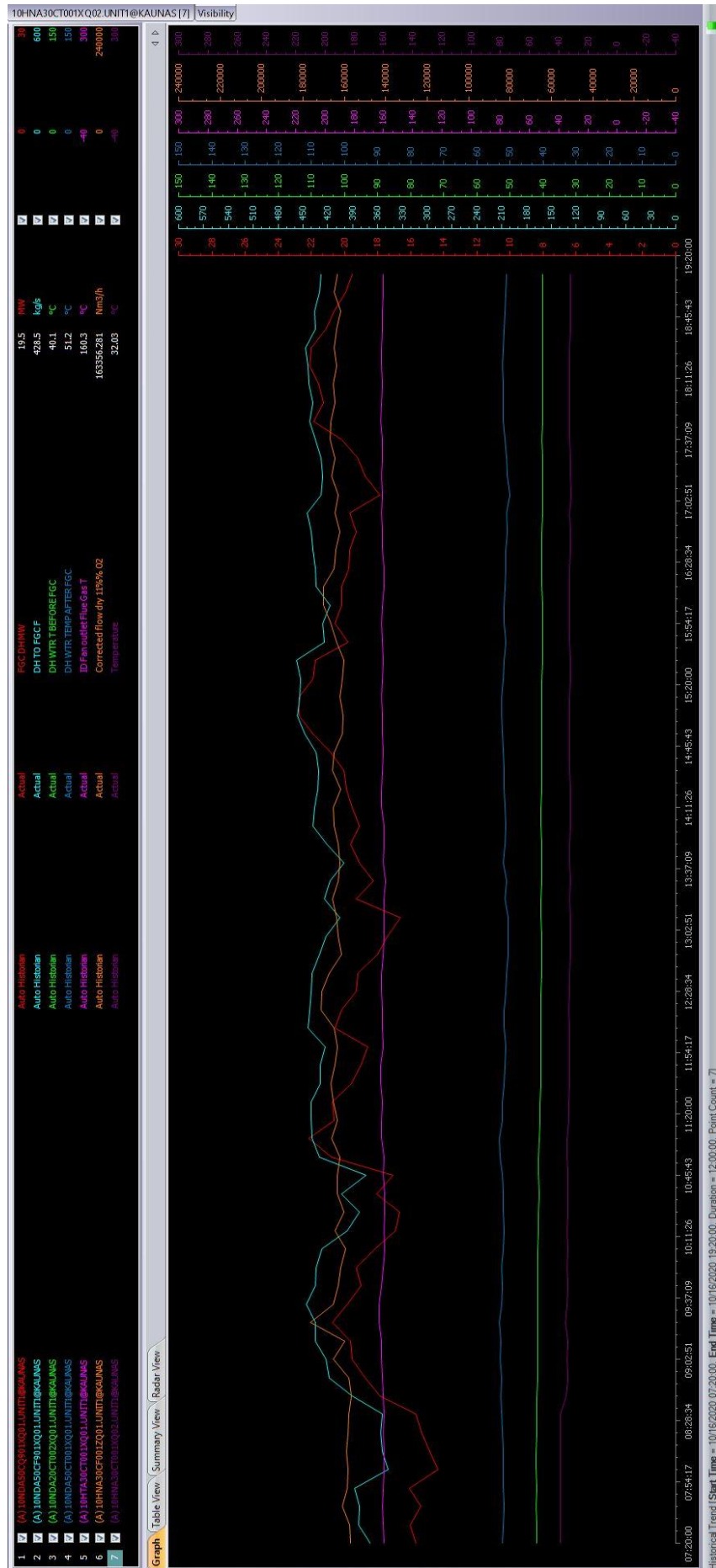


74 pav. Kondensacinio ekonomaizerio duomenų rinkinys (2000 taškų, rinkinys Nr. 1)





76 pav. Kondensacinio ekonomizerio duomenų rinkinys (70 taškų, rinkinys Nr. 1)



77 pav. Kondensacinio ekonomizerio duomenų rinkinys (70 taškų)

#### 4 priedas. Termofikacinio vandens aušyklės AANT apmokymo duomenų rinkinys

7 lentelė. Termofikacinio vandens aušyklės AANT apmokymo duomenų rinkinys

10NDG2 0CQ901 XQ01	10NDG2 0CF901 XQ01	10NDA3 0CT901 XQ01	10NDG2 0CT001 XQ01	10PGD2 0CF901 XQ01	10PGD5 0CT002 XQ01	10PGD3 0CT002 XQ01
MW	kg/s	°C	°C	kg/s	°C	°C
1	2	3	4	5	6	7
4,4	34,2	74,1	42,4	123,6	42,3	51,7
6,6	48,5	74,2	41,3	122,9	41,2	55,8
9	68,2	73,8	41,8	123,2	41,8	61,2
10,9	85,9	74,2	43,4	121,3	41,2	65
11,8	93	74,2	43,5	131,5	41,7	66,1
13,9	110,6	74,1	43,8	153,1	41,2	65,9
14,2	114,5	73,8	44	158,1	41,8	65,9
14,8	117,5	73,9	43,5	165,5	41,2	65,3
14,9	120,1	74,1	44,1	165,3	41,2	65,7
15,9	127,5	74,3	44,4	175,9	40,5	65,7
16,1	130,3	74,1	44,1	187,7	41,2	65,4
16,7	135,9	73,9	44,2	186,9	41,3	65,4
18,4	151,5	74,3	45,1	201,6	41,4	66,3
19,4	165,3	73,7	45,4	228,3	42	65,5
18,7	158,1	73,5	45	226,6	42,2	64,7
18,4	154,3	73,7	44,9	225	41,1	64,7
18,8	152,6	74	44,2	214,7	40,3	65,1
18,7	152,2	73,7	44,1	214,8	40,5	64,7
18,8	153,6	73,9	44,3	216,3	40,5	65,1
19,6	162,5	73,6	44,6	213,6	40,6	65,6
20,6	177,5	73,7	45,7	239,5	41,6	65,6
21,6	182,6	73,9	45,4	242	40,3	66
22,2	189,6	73,8	45,7	252,5	41	66
22,6	190,1	73,7	45,1	255,6	40,8	65,6
22,8	193	73,6	45,2	253	40,5	65,5
22,7	194,9	73,5	45,4	256,3	41	65,7
22,6	201,1	73,4	46,3	265,4	41,5	65,7
25,4	223	73,7	46,2	283,9	40,3	66
26	229,8	73,6	46,3	296,8	40,9	65,9
26	226,3	73,3	45,7	300,8	40,4	65,2
27,9	250,8	73,8	46,9	318,9	41,9	66
30,4	275,6	73,8	47,2	353,9	41,2	66
30,7	270,2	73,7	46,3	355,9	40,3	65,4
31	274,8	73,6	46,4	352,3	40,5	65,4
30,8	271,6	73,5	46,2	357,2	40,6	65,2
30,9	272,7	73,7	46,4	355,4	41,1	65,5
30,8	272,6	73,7	46,6	358,6	40,9	65,5
30,3	267,6	73,6	46,3	356,6	40,6	65,3
30,6	270,6	73,7	46,4	356,5	40,6	65,3
30,4	269,2	73,7	46,4	360,6	40,4	65,2
30,4	266,9	73,6	46,1	352,9	40,4	65

30,8	266,4	73,7	45,8	356,8	40,4	65
30,4	265,5	73,9	46,2	356,4	41	65,2
30,2	263,5	73,8	46,2	357	40,7	65,2
30,1	259,5	73,6	45,6	349,5	40,3	64,6
29,5	252,7	73,8	45,8	357,5	41,3	64,4
29,2	247,9	73,7	45,4	348	40,3	64,4
28,7	246,2	73,7	45,5	334	41	64,6
27,9	243,4	73,7	46,1	340,9	41,1	64,7
27,7	236,7	73,8	45,6	321,9	41,2	64,9
27,1	240,4	73,6	46,6	320,4	41,5	65,5
27,5	240	73,8	46,1	319,9	40,8	65,4
27,6	237,6	73,8	45,8	322,5	40,7	65,2
27,2	240	73,7	46,2	319,8	41,3	65,5
26,7	226,2	73,6	45	317,1	40,8	64,2
24,8	208,8	73,6	44,8	295,1	40,9	64,5
22,7	186,5	73,6	44,4	261,2	40,6	64,4
20	164	73,7	44,1	239,3	40,9	64,2
17,2	137,4	73,6	43,5	205,1	40,2	64,1
14,6	113	73,7	42,6	161,8	40,4	64,4
13,4	104,7	73,4	42,5	151,8	40,7	64,7
12,2	93,6	73,8	42,4	131,4	41	64,9
10,4	79	73,6	41,8	121,6	40,7	64
9,5	68,1	74,1	40,7	121,6	40,3	60,9
7,6	55,1	74,2	40,5	121,7	40,2	57,7
4	27,8	74	39,9	121,6	40,2	50,1

## 5 priedas. Apmokyto termofikacinio vandens aušyklės AANT neuronų svoriai

8 lentelė. Apmokyto termofikacinio vandens aušyklės AANT neuronų svoriai

W1							
0,1276	-0,380	-0,3908	0,1445	-0,2080	0,3856	-0,1489	0,1381
0,2130	0,3992	0,2605	-0,0904	-0,0552	0,2020	0,4876	0,4236
0,4238	0,0105	-0,0220	-0,0557	-0,2794	0,2939	-0,0790	0,1668
0,3166	0,0562	-0,0628	0,2486	-0,3809	-0,0556	0,1266	0,3487
-0,1150	0,0467	-0,1961	0,3761	-0,3428	-0,3601	-0,2986	-0,3162
0,2333	-0,0612	0,1664	0,3962	-0,5071	-0,0947	-0,3674	-0,4477
0,2169	-0,3862	-0,0032	0,2551	0,1378	-0,0363	0,0726	0,4969
-0,3246	-0,2392	-0,1750	-0,2677	0,0992	0,0846	0,1682	0,4908

9 lentelė. Apmokyto termofikacinio vandens aušyklės AANT neuronų svoriai

W2								
-0,1490	0,1663	-0,3526	-0,2576	0,4767	-0,0678	0,2659	0,5421	-0,3257
0,4192	-0,2840	0,1348	0,3420	0,3379	-0,2400	-0,0913	-0,2539	-0,4675
-0,1752	0,3485	-0,3821	-0,5397	0,1472	0,4077	-0,2457	0,6982	0,1783

10 lentelė. Apmokyto termofikacinio vandens aušyklės AANT neuronų svoriai

W3		
1,7448	-0,8652	-3,0327
2,0859	-1,3151	-1,0324

-0,4171	0,9141	0,8261
-0,0527	-0,6575	0,1798
-0,840	2,1172	2,0065
0,6636	0,7036	-0,2037
-0,5578	-0,2401	0,5922
0,1949	-0,1514	0,7321

**11 lentelė.** Apmokyto termofikacinio vandens aušyklės AANT neuronų svoriai

W4							
0,9153	0,5779	-0,7894	-0,0118	0,3295	-0,9372	-0,4734	-0,9017
1,1218	0,6563	-0,1214	-0,4075	0,1264	-1,1657	0,0627	-0,8836
-0,2054	0,2551	-0,6456	1,5536	0,4378	-2,1152	1,3576	0,5084
0,5890	-0,4044	-0,5138	-0,4884	-1,1399	0,1599	0,37162	-0,3736
1,4629	0,6401	-0,1055	0,2613	0,6585	-1,2869	0,1008	-0,9379
-0,9041	-3,2354	0,5371	-0,5516	-2,3472	0,7446	1,1601	-0,0963
-2,2174	0,0786	-1,7505	-1,2391	-2,4504	1,6080	-1,4243	-1,3479
0,9153	0,5779	-0,7894	-0,0118	0,3295	-0,9372	-0,4734	-0,9017

**6 priedas. Kondensacinio ekonomizerio AANT apmokymo duomenų rinkinys**

**12 lentelė.** Kondensacinio ekonomizerio AANT apmokymo duomenų rinkinys

<b>10NDA5 0CQ901 XQ01</b>	<b>10NDA5 0CF901 XQ01</b>	<b>10NDA2 0CT002 XQ01</b>	<b>10NDA5 0CT001 XQ01</b>	<b>10HTA3 0CT001 XQ01</b>	<b>10HNA30C F001ZQ01</b>	<b>10HN A30CT 001XQ 02</b>
<b>MW</b>	<b>kg/s</b>	<b>°C</b>	<b>°C</b>	<b>°C</b>	<b>Nm<sup>3</sup>/h</b>	<b>°C</b>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
5,1	176,5	39,1	46	155,8	80977,992	42,41
3,5	174,3	40,4	45,2	154,7	84537,180	41,78
4,3	179,8	40	45,9	155,6	84268,578	41,78
4,4	183,3	40,1	45,9	158	120993,43	41,68
6,5	229	40,5	47,2	160,4	127927,18	40,43
5,7	207,8	40,5	47,1	161,1	123974,414	39,81
6,5	218,4	40,1	47,3	162,8	124642,781	39,81
6,3	229,7	40,2	46,9	161,6	123951,359	39,4
6,3	216,3	39,9	46,9	160,8	124248,852	39,4
6,4	219,4	39,6	46,5	160,4	125338,477	39,29
6,2	213,9	39,5	46,3	160,4	125479,633	38,98
6,1	218,2	39,9	46,5	161,1	125255,484	39,19
6,3	225,9	39,9	46,5	160,8	126778,461	39,19
6,4	225,3	39,7	46,5	161	126574,906	39,19
6,4	225,7	39,7	46,5	161,2	126406,914	39,19
6,1	226,6	39,8	46,3	160,7	126400,656	38,98
5,6	222	40,4	46,5	161,1	125981,68	39,29
5,9	227	40,5	46,8	161,1	126619,914	39,4
6	226,9	40,5	46,8	160,7	126338,586	39,4
6	227,9	40,5	46,8	161,1	126040,523	39,6
6,2	232,2	40,4	46,9	161	125858,547	39,6
6,1	227,1	40,3	46,8	160,7	125591,516	39,6
6,2	229,4	40,2	46,8	161,1	124715,625	39,4

6,3	231,8	40,1	46,5	160,5	123064,383	39,29
5,4	203,3	40,3	46,5	160,2	121523,203	39,29
5,6	198,8	39,8	46,8	159,7	119439,813	39,4
5,3	195,4	39,9	46,5	159,2	119670,57	39,6
5	193	40,1	46,4	159,4	119544,961	39,6
6	208,2	40	46,8	160,1	122391,156	39,6
6,3	228,6	39,9	46,4	161,1	131025,719	39,6
6,8	259,7	40,2	46,5	162,9	124511,781	39,29
7,1	266,5	40,2	46,5	163,5	127634,422	38,98
7,7	294,2	40,3	46,6	163,8	133794,078	39,19
8,1	313,1	40,5	46,9	163,8	137673,797	39,29
8,3	321,3	40,5	46,8	164,2	135836,453	39,19
8,1	322,1	40,7	46,9	163,9	138230,031	39,4
8,4	306,7	40,5	47,1	163,2	134941,109	39,6
8,3	311,9	40,4	46,8	162,7	135223,484	39,6
8,8	294,7	40,3	47,4	162,9	141099,797	39,71
8,1	299	40,4	47	160,7	139259,281	39,19
8,2	299,1	40,5	47,2	157,5	141002,891	39,29
7,8	293,9	40,7	47,2	155,6	143126,969	39,19
6,8	214,6	40,7	48,3	158,7	171660,078	40,43
9,1	264,5	40,7	49	160	164844,563	38,98
8,6	265,3	40,8	48,6	160,1	170576,797	38,57
10	298,6	40,9	49	162,2	172503,328	38,88
9,7	327,5	41,3	48,5	161,4	158335,422	37,63
11,1	282,1	41,2	50,8	162,3	151881,359	38,46
12,8	331,2	41,7	51	160,6	148359,563	37,84
10,6	229,4	41,3	52,2	159,6	141356,813	38,98
11,6	266	41	51,6	157,2	139818,703	37,74
12,2	287,9	41,5	51,8	157,1	137386,625	38,15
12,4	280,8	41,4	52,1	158,7	142385,375	38,77
11,7	275,2	41,6	51,9	158,3	139634,969	38,46
11,7	266,3	41,7	52,2	158,7	141416,281	38,98
12,9	297,3	41,9	52,3	158,5	136732,688	38,57
13,3	306,2	42,1	52,6	159,9	138201,719	38,98
12,9	301,2	42,1	52,5	159,9	141499,672	38,88
13,3	315,8	42,1	52,3	159	137311,125	38,57
13,2	310,1	42,1	52,4	159,7	141900,031	38,88
13,5	314,5	42,1	52,5	159,5	141330,391	38,77
13,9	310,4	42,1	53,1	159,3	148983,563	39,4
15,2	327,4	42,1	53,3	159,4	147011,156	39,4
16,9	364,6	42,1	53,4	160	146163,031	38,98
15,8	354,7	42	52,8	161,1	149234,578	38,98
15,6	354,5	42	52,5	160,5	148623,688	38,77
14,3	349,6	42,1	52,1	159,8	153420,406	38,57
15	363	42	52	159,6	153476,906	38,57
14,8	359	41,9	51,9	159,5	151396,344	38,46
14,8	357,6	41,9	51,9	159	151145,766	38,25
15,7	368,8	42	52,3	159,5	157269,953	38,88



16	383,1	42	52,1	159,9	157531,016	38,77
15,4	381,6	42,1	51,9	160,8	161222,172	38,77
16,1	388,1	42	52	160,2	158521	38,77
14,3	347	42,1	52	159,6	158379,234	38,88
14,8	354,4	42,1	52,1	159,4	159223,875	38,98
15,4	355,8	42	52,4	160	158235,625	38,98
15,7	354,2	42	52,6	159,9	157834,359	38,98
17,9	390,5	41,9	52,7	160,1	156876,828	35,14
18,8	418,2	41,8	52,6	160,9	157815,516	33,79
19,6	422,9	41,8	52,9	161,6	165436,188	34,52
19,7	435,5	41,8	52,7	161,4	159985,641	33,79
20,7	434,9	41,8	53,2	163	176639,453	35,24
19,8	446,1	41,8	52,5	162,8	165491,469	34,1
19	435,6	41,7	52,3	161,3	162904,906	34
19,3	434,3	41,8	52,5	160,4	161999,531	34,21
18,1	427,7	41,7	51,9	159,3	159403,672	33,79
16,9	396,2	41,6	51,8	159,5	164415,031	34,1
16,7	381,7	41,3	51,9	159,1	160140,313	33,58
18,1	403,4	41,2	52,2	159,3	163265,797	34
17,1	374,4	41,3	52,3	160,1	163437,016	33,58
20,8	430,6	41,3	52,9	159,8	162088,719	34,1
22,2	439,4	41,1	53,2	161,5	166378,156	34,21
20,6	440	41	52,3	161,4	163498,375	33,38
20,7	439,8	40,9	52,3	160,8	165317,563	33,27
19,6	429,2	40,8	51,8	162,1	166669,734	32,96
19	429,8	40,8	51,5	161,8	164782,234	32,86
18,6	423,4	40,7	51,3	160,1	163544,453	32,55
20,6	444,1	40,7	51,8	160,5	165107,688	32,96
20,2	442,1	40,7	51,7	160,9	171244,609	33,27
19,3	440,1	40,7	51,3	161,1	170953,25	32,86
19,2	439,4	40,5	51,1	161,2	167448,047	32,44
18	430,6	40,5	50,5	160,1	161260,75	31,92
17,4	422,5	40,5	50,5	159,8	163008,844	32,03
16,6	405,7	40,7	50,5	159,4	163618,641	32,13
19,3	424,1	40,7	51,6	159,8	165689	32,86
18,3	417,2	40,6	51,1	158,8	163214,813	32,13
19,1	400,6	40,6	52,1	160,2	162367,453	32,86
19,6	421,5	40,5	51,7	159,4	163084,938	32,44
19,1	437,7	40,7	51,2	159,7	164956,766	32,34
19,5	436,5	40,8	51,5	161,2	165327,734	32,34
19,9	432,2	40,6	51,8	161,3	161679,938	32,34
20	431,1	40,6	51,9	160,7	165720,953	32,44
20,8	434,3	40,6	52,1	160,1	165084,313	32,75
21,9	447,8	40,6	52,3	160,7	161197,531	32,55
22,8	456,4	40,4	52,4	160,9	160805,859	32,55
22,7	453,9	40,3	52,4	161,5	162249,875	32,44
21,9	452,7	40,3	52	160,7	161074,375	32,34
21,8	457,9	40,2	51,7	160,2	160234,375	32,13

19,8	424,1	40,2	51,4	160,9	163874,234	32,44
20,6	426,2	40,2	51,8	161,4	166728,016	32,75
20,2	417,1	40,1	51,8	160,5	170021,766	32,86
20,2	434,7	40,3	51,6	159,4	170084,313	32,75
19,7	435,1	40,4	51,4	159,8	165096,531	32,55
19,7	438,4	40,4	51,3	160,7	164242,219	32,34
19,3	439,9	40,3	50,9	161	162415,094	32,03
19,7	445,3	40,2	51	161,2	164603,656	32,44
17,8	428	40,2	50,2	160,7	162955,641	31,61
18,8	426,5	40,2	50,9	161,5	166063,094	32,34
19,2	428,6	40,1	51	160,5	164539,578	32,13
20,2	435,4	40,3	51,5	160,7	166837,484	32,75
21,9	442	40,2	52,1	162	166432,375	32,75
21,3	438	40,2	52	161,1	164706,203	32,44
21,6	442,9	40,2	52	161	165468,641	32,75
22,1	444,2	40,1	52,2	161	163889,828	32,75
22	447,3	40,1	52,1	161,1	165060,859	32,44
21,1	435,5	40,1	51,8	161	163827,141	32,34
20,6	436,4	40,1	51,6	160,9	162016,125	32,13
19,9	429,8	40,2	51,4	160,1	164807,844	32,03
19,5	428,5	40,1	51,2	160,3	163356,281	32,03

## 7 priedas. Apmokyto kondensacinio ekonomazerio AANT neuronų svoriai

13 lentelė. Apmokyto kondensacinio ekonomazerio AANT neuronų svoriai

W1							
0,2814	0,0014	0,0613	0,0471	-0,1998	-0,1948	0,1702	-0,5752
0,1032	0,3477	-0,2150	-0,1211	-0,2977	0,0525	-0,3506	0,2519
-0,1937	-0,3706	-0,3339	-0,4486	0,0810	-0,2897	-0,4580	-0,7563
0,4747	-0,3452	0,3954	0,1565	-0,1697	0,0969	-0,2484	0,3728
-0,1135	0,0506	0,4510	0,1014	0,1959	-0,0012	0,1336	0,5537
-0,1783	-0,5688	0,1663	0,1434	0,2284	-0,1881	0,5123	0,2897
-0,0262	0,4939	0,1557	-0,0648	0,2417	0,2971	-0,3025	0,0966
0,1642	-0,2427	0,0140	-0,1152	-0,2438	0,1297	0,0976	0,4335

14 lentelė. Apmokyto kondensacinio ekonomazerio AANT neuronų svoriai

W2								
-0,4188	-0,2312	-0,4396	0,0248	-0,0368	-0,4437	0,0153	-0,2590	-0,5743
-0,3917	0,3493	0,2649	-0,3341	-0,2707	0,334	-0,1568	-0,0013	0,2684
-0,4857	-0,2813	0,5235	-0,2421	0,6952	0,2375	0,1889	-0,4874	-0,2860

15 lentelė. Apmokyto kondensacinio ekonomazerio AANT neuronų svoriai

W3		
-3,3114	-1,1212	0,7195
0,6371	0,2149	-0,2522
1,0818	0,6408	-0,4282
0,5342	-0,6884	-0,3048
-0,5786	-1,1523	0,2864
-1,1605	-0,3726	0,0845

1,3319	1,4873	0,0552
-0,0535	0,0983	-0,5629

**16 lentelė.** Apmokyto kondensacinio ekonomizerio AANT neuronų svoriai

<b>W4</b>							
-0,2942	-0,5055	0,2095	1,1205	1,1010	-0,2397	0,7632	0,4708
-0,0902	0,2859	0,2694	0,8606	1,1493	-0,4496	0,8564	-0,0119
0,7247	0,9115	0,4136	0,6691	2,0255	0,5577	-0,1071	-0,4055
0,2579	0,7218	0,1710	1,2639	2,0733	0,7579	1,0853	0,8929
0,2837	-0,2664	-0,5686	0,6708	1,6608	-1,7820	2,2989	-2,0685
2,6036	0,8337	2,2549	1,4754	0,4676	-2,2387	1,0252	-0,4503
0,5223	-0,3219	-0,2972	-1,1810	0,3994	-0,4075	-0,3793	0,4625
-0,2942	-0,5055	0,2095	1,1205	1,1010	-0,2397	0,7632	0,4708