

VILNIAUS PEDAGOGINIS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Pavel Stefanovič

**Saviorganizuojančių neuroninių tinklų (SOM) sistemų
lyginamoji analizė**

Magistro darbas

Darbo vadovas: dr. Olga Kurasova

Vilnius, 2010

Turinys

1 Įvadas.....	3
2 Dirbtiniai neuroniniai tinklai	5
2.1. Biologinio neurono modelis	5
2.2. Dirbtinio neurono modelis.....	6
2.3. Dirbtinių neuroninių tinklų mokymas	7
2.4. Saviorganizuojantys neuroniniai tinklai	8
2.5. Saviorganizuojančio tinklo mokymas	9
2.6. Kvantavimo ir topografinė paklaidos	10
3 Saviorganizuojančių tinklų sistemos	11
3.1. Sistema <i>NeNet</i>	11
3.2. Sistema <i>SOM-Toolbox</i>	13
3.3. Sistema <i>Databionic ESOM</i>	14
3.4. Sistema <i>Viscovery SOMine</i>	16
3.5. Neuroninio tinklo paketo Matlab'e įrankiai SOM tinklams	20
4 SOM sistemų eksperimentinis palyginimas	24
4.1. Tyrimo duomenys.....	24
4.2. Gautų žemėlapių analizė.....	25
4.3. Topografinės ir kvantavimo paklaidos sistemose.....	29
5 Nauja saviorganizuojančių tinklų sistema	31
5.1. Sistemos apmokymo algoritmas.....	32
5.2. Sistemos grafinė sąsaja	33
5.3. SOM vizualizavimas naujoje sistemoje.....	34
5.4. Eksperimentinio tyrimo rezultatai	36
5.4.1. Kvantavimo ir topografinės paklaidų priklausomybė nuo epochų skaičiaus	36
5.4.2. Mokymo taisyklės modifikavimas	42
6 Išvados	46
7 Literatūros sąrašas	48

1 Įvadas

Šiuo metu įvairiose srityse plačiai taikomi dirbtiniai neuroniniai tinklai, kurie yra naudojami klasifikavimo, klasterizavimo, prognozavimo, vaizdų, garsų atpažinimo ir kitiems įvairiems uždaviniams spręsti. Neuroniniai tinklai gali padėti įvairiose srityse, kurios yra aktualios žmonių gyvenime: teisėsaugos, technologijoje, ekologijos, medicinoje, chemijoje ir kitose srityse. Saviorganizuojantis neuroninis tinklas (SOM) yra vienas iš kelių egzistuojančių neuroninių tinklų tipų. SOM tinklai dažnai naudojami medicinoje. Jų pagalba yra analizuojami krūties vėžio, plaučių vėžio, širdies ligų ir daugelis kitų medicininio pobūdžio duomenų. Tobulėjant kompiuteriams ir didėjant duomenų kiekiams galima gauti daug tikslesnius rezultatus, tad neuroninių tinklų nagrinėjimas tampa vis svarbesne informatikos mokslo šaka.

Dažniausiai skirtingiems uždaviniams naudojami skirtingo tipo neuroniniai tinklai. Kad galėtume naudoti neuroninius tinklus praktiniams uždaviniams spręsti, būtinos sistemos, turinčios patrauklias grafines sąsajas, įvairių parametrų nustatymo galimybes, lengvą valdymą ir pan. Sistemų yra įvairių, vienoms išmokti reikia sugaišti be galo daug laiko, kitoms vos keletą minučių, tačiau jos gali būti su ribotomis galimybėmis. Todėl svarbu žinoti, kokią sistemą geriausiai naudoti vienu ar kitu atveju, kad gautume geriausius rezultatus ir nebūtų eikvojimas veltui laikas. Šiuo metu viena populiariausių sistemų yra SOM-Toolbox, kuri sukurta Matlab aplinkoje. Dažnai naudojama, pastoviai tobulinama Viscovery SOMine sistema.

Pirmoje darbo dalyje bus glaustai pateikta dirbtinių neuroninių tinklų teorija. Pateiktas biologinis ir dirbtinis neurono modeliai, jų ypatumai; dirbtinių neuroninių tinklų apmokymas, skirtingi mokymo algoritmai. Išanalizuoti saviorganizuojančių neuroninių tinklų veikimo principai. Pateikta teorija apie saviorganizuojančių neuroninius tinklus: tinklo apmokymas, apmokyto tinklo kokybės nustatymas. Paašškintos pagrindinės sąvokos: topografinė ir kvantavimo paklaidos, neuronas nugalėtojas, bei daugelis kitų reikalingų sąvokų.

Antroje darbo dalyje analizuojami vieno tipo neuroniniai tinklai – tai saviorganizuojantys neuroniniai tinklai, dar vadinami žemėlapiiais (angl. *self-organizing maps*, SOM) ir juos realizuojančios sistemos. Nagrinėtos šios sistemos: SOM-Toolbox, NeNet, Databionic ESOM, Viscovery SOMine, Matlab funkcijos „nntool“ ir „nctool“, o taip pat naują vizualizavimo būdą turinti sukurta nauja SOM sistema. Šios sistemos pasirinktos todėl, kad yra dažniausiai naudojamos, bei turi gerą ir paprastą valdymo sistemą. Pagrindinis saviorganizuojančių neuroninių tinklų tikslas yra duomenų klasterizavimas ir vizualizavimas. Vertinant sistemas būtina atsižvelgti į tai, kaip duomenys juose yra klasterizuojami ir pateikiami SOM žemėlapiuose. Sistemos skiriasi viena nuo kitos tiek grafine sąsaja,

tiesiogiai įvairiomis tinklo mokymo funkcijomis, duomenų pateikimu, rezultatų vizualizavimo būdu ir kt. Natūralu, jog kiekviena sistema turi savo teigiamų ir neigiamų savybių, darbe bus pateiktas kiekvienos sistemos aprašymas, išvardintos įvairios galimybės, bei jos išskirtinumas.

Taip pat pateiktas naujai sukurtos SOM tinklų sistemos aprašymas, kurioje tinklas apmokomas kitaip, negu kitose jau esančiose sistemose, t. y. realizuotas kitas nei įprasta mokymo algoritmas, panaudota kita kaimynystės funkcija. Ši sistema yra sukurta Matlab programos aplinkoje, turi nesudėtingą grafinę vartotojo sąsają.

Tyrimo tikslas – palyginti saviorganizuojančių neuroninių tinklų sistemas įvairiais aspektais, sukurti sistemą, kurioje realizuotas naujas vizualizavimo būdas.

Uždaviniai:

1. Išanalizuoti saviorganizuojančio neuroninio tinklo veikimo principus;
2. Apžvelgti esamas saviorganizuojančių neuroninių tinklų sistemas;
3. Atlikti pasirinktų SOM sistemų lyginamąją analizę:
 - išnagrinėti skirtingus sistemų duomenų vizualizavimo būdus;
 - išanalizuoti gautas topografines ir kvantavimo paklaidas;
4. Tirti tinklo apmokymo algoritmų parametrų įtaką gaunamiems rezultatams;
5. Sukurti SOM sistemą, kurioje realizuotas naujas vizualizavimo būdas.

Dalis tyrimų rezultatų pristatyti XIV tarptautinėje kompiuterininkų konferencijoje, vykusioje renginio „Kompiuterininkų dienos 2009“ metu Kaune, 2009 m. rugsėjo 25-26 d. Skaitytas pranešimas „Saviorganizuojančių neuroninių tinklų sistemų lyginamoji analizė“.

Tyrimų rezultatai paskelbti mokslinėje publikacijoje:

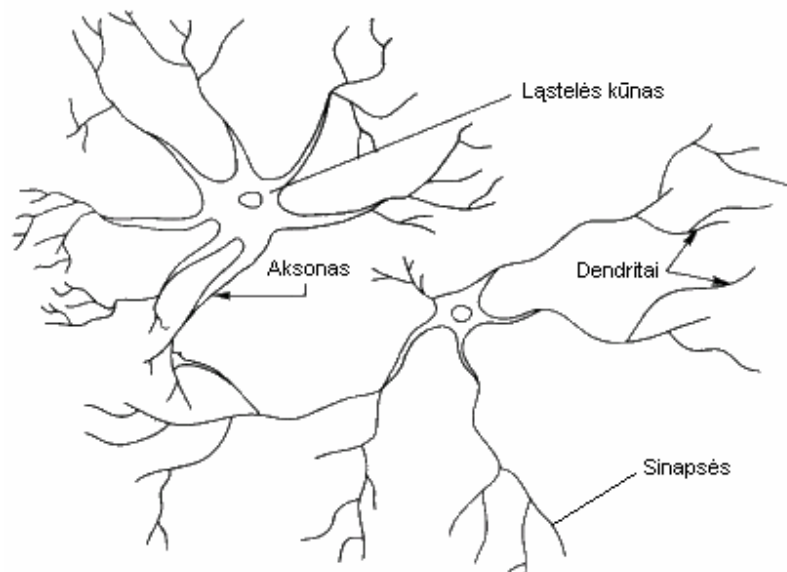
P. Stefanovič, O. Kurasova. Saviorganizuojančių neuroninių tinklų sistemų lyginamoji analizė // Informacijos mokslai. ISSN 1392-0561. T. 50 (2009), p. 334–339.

2 Dirbtiniai neuroniniai tinklai

Dirbtinius neuroninius tinklus pirmieji kūrė ir tyrinėjo McCulloch ir Pitts 1943 metais. Savo darbuose jie suformulavo pagrindinius dirbtinių neuroninių tinklų ir neuroninių tinklų kūrimo principus. Taip pat didelis žingsnis neuroninių tinklų kūrime yra padarytas Rosenblatt 1962 metais, kuris sukūrė vienasluoksnį neuroninį tinklą, dar vadinamu Perceptrono modeliu. Vėliau Minsky ir Parret įrodė matematiniais skaičiavimais, jog neuroninių tinklai negali išspręsti ir paprastos užduoties, kaip „XOR“, todėl ilgą laiką neuroniniai tinklai buvo pamiršti. Tik keletas mokslininkų kaip Kohonen, Anderson ir kiti tęsė mokslinius tyrimus šia kryptimi. Kaip paaiškėjo vėliau, Minsko išvados apie neuroninius tinklus buvo labai pesimistiškos, nes užduoties, kurias jis apibūdino kaip neišsprendžiamas, šiandieną neuroninių tinklų pagalbą sprendžiami labai lengvai (MicroStat analytics Ltd, 2010).

2.1. Biologinio neurono modelis

Prieš pradėdant nagrinėti dirbtinius neuroninius tinklus reikia suprasti biologinių neuroninių tinklų modelį. Kiekvienas neuronas turi ląstelės kūną – somą, įėjimus – dendritus ir išėjimus – aksonus (Stergius, Siganos). Neuronai tarpusavyje jungiasi sinapsėmis (2.1 pav.)

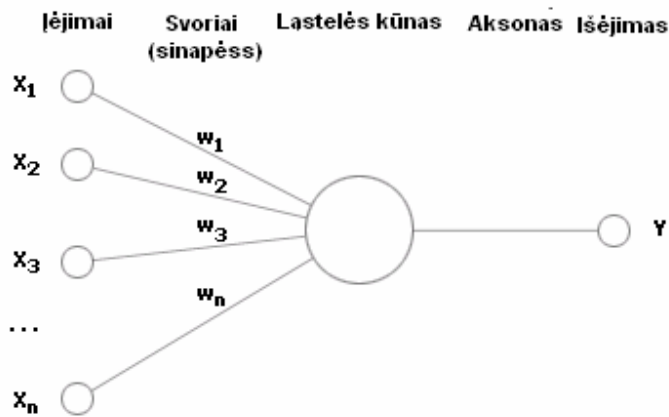


2.1 pav. Biologinis neurono modelis

Dendritai perduoda impulsus į neuronus, o aksonais impulsai išeina iš neurono. Vienas neuronas vienu metu jungiasi su daug kitų neuronų, dėl to jis gali sudaryti iki 100 000 sinapsių vienu metu.

2.2. Dirbtinio neurono modelis

Dirbtinį neuroninį tinklą sudaro neuronai, susijungę vienas su kitu daugeliu jungčių. Kiekviename neurono įėjime x_1, x_2, \dots, x_n yra paduodamas impulsas, kuris išeidamas iš neurono atitinkamai dauginamas iš svorio w_1, w_2, \dots, w_n ir skaičiuojama sandaugų suma $\sum_{k=1}^n x_k w_k$. Kiekvienas įėjimas turi savo svorį. Pats paprasčiausias dirbtinio neurono pavyzdys yra McCulloch ir Pitts dirbtinio neurono modelis (2.2 pav.)



2.2 pav. McCulloch ir Pitts dirbtinis neuronas

Šis dirbtinis neuronas modelis apibūdinamas taip:

1) Neuronas gali turėti be galo daug įėjimų $x_1, x_2, \dots, x_n, n = 1, \dots, k$ ir kiekvienas iš jų turi savo svorį (perdavimo koeficientą) $w_1, w_2, \dots, w_n, n = 1, \dots, k$;

2) Skaičiuojamas sužadinimo signalas: $a = x_1 w_1 + x_2 w_2 + \dots + x_n w_n = \sum_{k=1}^n x_k w_k$;

3) Neuroną apibūdina funkcija: $y = f(a) = f\left(\sum_{k=1}^n x_k w_k\right)$, kur $f(a) = \begin{cases} 1, & \text{jeigu } a \geq w_0 \\ 0, & \text{jeigu } a < w_0 \end{cases}$, o w_0 yra

slenksčio reikšmė. Kai sužadinimo signalas yra mažesnis už slenksčio w_0 reikšmę, funkcijos $f(a)$ reikšmė yra lygi 0, priešingu atveju 1.

Dirbtinio neurono modelyje gali būti naudojama ne tik slenksstinė funkcija, bet ir kitos neurono aktyvacijos funkcijos:

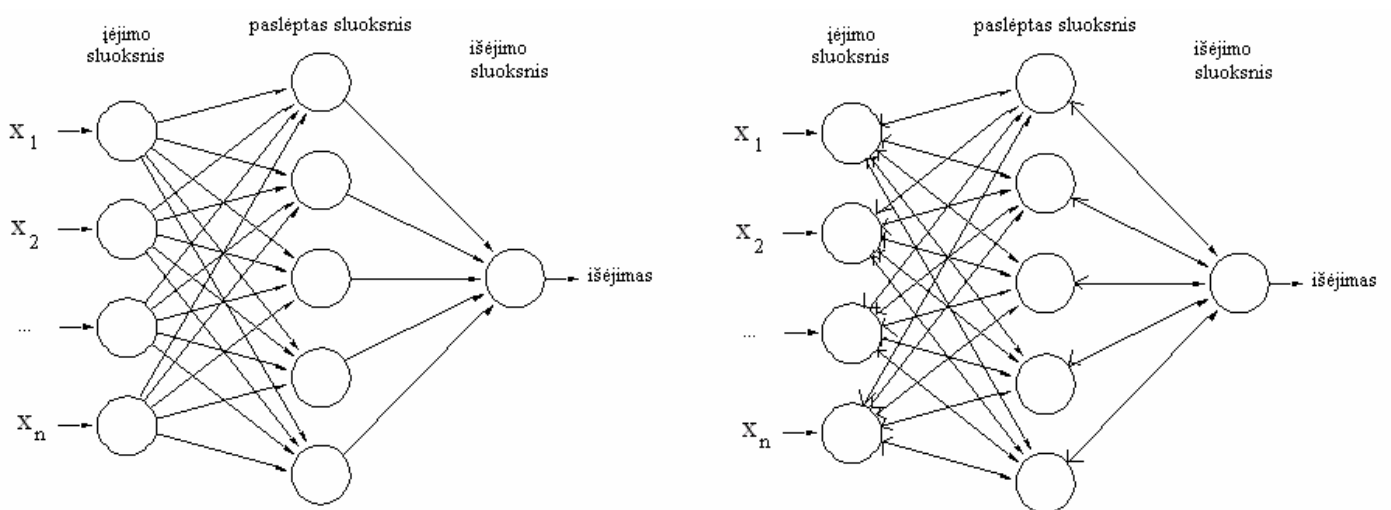
- Sigmoidinė funkcija: $f(a) = \frac{1}{1 + e^{-a}}$;

- Hiperbolinis tangentas: $f(a) = \tanh a = \frac{e^a - e^{-a}}{e^a + e^{-a}}$

- kitos.

Taigi biologinio ir dirbtinio neurono modelis yra labai panašūs (2.1 pav., 2.2 pav.), kadangi turi tuos pačius elementus ir labai panašų veikimo principą. Dirbtinį neuroninį tinklą kaip ir biologinį sudaro daug neuronų, sujungtų vienas su kitu. Pagal jungimo konstrukciją neuroniniai tinklai skirstomi į dvi kategorijas (2.3 pav.) (Smith L., 2008):

- 1 tiesioginio sklidimo (*feedforward*) tinklai;
- 2 grįžtamojo ryšio (*feedback*) tinklai.



2.3 pav. Tiesioginio sklidimo (kairėje) ir grįžtamojo ryšio (dešinėje) tinklai

2.3. Dirbtinių neuroninių tinklų mokymas

Šiuo metu nėra sukurto bendro dirbtinio neuroninio tinklo mokymo apibrėžimo, bet jis traktuojamas kaip tinklo struktūros ir jungčių svorių keitimo uždavinys, siekiant, kad tinklas galėtų atlikti jam skirtą užduotį.

Yra trys pagrindinės neuroninio tinklo mokymo paradigmos:

- 1 mokymo su mokytoju algoritmai (*supervised learning*);
- 2 mokymo be mokytojo algoritmai (*unsupervised learning*);
- 3 hibridinis mokymas (*hybrid learning*).

Mokymas su mokytoju algoritmas taikomas tada, kai norime analizuoti duomenis, kurių trokštamoms (*target*) reikšmės yra žinomos. Trokštamomis reikšmėmis gali būti klasių pavadinimai, prognostinės reikšmės ir kt. Mokymo tikslas – rasti tokius jungčių svorius, kad skirtumas tarp

neuroninio tinklo išėjimų ir trokštamų reikšmių būtų kiek galima mažesnis. Mokymas su mokytojo vyksta paprastai dvejais etapais:

- 1) Tinklas apmokomas naudojant mokymo duomenis su trokštamomis reikšmėmis;
- 2) Apmokytas tinklas yra testuojamas, į tinklą pateikiant duomenis, kurie nebuvo naudojami tinklui mokyti.

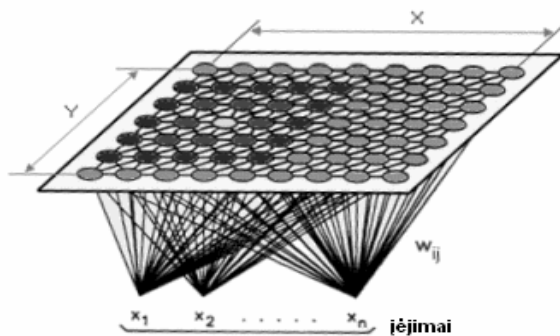
Mokymas be mokytojo algoritme tinklas yra apmokomas, kai trokštamoms reikšmėms nėra žinomos. Šis metodas taikomas tada, kai norime rasti turimų duomenų panašumus, juos suklasifikuoti ar pan.

Hibridinis mokymas taikomas naudojant tiek mokymą su mokytoju, tiek mokymą be mokytojo algoritmus. Dalis tinklo svorių nustatomi pagal mokymą su mokytoju, kita dalis gaunama iš mokymo be mokytojo.

2.4. Saviorganizuojantys neuroniniai tinklai

Sakykime, kad turime analizuojamų duomenų aibę $X = \{X_1, X_2, \dots, X_m\}$, kurią sudaro vektoriai X_1, X_2, \dots, X_m , charakterizuojantys tam tikrą objektų aibę, kuri apibūdinama bendrais parametrais x_1, x_2, \dots, x_n ; čia m – analizuojamų objektų (vektorių) skaičius, n – parametrų skaičius. Parametrų reikšmės $x_{i1}, x_{i2}, \dots, x_{in}$ ir yra vektoriaus $X_i, i \in \{1, \dots, m\}$ komponentės. Taigi, X_1, X_2, \dots, X_m yra n -mačiai vektoriai, kurie interpretuojami kaip taškai n -matėje erdvėje R^n ; čia n – erdvės matmenų skaičius. Iš tiesų turime analizuojamų duomenų aibės matricą $X = \{X_1, X_2, \dots, X_m\} = \{x_{ij}, i = 1, \dots, m, j = 1, \dots, n\}$, jos eilutės yra vektoriai $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$, $i = 1, \dots, m$; čia x_{ij} yra i -ojo vektoriaus j -oji komponentė.

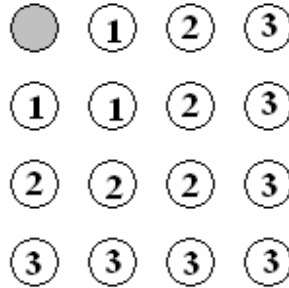
Saviorganizuojantis neuroninis tinklas yra neuronų, paprastai išdėstytų dvimačio tinklelio, dar vadinamo žemėlapiu arba lentele, mazguose, masyvas M (Kohonen, 2001). Stačiakampės topologijos atveju $M = \{M_{kl}, k = 1, \dots, r, l = 1, \dots, s\}$; čia r – tinklo eilučių skaičius, s – tinklo stulpelių skaičius, M_{kl} – vektorius, kurio matmenų skaičius n yra toks pat kaip mokymo duomenų vektorių. Dažniausiai yra analizuojami dvimačiai SOM tinklai (2.4 pav.), nors galimi ir didesnio matmenų skaičiaus tinklai.



2.4 pav. Dvimatis SOM tinklas

Nagrinėjant saviorganizuojančius tinklus yra naudojama sąvoka epocha. Epocha – tai mokymo proceso dalis, kurios metu visi mokymo aibės vektoriai nuo X_1 iki X_m (m –vektorių skaičius) po vieną kartą pateikiami į tinklą nuosekliai arba atsitiktine tvarka (Dzemyda, Kurasova, Žilinskas, 2008).

Turėdami SOM žemėlapyje atvaizduotus neuronus nugalėtojus, galime apibrėžti sąvoką „kaimynystės eilė“. Apie neuroną nugalėtoją esantys neuronai yra vadinami pirmos eilės kaimynais. Toliau esantys neuronai yra antros eilės kaimynai ir t. t. (2.5 pav.)



2.5 pav. Kaimynystės eilė neurono nugalėtojo (pilkas) atžvilgiu

Pats populiariausias SOM vizualizavimo būdas yra unifikuotos matricos vaizdavimas (U-matrica, *unified distance matrix*). U-matricą sudaro atstumai tarp kaimyninių SOM neuronų. Paprastai daugumoje sistemų jos vaizduojamos keliomis spalvomis. Jei vidutinis atstumas tarp kaimyninių neuronų yra mažas — naudojama šviesesnė spalva, jeigu didelis – tamsesnė (Dzemyda, Kurasova, 2002).

2.5. Saviorganizuojančio tinklo mokymas

Mokymo pradžioje neuronų (vektorių) M_{kl} komponentių pradinės reikšmės dažniausiai nustatomos atsitiktinai. Neuroniniam tinklui daug kartų pateikiama skirtingų objektų, nusakomų n -mačiais vektoriais X_1, X_2, \dots, X_m . Kiekviename mokymo žingsnyje (iteracijoje) vienas mokymo aibės vektorius $X_p \in \{X_1, X_2, \dots, X_m\}$ pateikiamas į tinklą. Vektorius X_p palyginamas su visais neuronais M_{kl} : dažniausiai skaičiuojamas Euklido atstumas $\|X_p - M_{kl}\|$ tarp šio vektoriaus X_p ir kiekvieno neurono M_{kl} . Randama, iki kurio neurono $M_c \in \{M_{kl}, k=1, \dots, r, l=1, \dots, s\}$ atstumas yra mažiausias; rastas neuronas M_c vadinamas neuronu (vektoriumi) nugalėtoju. Visų tinklo neuronų komponentės keičiamos naudojantis iteracine formule

$$M_{kl}(t+1) = M_{kl}(t) + h_{kl}^c(t)(X_p - M_{kl}(t)). \quad (1)$$

Šioje formulėje t yra iteracijos numeris, h_{kl}^c – kaimynystės funkcija. Baigus tinklo mokymą, mokymo ar kita, vadinamoji testavimo duomenų (vektorių) aibė pateikiama į tinklą, randamas kiekvieno vektoriaus vektorius nugalėtojas, jį atitinkančiame žemėlapyje užrašomas vektoriaus eilės numeris arba klasės, kuriai priklauso šis vektorius, pavadinimas, t. y. žemėlapis yra sužymimas.

2.6. Kvantavimo ir topografinė paklaidos

Kai tinklas išmokytas, būtina įvertinti jo kokybę. Tam dažniausia naudojamos dvi paklaidos: kvantavimo ir topografinė. Kvantavimo paklaida parodo, kaip tiksliai jau išmokyto tinklo neuronai prisiderina prie mokymo aibės vektorių. Kvantavimo paklaida E_{QE} – tai vidutinis atstumas tarp duomenų vektorių X_p ir jų vektorių nugalėtojų $M_{c(p)}$:

$$E_{QE} = \frac{1}{m} \sum_{p=1}^m \|X_p - M_{c(p)}\|. \quad (2)$$

Topografinė paklaida parodo, kaip gerai SOM tinklas išlaiko analizuojamų duomenų topografiją, t. y. tarpusavio išsidėstymą. Topografinė paklaida E_{TE} skaičiuojama pagal šią formulę:

$$E_{TE} = \frac{1}{m} \sum_{p=1}^m u(X_p). \quad (3)$$

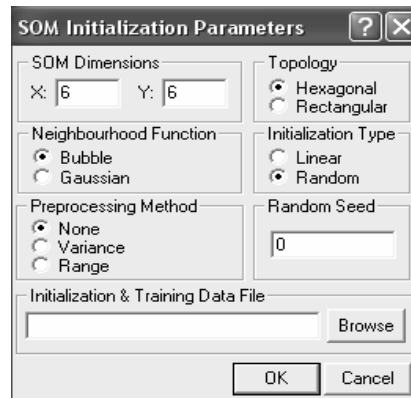
Jeigu SOM žemėlapyje vektoriaus X_p neuronas nugalėtojas yra šalia neurono, iki kurio atstumas nuo X_p yra mažiausias, neskaičiuojant iki neurono nugalėtojo, tai (3) formulėje $u(X_p) = 0$, priešingu atveju $u(X_p) = 1$.

3 Saviorganizuojančių tinklų sistemos

Šiuo metu internete galima rasti įvairių sistemų, kuriose realizuoti saviorganizuojantys neuroniniai tinklai. Darbe nagrinėjamos šios sistemos: „NeNet“, „Som-Toolbox“, „Databionic ESOM“, „Viscovery SomMine“ ir Matlab funkcijos „nntool“, „nctool“.

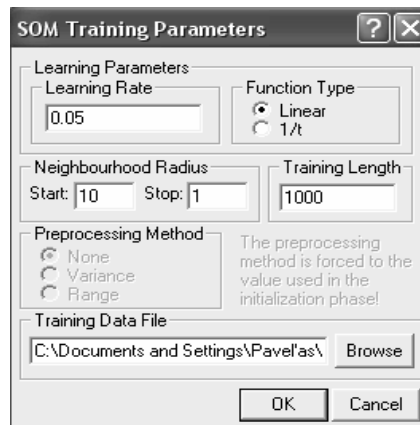
3.1. Sistema NeNet

Sistema *NeNet* yra pritaikyta *Windows* operacinei sistemai. Internete platinama tik demonstracinė versija, todėl ji turi apribojimų (Hassinen ir kt., 1999). Galima sukurti žemėlapi, kurio ir eilučių, ir stulpelių skaičius neviršija 10, nagrinėti vektorius, kurių matmenų skaičius neviršija 6 ir vektorių skaičius neviršija 2000. Programa turi savo grafinę sąsają, kurioje galima lengvai mokytis ir vizualizuoti saviorganizuojančius neuroninius tinklus (SOM). Žemėlapis kuriamas trimis etapais: inicijavimo (*init*), mokymo (*train*) ir testavimo (*test*). Žemėlapio inicijavimui skirtoje kortelėje (3.1 pav.) galima nurodyti SOM eilučių ir stulpelių skaičių (*SOM Dimension*), pasirinkti šešiakampio ar stačiakampio formos žemėlapi (*Topology*), nurodyti, kuriuo būdu (burbuliuko ar Gauso metodu) bus skaičiuojama kaimynystės funkcija (*Neighbourhood Function*), pasirinkti inicijavimo tipą – linijinį arba atsitiktinį (*Initialization Type*), atsitiktinių skaičių generatoriaus inicijavimo skaitmenį (*Random Seed*) ir įkelti mokymo duomenų failą (*Initialization & Training Data File*).



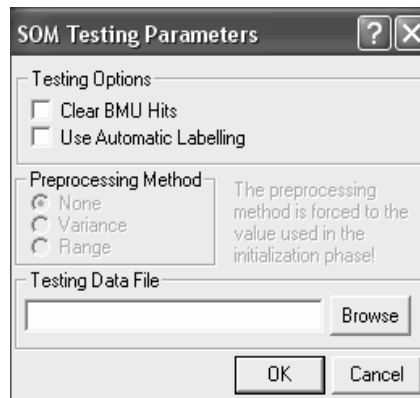
3.1 pav. Tinklo inicijavimo kortelė sistemoje NeNet

Tinklo mokymui skirtoje kortelėje (3.2 pav.) galima keisti įvairius mokymo parametrus: funkcijos tipą (*Function Type*), mokymo greitį (*Learning Rate*), kaimynystės spindulį (*Neighbourhood Radius*) ir mokymo trukmę (*Training Length*). Prieš testavimą nurodomas testuojamų duomenų failas (*Testing Data File*).



3.2 pav. Tinklo mokymo kortelė sistemoje NeNet

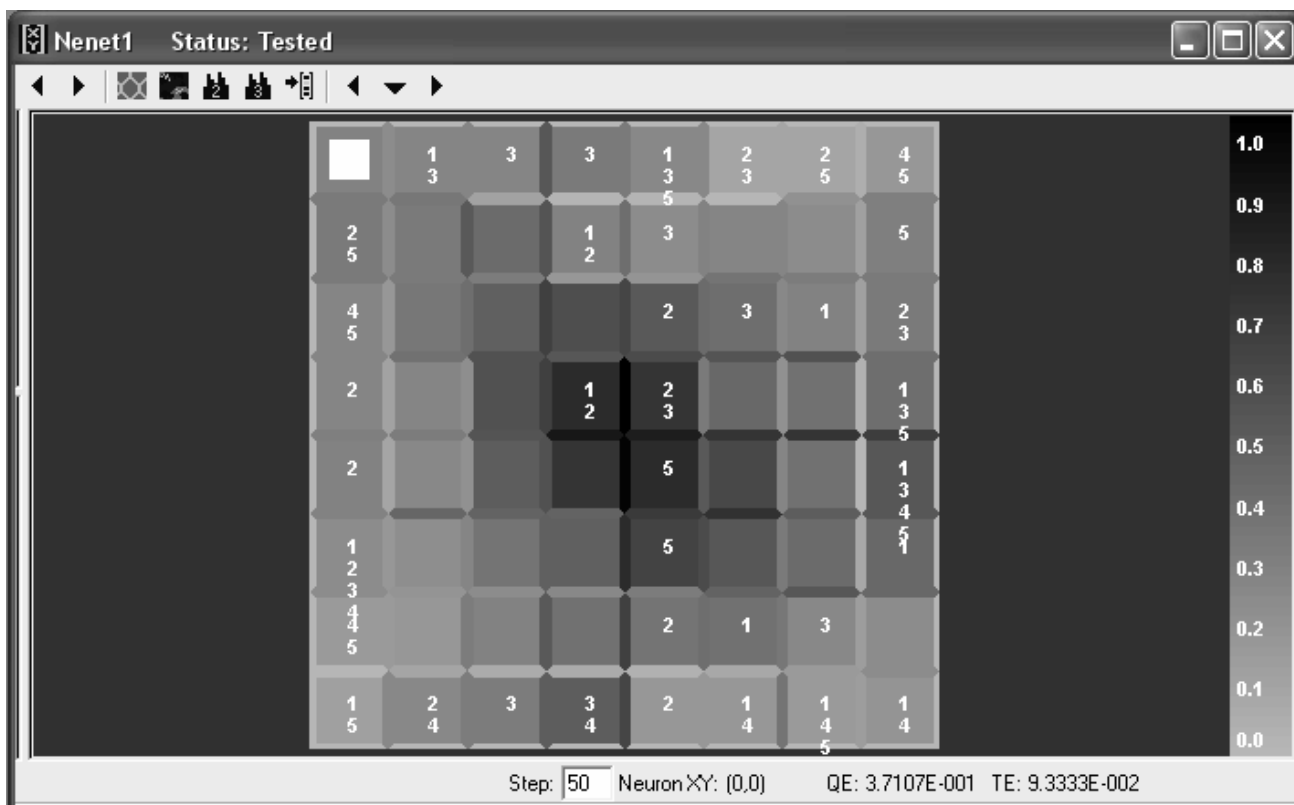
Testavimo kortelėje (3.3 pav.) keičiami testavimo parametrai (*Testing Options*), bei pasirenkamas testavimo failas, kuriame pateikti vektorių pavadinimai bus atvaizduoti SOM žemėlapyje.



3.3 pav. Tinklo testavimo kortelė sistemoje NeNet

Sistemai *NeNet* duomenys turi būti pateikti tekstiniame faile. Skaičius pirmoje eilutėje nurodo vektoriaus matmenų skaičių, po juo surašomi analizuojamus duomenis atitinkantys vektoriai. Paskutinis stulpelis skirtas vektorių pavadinimams nurodyti. Duomenų sveikoji dalis būtinai privalo būti atskirta tašku. Atlikus inicijavimą, apmokymą ir testavimą gaunamas SOM žemėlapis (3.4 pav.).

Gautame SOM žemėlapyje matome analizuojamų vektorių išsidėstymą pagal klases. Jei tarp neuronus atitinkančių langelių rėmelis yra tamsesnis, tai reiškia, kad į tuos neuronus pakliuvę vektoriai yra tolimesni. Langeliuose nurodomas tik vienas tos pačios klasės pavadinimas, tačiau tos pačios vektorių klasės narių gali būti ir daugiau tame pačiame langelyje. Kai kuriuose žemėlapio langeliuose nurodytos dvi, trys klasės, tai reiškia, jog tą patį neuroną atitinka kelių klasių vektoriai. Apačioje pateikiamos QE (kvantavimo) ir TE (topografinė) paklaidos, kurios parodo, kaip gerai buvo apmokytas tinklas.



3.4 pav. Sistema NeNet gautas SOM žemėlapis

3.2. Sistema *SOM-Toolbox*

Sistema *SOM-Toolbox* yra sukurta *Matlab* programa, todėl ją galima naudoti tik šioje programoje (Vesanto ir kt., 2005). *SOM-Toolbox* turi daugiau kaip 200 įvairių funkcijų, jos suskirstytos į rinkinius pagal veikimo pobūdį. Pagrindiniai funkcijų rinkiniai, skirti SOM žemėlapiui sukurti, yra šie: struktūros ir failai (*Struct Conversion and File I/O*), duomenų apdorojimas (*Data Preprocessing*), inicijavimo ir apmokymo funkcijos (*Initialization and Training Function*) ir vizualizavimas (*Visualization*). Toliau pateikiamos funkcijos, kuriomis galima inicijuoti, išmokyti ir pavaizduoti paprasčiausią SOM žemėlapi:

sd=som_read_data('filename') – duomenų nuskaitymas iš failo, 'filename' – failo pavadinimas, sd – gražinamas funkcijos parametras – duomenų struktūra.

sd=som_normalize(sd,'var') – duomenų normavimas, sd – pradiniai duomenys, nurodžius 'var' duomenys normuojami taip, kad dispersija būtų lygi 1.

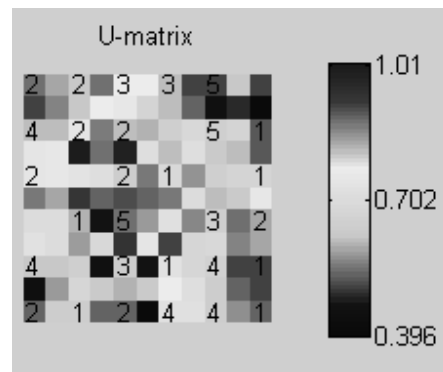
sm=som_make(sd) – žemėlapių sukūrimas ir mokymas, sd – perduoti pradiniai duomenys, sm – gražinamas funkcijos parametras – išmokytas žemėlapis.

sm=som_autolabel(sm,sd,'vote') – žemėlapis elementų žymėjimas, sd – perduoti pradiniai duomenys, sm – žemėlapis, kurį norime sužymėti, nurodžius 'vote' bus rodoma tik dominuojanti klasė viename žemėlapis langelyje.

map=som_show(sm) – žemėlapis vizualizavimas, sm – vizualizuojamas žemėlapis.

som_show_add('label',sm) – sužymėto žemėlapis vizualizavimo funkcija, sm – žemėlapis, kurį sužymime ir vizualizuojame.

Duomenys sistemai *SOM-Toolbox* pateikiami kaip ir sistemai *NeNet* tekstiniame faile. Vektorių matmenų skaičius gali būti nurodomas tekstinio failo pirmoje eilutėje arba vektorių nuskaitymo funkcijos argumente `som_read_data(filename, dimension)`. Duomenų faile paskutinis stulpelis skirtas vektorių pavadinimams. Nustačius įvairius parametrus SOM žemėlapis gali būti skirtingai pateikiamas. Vienas iš būdų yra unifikuotos matricos (*U-matrix*) pateikimas. (3.5 pav.)



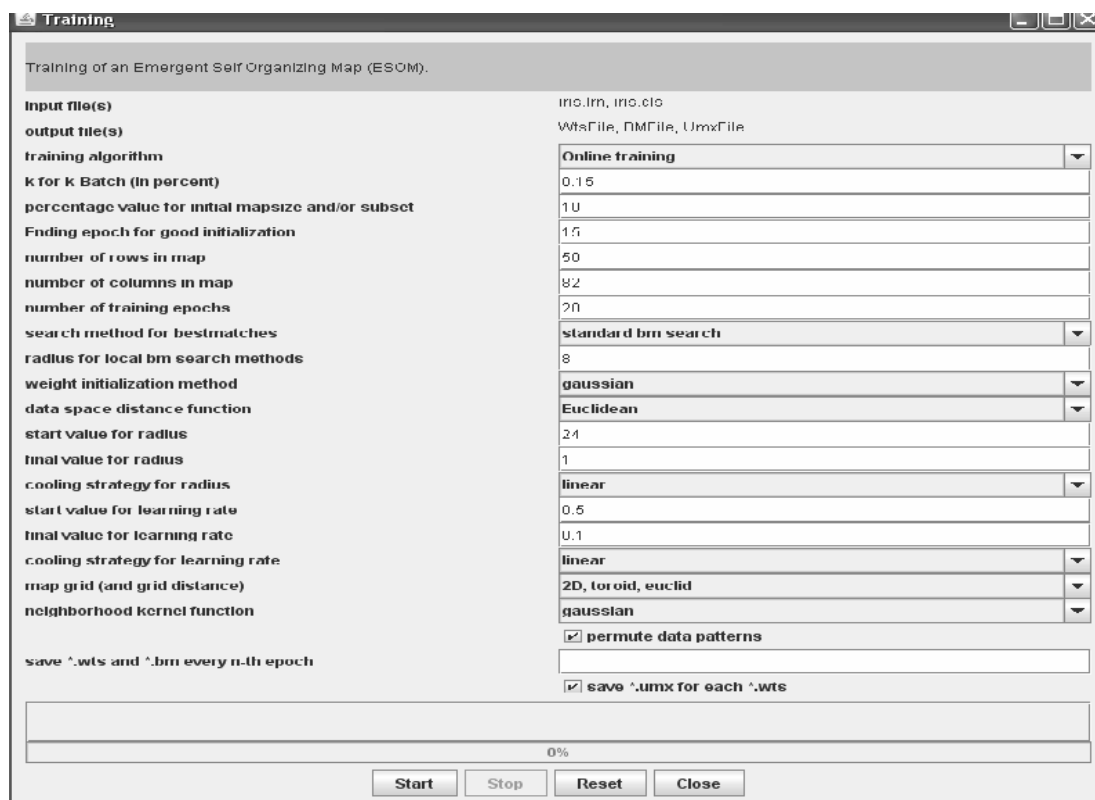
3.5 pav. Sistema *SOM-Toolbox* gautas SOM žemėlapis

Taip pat kaip ir sistemoje *NeNet* klases skiria šviesesni arba tamsesni langeliai, kurie žymi kaip vektoriai nutolę vienas nuo kito. Galima nustatyti, kad viename žemėlapis langelyje būtų arba vienas duomuo arba išvardinti visų į jį patekusių vektorių klasių pavadinimai, tačiau tuomet duomenys uždengtų vienas kitą ir sunku būtų suprasti gautą rezultatą. Kadangi *SOM-Toolbox* turi daug įvairių komandų, todėl žemėlapi įmanoma vizualizuoti įvairiais būdais. Atlikus tinklo apmokymą, Matlab programos aplinkoje pateikiamos kvantavimo ir topografinės paklaidos, gali būti išvesti kiti reikalingi duomenys.

3.3. Sistema *Databionic ESOM*

Sistema *Databionic ESOM* yra neseniai sukurta sistema ir yra skirta didelėms duomenų aibėms klasterizuoti ir vizualizuoti (Ultsch, Moerchen, 2005). Kaip ir sistema *NeNet* ji turi grafinę sąsają. *Databionic ESOM* yra nemokama sistema, todėl jokių apribojimų neturi. Norint sukurti SOM žemėlapi

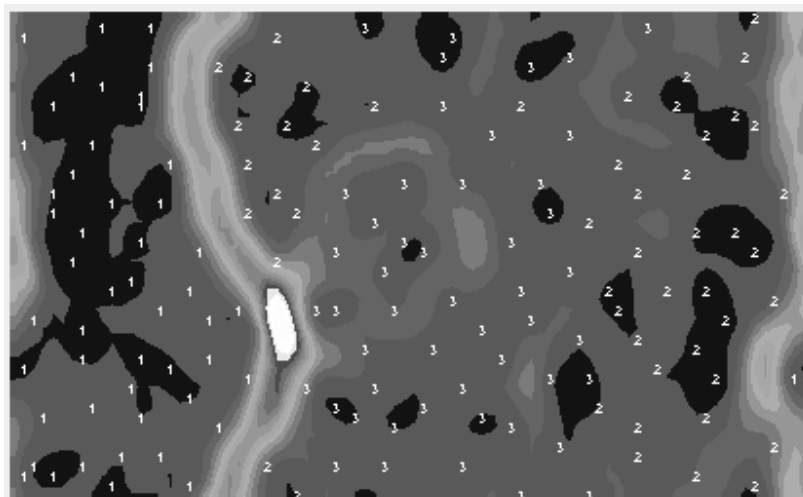
iš pradžių reikia paruošti du failus: mokymo failą ir vektorių klasių pavadinimų failą. Šiuos failus galima sukurti paprasčiausia užrašinės programa, mokymo duomenų failą išsaugoti prievardžiu *.lrn, o klasių pavadinimų failą prievardžiu *.cls. Pirmoje mokymo failo eilutėje rašomas vektorių skaičius, antroje – stulpelių skaičius plus vienas stulpelis indeksams, trečioje – skaičius 9 (jis yra unikalus, nekeičiamas) ir pagal tai, kiek duomenų stulpelių (parametrų), rašome 1 arba 0 (skaičius 1 reiškia, jog stulpelis nuskaitomas, skaičius 0 – jog stulpelis bus ignoruojamas), ketvirtoje – stulpelių pavadinimai. Vektorių pavadinimų faile pirmoje eilutėje rašomas vektorių skaičius, žemiau pateikiami visi vektoriai. Pirmas stulpelis kaip ir mokymo faile skirtas eilučių indeksams. Turint abu failus, vizualizavimas vyksta dviem etapais: mokymas ir žemėlapio sužymėjimas. Pirmame etape įkeliamas mokymo failas (*File/Load *.lrn*), ir žemėlapis yra mokomas (*Training*). Prieš mokymą galima keisti įvairius parametrus (3.6 pav.): žemėlapio dydį, žemėlapio formą, kaimynystės funkciją, skaičiavimo greitį ir daugelį kitų.



3.6 pav. Sistemos Databionic ESOM apmokymo kortelė

Antrame etape įkeliamas vektorių pavadinimų failas (*File/Load *.cls*). Kad klasės būtų sužymėtos gautame žemėlapyje, reikia kortelėje *Classes* uždėti varnelę prie *Draw letters*. Atlikus apmokymą gaunamas SOM žemėlapis (3.7 pav.)

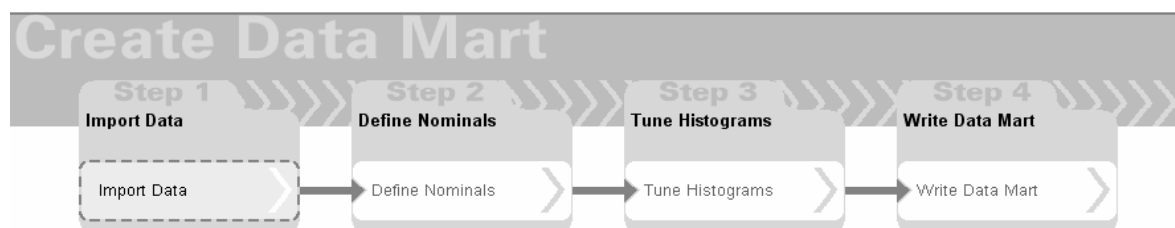
Šioje sistemoje žemėlapis pateikiamas panašiai kaip žemės paviršiaus žemėlapis, kuriame matomi vandens telkiniai, kalnai, įdubos ir t. t. Klases paprastai skiria šviesesni plotai, kurie žymi didesnę atstumą tarp klasių. Jeigu gautas vaizdas netinkamas jį galima redaguoti. Kadangi ši sistema pritaikyta dideliems duomenų kiekiams vizualizuoti, todėl dideliame žemėlapyje puikiai matosi išskirtos vektorių klasės. Sistema turi didelį trūkumą, kad negalima pamatyti kvantavimo ir topografinės paklaidų.



3.7 pav. Sistemoje Databionic ESOM gautas SOM žemėlapis

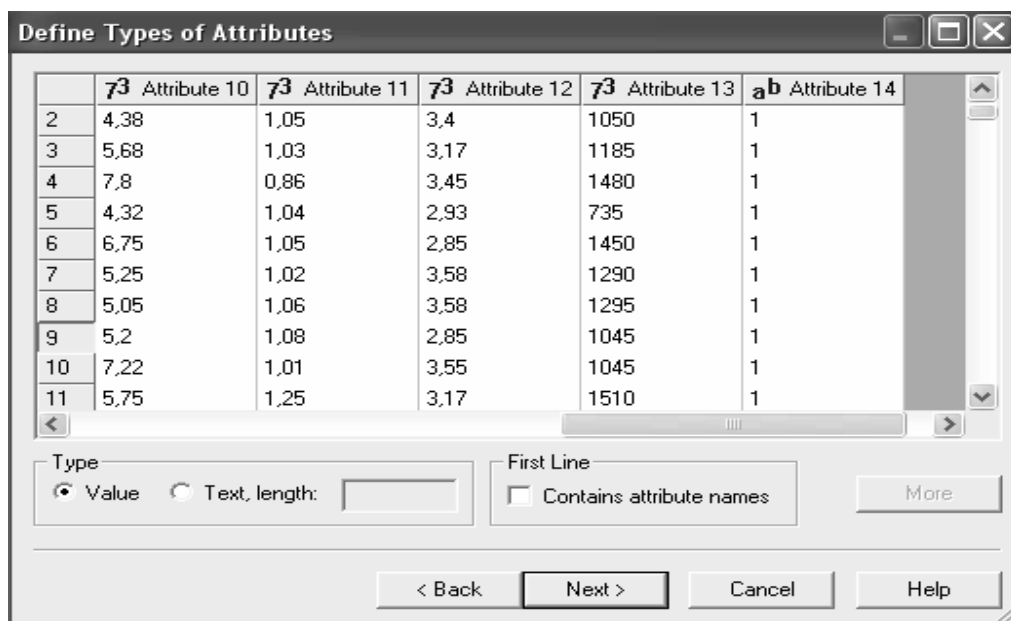
3.4. Sistema Viscovery SOMine

Sistema Viscovery SOMine šiuo metu yra labiausiai naudojama ir nuolat tobulinama saviorganizuojančių neuroninių tinklų programa. Bandomąją 30 dienų versiją galima atsisiųsti internete nemokamai. Pilnos versijos licenziją reikia pirkti. Svetainėje yra pateikiamos sistemos galimybės nagrinėjant SOM tinklus (Viscovery SOMine 5.0, 2010), bei naudojimo demonstracijos (Viscovery SOMine 5.0 demo, 2010). Bandomoji versija tinklo apmokymo ar kitų nustatymų apribojimų neturi. Sistema turi ganėtinai patrauklią grafinę sąsają ir nesudėtingą valdymą. Paprasčiausio žemėlapio kūrimas vyksta dvejais etapais: duomenų (*Create Data Mart*) ir modelio sukūrimais (*Create Model*). Duomenų sukūrimo etapai (3.8 pav.) yra: duomenų įkėlimas (*Import Data*), nominalių kintamųjų apibrėžimas (*Define Nominals*), histogramos (*Tune Histograms*) ir paruoštų duomenų įrašymas (*Write Data Mart*).



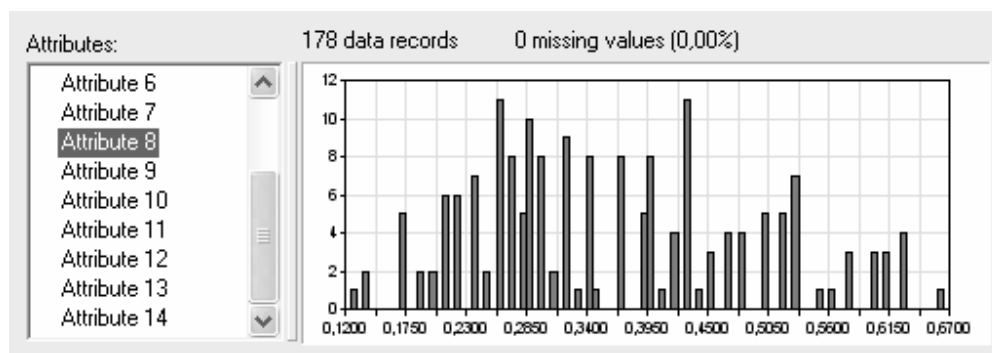
3.8 pav. Duomenų sistemai Viscovery SOMine paruošimo etapai

Įkėlus pasirinktus duomenis, galima nurodyti, iš kurių parametų (atributų) bus sudaryti analizuojami daugiamačiai vektoriai, ir kurie žymi vektorių pavadinimus (3.9 pav.).



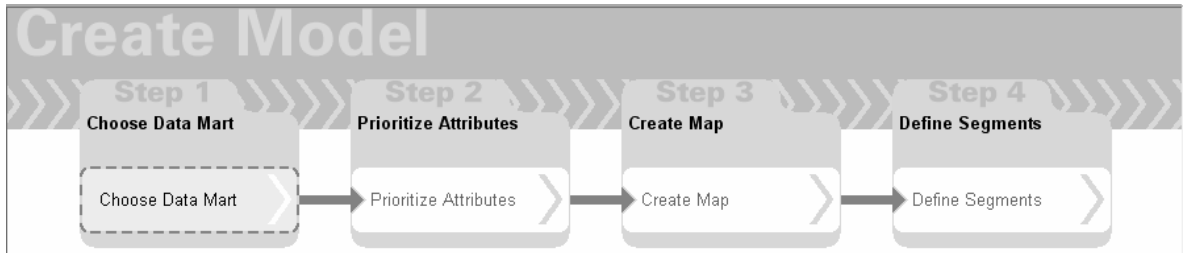
3.9 pav. Paruošti duomenys sistemai Viscovery SOMine

Sistema automatiškai sunumeruoja stulpelius suteikdama jiems vardus, bet šiame etape vartotojas juos gali pakeisti. Užrašas prie vektoriaus parametro (Attribute) a^b reiškia, kad duomenys bus traktuojami kaip tekstiniai. Taip pat įmanoma, panaudojus įvairias skaičiavimo formules, sukurti papildomus išvestinius parametrus, vieną pasirinktą parametą padaryti pirminiu raktu. Histogramose matosi vaizdas, kaip pasiskirsto vieno ar kito parametro reikšmė (3.10 pav.)



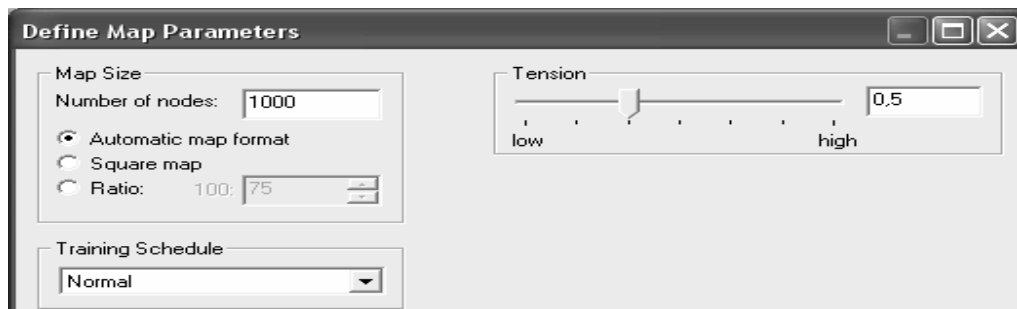
3.10 pav. Parametų reikšmių pasiskirstymo histogramos sistemoje Viscovery SOMine

Nustačius reikiamus duomenų parametrus prasideda SOM modelio kūrimas. Jis vyksta taip pat etapais (3.11 pav.): praeitime etape paruoštų duomenų pasirinkimas (*Choose Data Mart*), parametrų (atributų) prioritetus nustatymas (*Prioritize Attribute*), žemėlapių sukūrimas (*Create Map*) ir segmentų apibrėžimas (*Define Segments*).



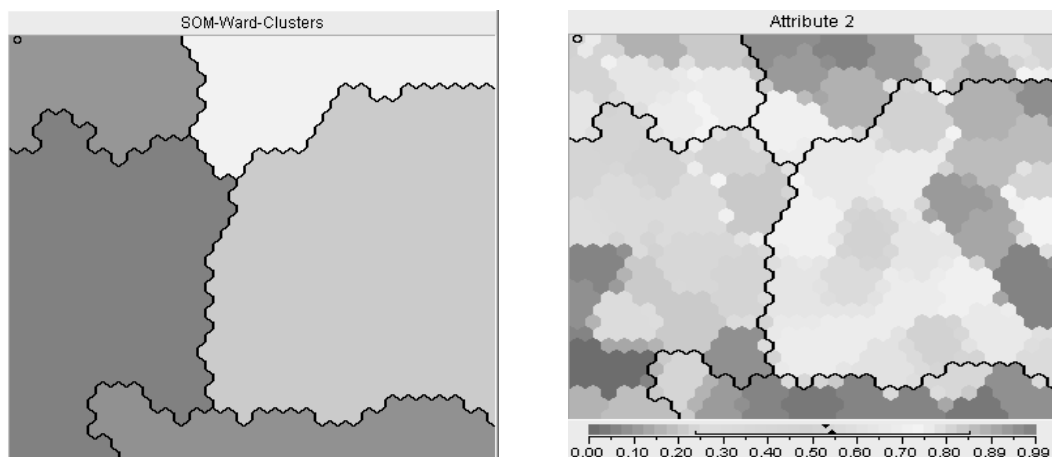
3.11 pav. SOM modelio kūrimo etapai sistemoje Viscovery SOMine

Šiame etape pasirinkus ir įkėlus duomenis kortelėje „*create map*“ galima pasirinkti mokymo parametrus (3.12 pav.), keisti žemėlapių formą (*Map size*), žemėlapių elementų (langelių) skaičių (*Number of nodes*) ir kitus.



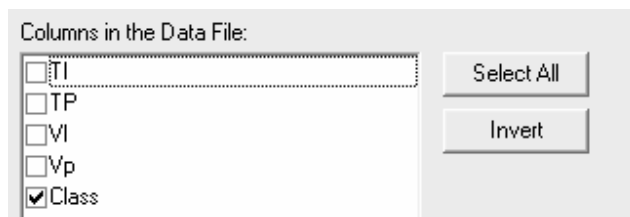
3.12 pav. SOM parametrų nustatymas sistemoje Viscovery SOMine

Užbaigus apmokymą gauname SOM žemėlapi (3.13 pav.). Programoje galima matyti susidariusius žemėlapius pagal kiekvieną parametą (atributą) ir bendrą SOM žemėlapi. Yra galimybė išmesti nereikalingus parametrus.



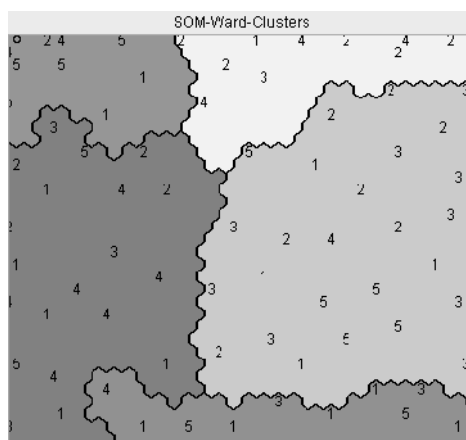
3.13 pav. SOM žemėlapiai sistemoje Viscovery SOMine

Tam, kad žemėlapyje matytume susidariusias klases, reikia gautiems žemėlapiams nurodyti žymes (*labels*). Tai galima padaryti meniu juostoje pasirinkus *File* → *Import* → *Labels*. Nurodžius failą, iš kurio bus imamos klasės, pasirenkame stulpelį, kuriame nurodytos duomenų klasės (3.14 pav.).



3.14 pav. Klasių parinkimas sistemoje Viscovery SOMine

Tuomet žemėlapyje yra matomos klasės (3.15 pav.).



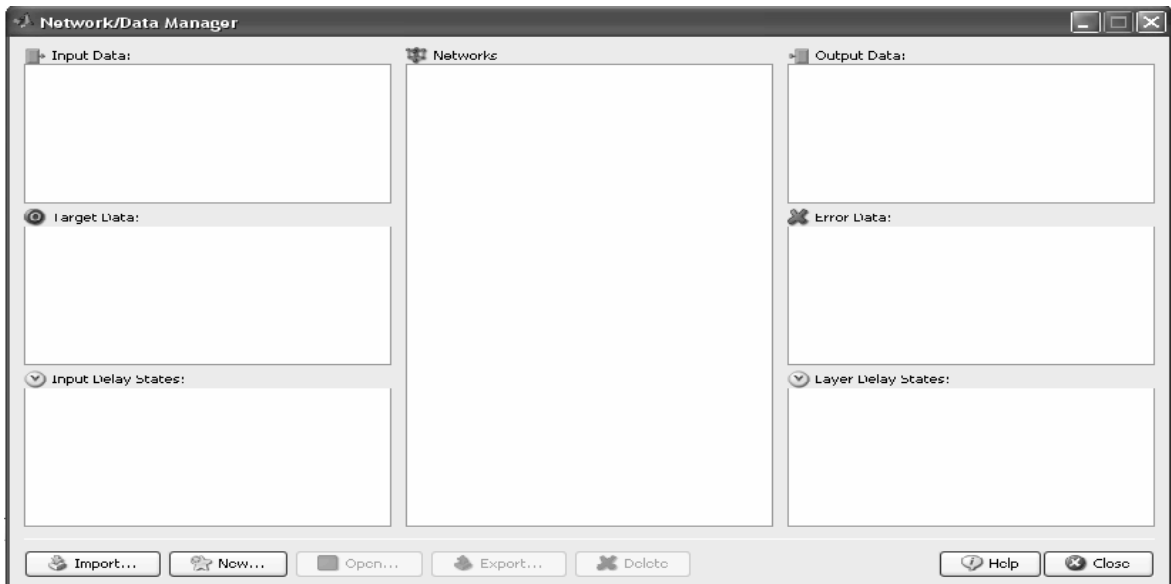
3.15 pav. Klasių numeriais sužymėtas SOM žemėlapis sistemoje Viscovery SOMine

Duomenis apmokymui sistemoje SOMine galima įkelti ne tik iš tekstinio failo, bet ir Microsoft Excel dokumento, SPSS dokumento ir pačios sistemos jau prieš tai sukurtų failų. Norint peržiūrėti kvantavimo paklaidą, ją galime rasti: File → Map History → Description.

3.5. Neuroninio tinklo paketo Matlab'e įrankiai SOM tinklams

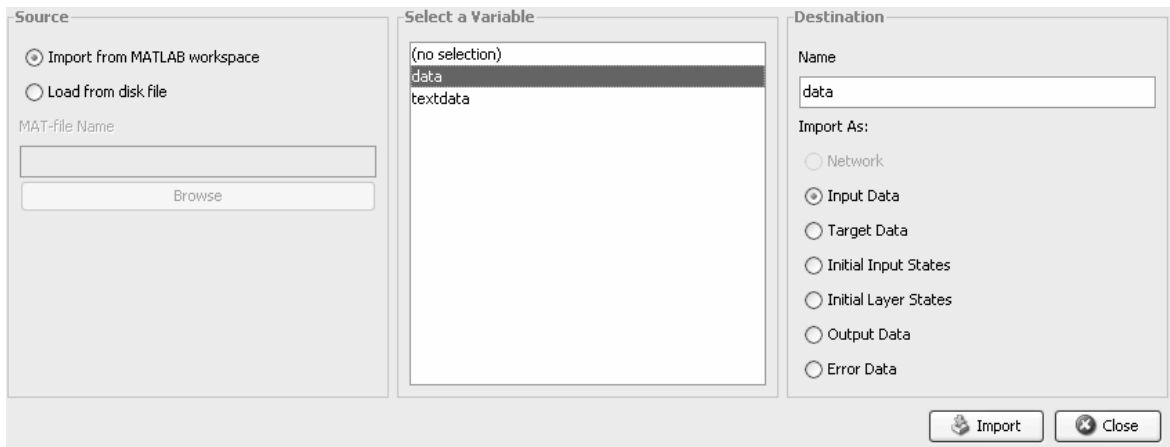
Neskaitant papildomo, kitų autorių sukurto paketo *SOM-Toolbox*, pati sistema Matlab taip turi neuroninių tinklų paketą „*nnet*“ (Neural Networks Toolbox 6). Šiame pakete yra du įrankiai, kurių pagalbą galima gauti SOM žemėlapi: *nntool* (*neural network tool*) ir *nctool* (*neural network clustering tool*). Įrankis „*nctool*“ yra Matlab 7.7 ir vėlesnėse versijose. Abiejuose įrankiuose duomenys turi būti arba tekstiniam failo arba kintamasis, sukurtas Matlab kintamųjų lauke (*workspace*).

Aktyvavus įrankį (3.16 pav.), apmokymas vyksta tokiais etapais: duomenų paruošimas, tinklo kūrimas ir mokymas.



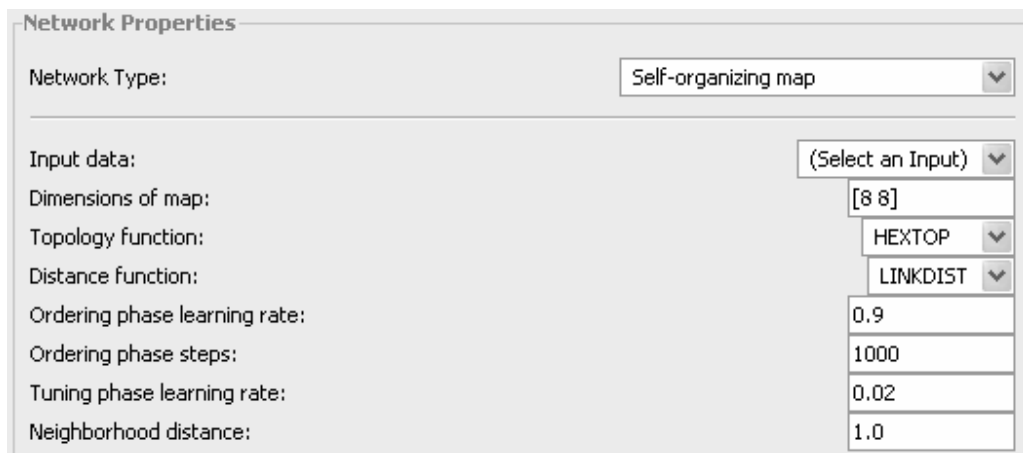
3.16 pav. Matlab įrankis nntool

Galima sukurti naujus duomenis (mygtukas „New...“) arba pasirinkti iš tekstinio failo (mygtukas „Import...“). Prieš tai reikia pasirinkti Input Data laukelį. Duomenis galima įkelti kaip įėjimo duomenis (Input Data), trokštamus išėjimo duomenis (Target Data), paklaidas (Error data) ir kt. (3.17 pav.).



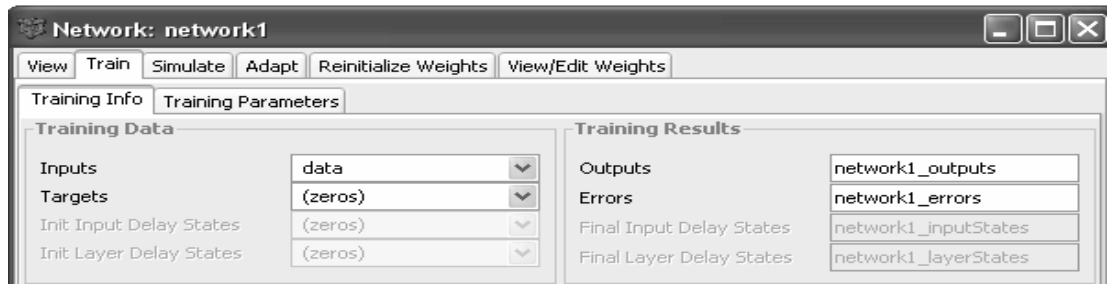
3.17 pav. Duomenų įkėlimas įrankyje nntool

Kai duomenys yra įkelti, reikia sukurti SOM tinklą. Tai darome paspaudus mygtuką „New“, prieš tai reikia pasirinkti „Networks“ laukelį. Atsiradusioje lentelėje (3.18 pav.) visų pirma reikia pasirinkti tinklo tipą (*Network type*) ir apmokymo duomenis (*Input data*). „Nntool“ leidžia kurti ne tik SOM tinklą, bet ir kitų tipų neuroninius tinklus. Kiti parametrai nebūtinai ir pritaikomi pagal kiekvieną atskirą atvejį: žemėlapių dydis (*Dimensions of map*), topologijos funkcija (*Topology function*), atstumo funkcija (*Distance function*), kaimyninė funkcija (*Neighborhood function*) ir kiti.



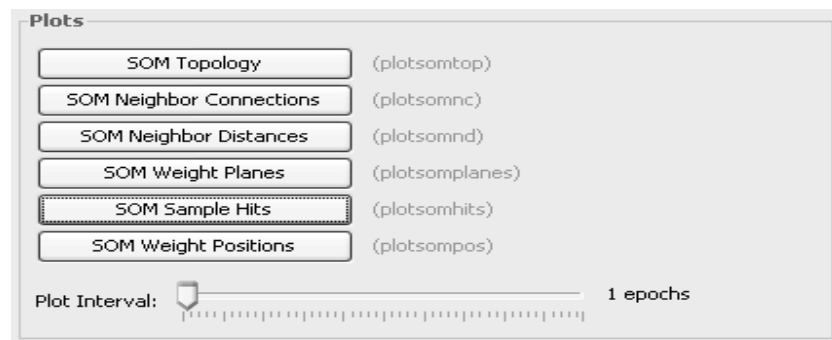
3.18 pav. Tinklo sukūrimo kortelė įrankyje nntool

Sukūrus tinklą, paspaudus du kartus ant jo pavadinimo galima nustatyti mokymo parametrus ir nurodyti duomenis, kurie bus naudojami mokant tinklą (3.19 pav.).



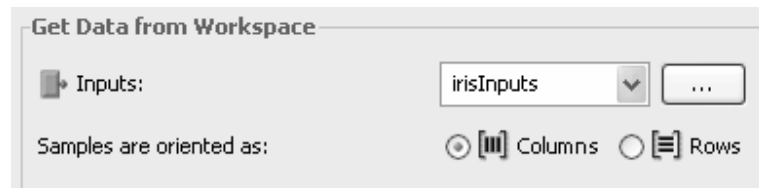
3.19 pav. Tinklo mokymo parametrų ir duomenų parinkimo kortelės įrankyje nntool

Apmokius tinklą įrankyje nntool galima peržiūrėti (3.20 pav.) tinklo topologiją (SOM Topology), kaimynų jungtis (SOM Neighbor Connections), kaimynų atstumus (SOM Neighbor Distance), svorių pasiskirstymą (SOM Weight Planes), neuronų pataikymą į žemėlapių segmentus (SOM Sample Hits) ir svorių pasiskirstymą (SOM Weight Positions).



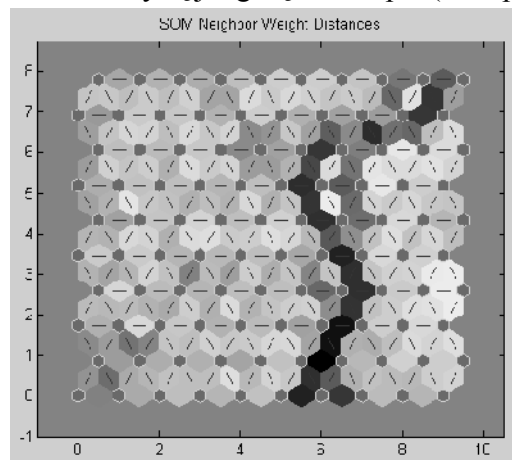
3.20 pav. Vaizdavimo galimybių pasirinkimas įrankyje nntool

Įrankis „nctool“ turi labai daug panašumų kaip ir „nntool“, bet skirtingai šis įrankis leidžia sukurti tik SOM žemėlapius, t. y. jis skirtas SOM žemėlapiams kurti. Abu įrankiai turi didelį trūkumą, jog negalima žemėlapiuose atvaizduoti susidariusias klases ir pamatyti gautas paklaidas. Taigi, šiuos įrankius būtina tobulinti. Įrankyje „nctool“ neuroninis tinklas yra apmokomas tokiais etapais: duomenų įkėlimas, apmokymas, testavimas. Pirmame etape duomenys gali būti įkelti iš tekstinio failo (3.21 pav.) arba iš Matlab aplinkoje paruoštų duomenų.



3.21 pav. Duomenų įkėlimas įrankyje nctool

Įkėlus duomenis iš pradžių galima pasirinkti tik tai SOM žemėlapiu dydį. Paspaudus mygtuką „train“ tinklas yra apmokomas ir tuomet galima peržiūrėti tuos pačius žemėlapius kaip ir įrankyje nntool (5 pav.). Taip pat pateikiamas kaimynų jungčių žemėlapis (3.22 pav.).



3.22 pav. Kaimyninių neuronų atstumų žemėlapis įrankiuose „nctool“ ir „nntool“

4 SOM sistemų eksperimentinis palyginimas

4.1. Tyrimo duomenys

Analizuoti trys duomenų rinkiniai: irisų, stiklo ir vyno duomenys (Asuncion, Newman, 2007). Irisų duomenys – tai trijų veislių gėlės: *Iris Setosa*, *Iris Versicolor* ir *Iris Virginica*. Yra išmatuoti keturi gėlių parametrai:

x_1 – taurėlapių ilgiai (*sepal length*),

x_2 – taurėlapių pločiai (*sepal width*),

x_3 – vainiklapių ilgiai (*petal length*),

x_4 – vainiklapių pločiai (*petal width*).

Sudaryti 4-mačiai vektoriai X_1, X_2, \dots, X_{150} ; čia $X_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4})$, $i = 1, \dots, 150$. Vektoriai X_1, X_2, \dots, X_{50} atitinka I klasės irisus (*Iris Setosa*), vektoriai $X_{51}, X_{52}, \dots, X_{100}$ – II klasės (*Iris Versicolor*) ir vektoriai $X_{101}, X_{102}, \dots, X_{150}$ – III klasės (*Iris Virginica*).

Stiklo duomenys buvo surinkti mokslininkų, kurie norėjo padėti atpažinti kriminalistų rastas stiklo šukes (1 priedas). Šių duomenų rinkinį sudaro 9-mačiai vektoriai, tačiau dėl sistemos *NeNet* demonstracinės versijos ribotumo buvo analizuoti 6-mačiai vektoriai, atsisakyta trijų paskutinių parametrų. Matuoti šie parametrai:

x_1 – lūžimo indeksas (*Refractive Index*),

x_2 – natris (*Sodium*),

x_3 – magnis (*Magnesium*),

x_4 – aliuminis (*Aluminum*),

x_5 – silikonas (*Silicon*),

x_6 – kalis (*Potassium*).

Taigi, turime 6-mačius vektorius X_1, X_2, \dots, X_{119} ; čia $X_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}, x_{i6})$, $i = 1, \dots, 119$. Pastatų langų (*Building windows*) stiklai sudaro I klasę, juos atitinka vektoriai X_1, X_2, \dots, X_{70} , taros stiklai (*Containers*) sudaro II klasę, (vektoriai $X_{71}, X_{72}, \dots, X_{83}$), stalo įrankiai (*Tableware*) sudaro III klasę, (vektoriai $X_{84}, X_{85}, \dots, X_{92}$) ir žibintai (*Headlamps*) sudaro IV klasę (vektoriai $X_{93}, X_{94}, \dots, X_{119}$).

Vyno duomenys yra surinkti tame pačiame Italijos regione, tik iš 3 skirtingų žemdirbių. Buvo tiriama vyno cheminė sudėtis. Internetu pateikiami 13-mačiai vektoriai, bet kaip ir stiklo duomenų atveju dėl sistemos *NeNet* ribotumo bus analizuojami 6-mačiai vektoriai X_1, X_2, \dots, X_{178} ; čia

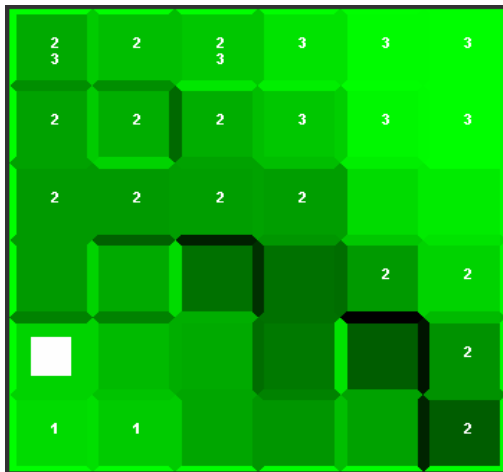
$X_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}, x_{i6})$, $i = 1, \dots, 178$. 6 parametrai buvo pasirinkti pagal atliktų bandymų metu geriausiai susidariusį SOM žemėlapi, o likusių 7 atsisakyta. Vektorius sudaro šie parametrai:

- x_1 – alkoholio laipsnis (alcohol)
- x_2 – obuolių rūgštis (Malic acid),
- x_3 – magnis (magnesium),
- x_4 – grynoji karbolio rūgštis (total phenols),
- x_5 – spalvos intensyvumas (color intensity)
- x_6 – atspalvis (hue).

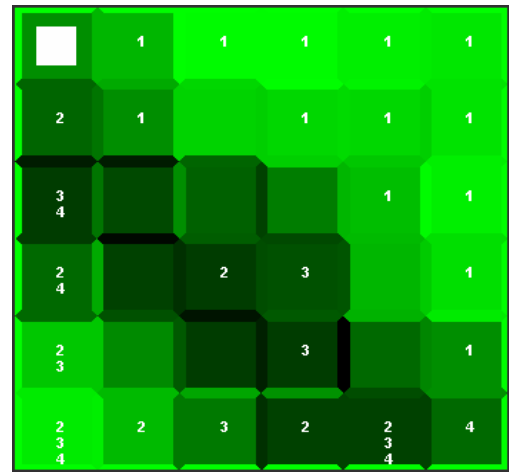
Pirmojo žemdirbio vyno duomenys sudaro pirmąją klasę, vektoriai X_1, X_2, \dots, X_{59} , II – klasę sudaro antrojo žemdirbio vyno duomenys (vektoriai $X_{60}, X_{61}, \dots, X_{130}$) ir III – klasę trečiojo žemdirbio vyno duomenys (vektoriai $X_{131}, X_{132}, \dots, X_{178}$).

4.2. Gautų žemėlapių analizė

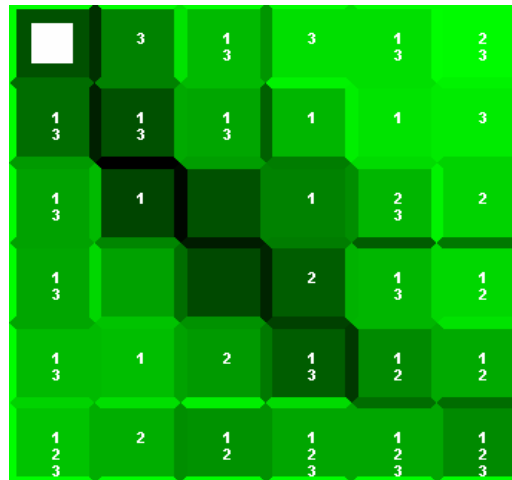
Sistemoje *NeNet* vektoriai yra išdėstomi $r \times s$ žemėlapyje (mūsų atveju $r = s = 6$). Atspalviai atitinka U-matricos reikšmes (Ultsch, Siemon, 1989). Tamsesnė spalva tarp žemėlapių langelių reiškia, kad tuose langeliuose esantys vektoriai yra tolimesni nei tie, tarp kurių briaunos spalva yra šviesesnė (Kurasova, Stefanovič, 2009). Irisų atveju aiškiai matyti, jog viename kampe dominuoja I klasės vektoriai, o kitoje pusėje – II ir III klasės vektoriai (4.1 pav.). Žemėlapyje matome po vieną arba du skaičius viename langelyje, nors iš tiesų į tą patį langelį gali patekti daug tos pačios klasės vektorių, bet sistema vaizduoja tik po vieną tos klasės narį. Stiklo duomenys, ne taip kaip irisų, išsidėsto labiau padrikai, bet vėl matyti, jog I klasė (pastatų langai) atsiskiria nuo kitų trijų klasių (4.2 pav.). Vyno duomenų atveju šioje sistemoje rasti skirtumus tarp klasių ganėtinai sunku (4.3 pav.), todėl reiškia jog pagal parametrus tų trijų klasių duomenys panašūs.



4.1 pav. Irisų duomenys 6×6 SOM žemėlapyje, gautame sistema *NeNet*

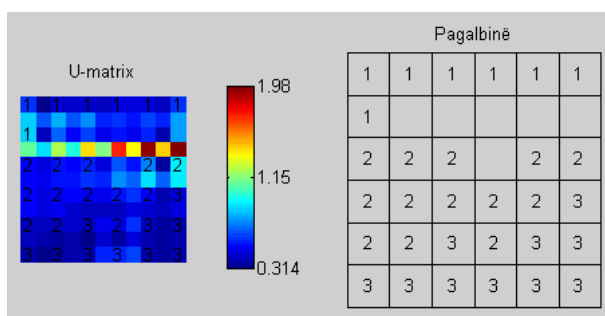


4.2 pav. Stiklų duomenys 6×6 SOM žemėlapyje, gautame sistema *NeNet*

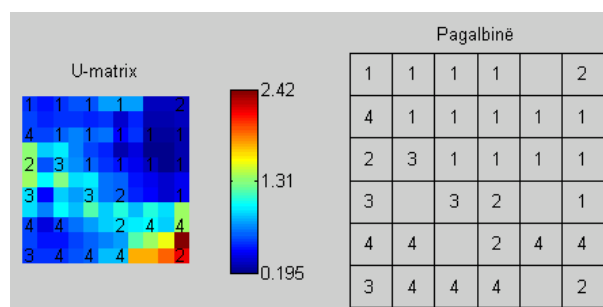


4.3 pav. Vyno duomenys 6×6 SOM žemėlapyje, gautame sistema *NeNet*

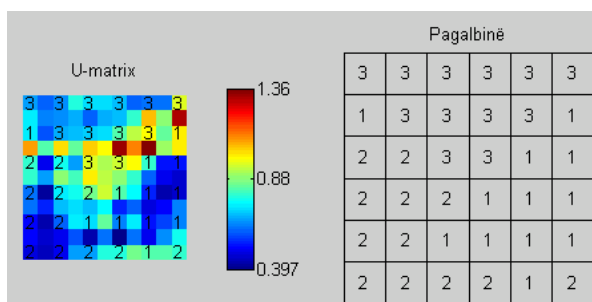
SOM-Toolbox sistemoje priešingai nei yra *NeNet* sistemoje žemėlapių elementus skiria ne siaura briauna, bet tarpiniai langeliai, todėl nors irisų, stiklo ir vyno atveju pasirinktas 6x6 žemėlapis, tačiau matomi 11x11 žemėlapiai (4.4–4.6 pav.). Pagalbinėse lentelėse matomi tik tinklo elementai be tarpinių langelių. 4.4 paveiksle vėl matomos išsiskiriančios dvi irisų grupės – tai I klasė ir II kartu su III klase. 4.5 paveiksle taip pat matyti, jog stiklo II, III, IV klasių vektoriai yra tarpusavyje susimaišę, o I klasės vektoriai visi greta. Šioje sistemoje sunkiau pastebėti skirtingas stiklo duomenų klases negu *NeNet* sistemoje. Vyno duomenys skirtingai negu sistemoje *NeNet* pasiskirsto lyg į tris skirtingas grupes, žemėlapių viršuje išsidėstę III klasės vektoriai, kairėje II klasės ir dešinėje I klasės nariai, todėl galima manyti, jog *SOM-Toolbox* atvaizduoja šiuos duomenis geriau.



4.4 pav. Irisų duomenys 6×6 SOM žemėlapyje, gautame sistema *SOM-Toolbox*

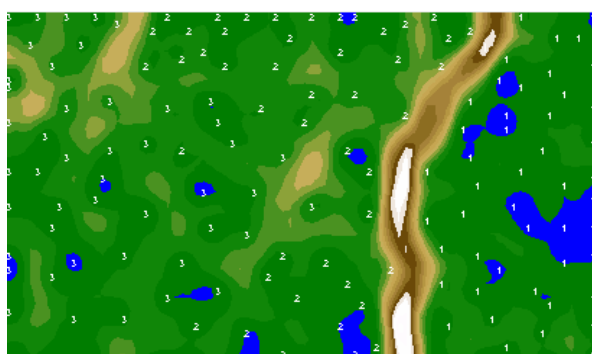


4.5 pav. Stiklo duomenys 6×6 SOM žemėlapyje, gautame sistema *SOM-Toolbox*

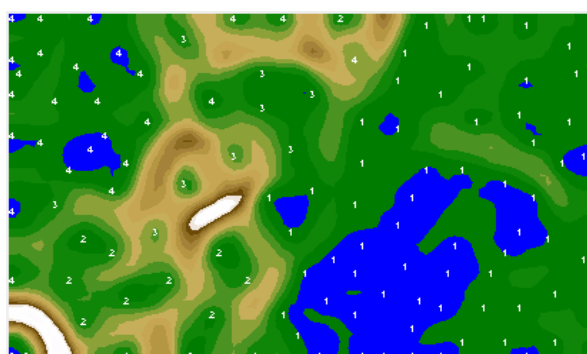


4.6 pav. Vyno duomenys 6×6 SOM žemėlapyje, gautame sistema *SOM-Toolbox*

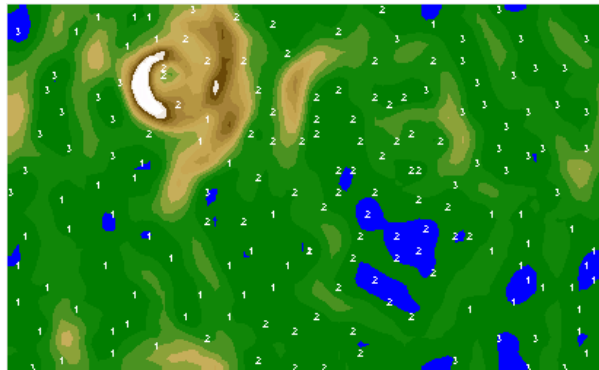
Databionic ESOM sistema skirta dideliems SOM tinklams kurti, todėl žemėlapių dydį reikia nustatyti pakankamai didelį (50×82). Kaip ir anksčiau pateiktuose vaizduose matome dvi atskiras irisų sritis: I klasė ir II su III klase kartu (4.7 pav.). Šioje sistemoje „grioviai“ parodo skiriamąją klasių ribą. 4.8 paveiksle I klasės stiklo duomenys atsiskiria nuo kitų klasių. Vyno duomenys šioje sistemoje (4.9 pav.) gerokai susimaišę, tačiau vis tiek galima pastebėti, jog vienoje vietoje susigrupavę vienos klasės atstovai, kai kuriose žemėlapių vietose jie persipina.



4.7 pav. Irisų duomenys SOM žemėlapyje, gautame sistema *Databionic ESOM*

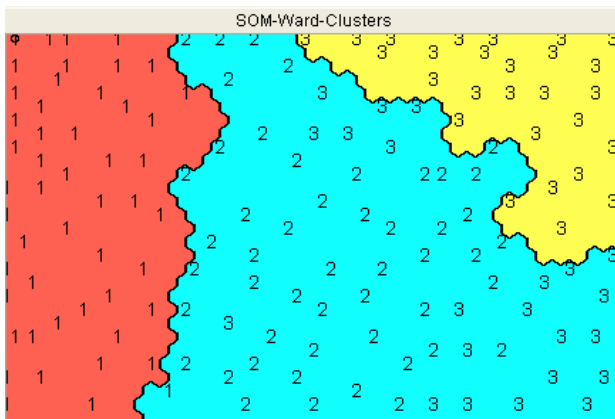


4.8 pav. Stiklo duomenys SOM žemėlapyje, gautame sistema *Databionic ESOM*

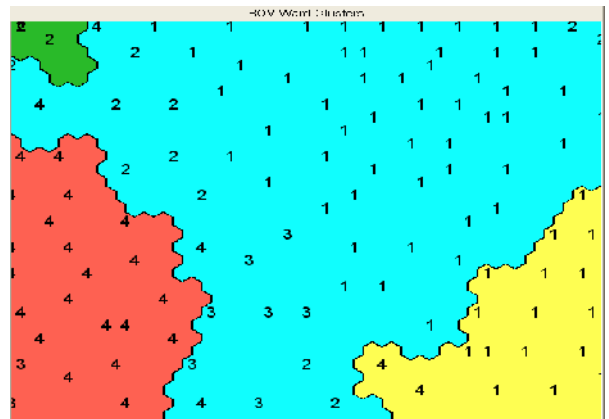


4.9 pav. Vyno duomenys SOM žemėlapyje, gautame sistema *Databionic ESOM*

Viscovery SOMine žemėlapių dydis nustatomas parinkus parametą „nodes“ (3.12 pav.). Sistema SOM žemėlapiuose duomenis atvaizduoja pagal kiekvieną vektoriaus parametą atskirai ir pagal visus duomenų parametrus. Šiame darbe pateikiamas tik bendras, t. y. pagal visus parametrus, SOM žemėlapis. Matome (4.10 pav.), jog sistema irisų duomenis suskirsto į tris skirtingus klasterius, kairėje pusėje raudoni I klasės duomenys, ties viduriu žydros spalvos II ir III klasės vektoriai ir vien tik III klasės vektoriai geltoni. Stiklo duomenys išsibarstę daugiau (4.11 pav.), tačiau matosi, jog kairėje pusėje dominuoja pirmos klasės vektoriai, o kitoje II, III ir IV klasės vektoriai.

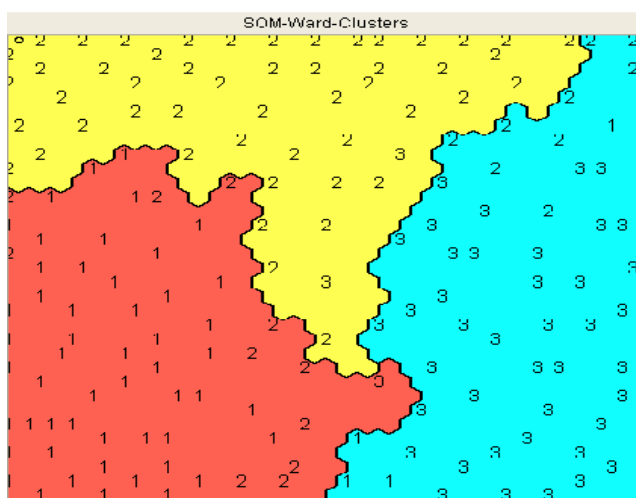


4.10 pav. Irisų duomenys SOM žemėlapyje, gautame sistema *Viscovery SOMine*



4.11 pav. Stiklo duomenys SOM žemėlapyje, gautame sistema *Viscovery SOMine*

Vyno duomenys taip pat pasiskirsto ganėtinai gerai (4.12 pav.). Žydrroje spalvoje dominuoja III klasės vektoriai, geltonoje II klasės vektoriai ir raudonoje I klasės vektoriai.



4.12 pav. Vyno duomenys SOM žemėlapyje, gautame sistema *Viscovery SOMine*

4. 3. Topografinės ir kvantavimo paklaidos sistemose.

Iš anksto žinant duomenų klases, apie duomenų klasterizavimo SOM žemėlapyje kokybę galima spręsti iš gautų vaizdų. Tačiau, kai kasės nėra žinomos, būtina vertinti skaitines paklaidas. Taigi, išmokius SOM žemėlapi, pagal formules (2), (3) apskaičiuojamos kvantavimo ir topografinės paklaidos.

Dėl skirtingų nagrinėtų sistemų epochų traktavimo, palyginti po apmokymo gautas kvantavimo paklaidas yra ganėtai sunku. Sistemoje „NeNet“ epochos parenkamos įprastai, įrašius norimą skaičių kortelėje „Train“. Sistemoje „SOM-Toolbox“ norėdami keisti apmokymo (training) trukmę, galima pasirinkti trumpą (short), numatytą (default) ir ilgą (long) skaičiavimą arba pasirinkti 1×2 vektorių, kuriame pirmas parametras nurodo grubų epochų skaičių (rough epochos), o antras suderintą epochų skaičių. (finetuning epochos). „Databionic ESOM“ sistemoje epochos pateikiamos įrašius laukelyje „number of training epochos“, tačiau kaip minėjome šioje sistemoje paklaidų pamatyti negalima. Žinoma, jas galima suskaičiuoti žinant neuronų nugalėtojų reikšmes, tačiau sistema pateikia neuronų nugalėtojų koordinatas SOM žemėlapyje atskirai nuo pačių neuronų nugalėtojų reikšmių, todėl reikia nemažai papildomo darbo norint nustatyti, į kurį SOM tinklo langelį, pakliuvo tam tikras analizuojamas vektorius. Šiame tyrime tai nėra atlikta. „Viscovery SOMine“ galimas apmokymo trukmės parinkimas: normalus (normal), greitas (fast) ir nuodugnus (accurate).

Žinoma, kad SOM tinklo kokybė priklauso nuo neuronų pradinių reikšmių parinkimo, mokymo taisyklės, žemėlapi dydžio ir kitų parametru, todėl norint spręsti apie tinklo kokybę neužtenka vertinti paklaidas, gautas vienu eksperimentu, o būtina jų atlikti daugiau ir vertinti vidutines paklaidas, tačiau

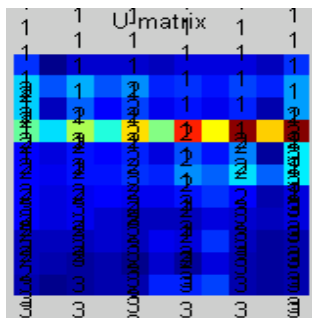
dėl skirtingų epochų parinkimo sistemose to padaryti negalima. Kadangi kiekviena sistema gaunami šiek tiek kitokie SOM vaizdai, todėl tikslinga naudoti kelias sistemas, vertinti jomis gautus vaizdus ir interpretuoti rezultatus. Keli gauti vaizdai leidžia daryti išsamesnes išvadas apie analizuojamus duomenis. Susisteminti analizuotų sistemų privalumai ir trūkumai pateikti 1 lentelėje.

1 lentelė. Analizuojamų sistemų privalumai ir trūkumai

Sistema	Privalumai	Trūkumai
NeNet	Lengvai suprantama pradedančiajam vartotojui. Turi patrauklią grafinę sąsają.	Yra tik bandomoji versija, todėl turi apribojimų. Jau senokai sistema nėra atnaujinama.
SOM-Toolbox	Turi daugiau kaip 200 funkcijų. Įvairūs vizualizavimo būdai.	Veikia tik Matlab sistemoje. Naujesnėse Matlab versijose kai kurios funkcijos neveikia. Reikalinga turėti žinių apie Matlab sistemą.
Databionic ESOM	Lengvai ir aiškiai atvaizduoja didelius duomenų rinkinius. Galima keisti daug mokymo parametrų.	Sudėtingai atskirai paruošiami mokymo ir vektorių pavadinimų failai. Nėra galimybės matyti kvantavimo ir topografinių paklaidų.
Viscovery SOMine	Įvairūs ir detalūs vizualizavimo parinkimai. Daug įvairių nustatymų, leidžiančių paruošti gautą žemėlapi pagal savo poreikius. Tinklas yra gerai apmokomas, nes gaunamos mažos paklaidos. Patraukli vartotojo grafinė sąsaja. Duomenys gali būti įkeliami įvairiais būdais.	Ganėtinai sunkus valdymas, todėl reikia daug žinių norint gauti norimą žemėlapi.

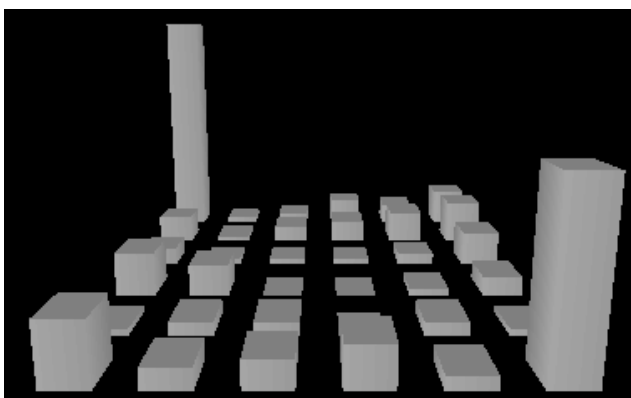
5 Nauja saviorganizuojančių tinklų sistema

Vizualizuojant duomenis susiduriame su problema, jog prieš tai nagrinėtose sistemose nėra galimybės matyti visus neuronus nugalėtojus patekusius į SOM žemėlapių langelį arba jos nėra aiškios. Sistemoje „SOM-Toolbox“ galima nurodyti (`sm=som_autolabel(sm,sd,'add')`), kad žemėlapyje būtų rodomi visų į neuroną pateikusių vektorių ar jų klasių pavadinimai, tačiau tuomet nieko neįmanoma suprasti (5.1 pav.).



5.1 pav. SOM-Toolbox sistemoje gautas žemėlapis, kuriame nurodyti visi analizuojamų vektorių pavadinimai

Sistemose „Databionic ESOM“ ir „Viscovery SOMine“ tokios galimybės visai nėra. „NeNet“ leidžia atlikus testavimą pasižiūrėti gauto žemėlapių histogramą (5.2 pav.), bet vis tiek lieka neaišku, kiek kokios klasės narių pateko į vieną SOM žemėlapių langelį. Tuo tikslu buvo sukurta nauja vizualizavimo būdą turinti SOM sistema, kurioje matosi, kiek kokios klasės narių pateko į susidariusio žemėlapių langelius.



5.2 pav. NeNet sistemos žemėlapių histograma

Nauja saviorganizuojančių tinklų sistema sukurta Matlab aplinkoje. Ją sudaro keturi m-failai: „pyragas“, „SOM“, „Som-table“ ir „Somas“. Failas „pyragas“ yra modifikuota Matlab funkcija „pie“, kuri skirta atvaizduoti duomenims skrituline diagrama. Iš funkcijos „pie“ buvo išmesti įvairūs papildomi tikrinimai, pritaikytas skritulinės diagramos dydis prie SOM žemėlapių langelių dydžio,

parinktos atsitiktinės spalvos priklausomai nuo klasių skaičiaus. M-faile „SOM“ suprogramuotas tinklo apmokymas, neurono nugalėtojo paieška ir kiti reikalingi skaičiavimai. „Som-table“ skirtas žemėlapiu braižymui ir klasių sužymėjimui. Ši programa pritaikyta duomenimis, kurių klasių skaičius ne daugiau 5, vizualizuoti. Matlab „GUI Builder“ pagalba buvo sukurtas failas „Somas“, kuriame naujos sistemos grafinės sąsajos programinis kodas.

5.1. Sistemos apmokymo algoritmas

Šioje sistemoje yra realizuotas kitoks kaimynystės funkcijos skaičiavimas, negu prieš tai nagrinėtose sistemose. Pagrindinės tinklo mokymo dalys yra tos pačios:

```

SOM mokymo algoritmas
FOR  $e = 1$  TO  $\hat{e}$ 
  FOR  $l=1$  TO  $m$ 
    FOR  $i=1$  TO  $r$ 
      FOR  $j=1$  TO  $s$ 
         $\|M_{ij} - X_l\| = \sqrt{\sum_{p=1}^n (m_{ij}^p - x_{lp})^2}$  // skaičiuojamas Euklido atstumas
      END
    END
     $c = \arg \min_{i,j} \{\|X_l - M_{ij}\|\}$ ,  $\hat{M}_c$  – vektoriaus  $X_l$  neuronas nugalėtojas
    FOR  $i=1$  TO  $r$ 
      FOR  $j=1$  TO  $s$ 
         $M_{ij}(t+1) = M_{ij}(t) + h_{ij}^c(t)(X_l - M_{ij}(t))$ , // SOM mokymo taisyklė
      END
    END
  END
END

```

Įprastai yra naudojamos Gaussian (4) arba burbuliuko (5) kaimynystės funkcijos:

$$h_{ij}^c = \alpha(t) \cdot \exp\left(\frac{-\|R_c - R_{ij}\|^2}{2\sigma^2(t)}\right), \quad (4)$$

$$h_{ij}^c = \begin{cases} \alpha(t), & \text{kai } (i, j) \in N_c \\ 0, & \text{kai } (i, j) \notin N_c \end{cases}, \quad (5)$$

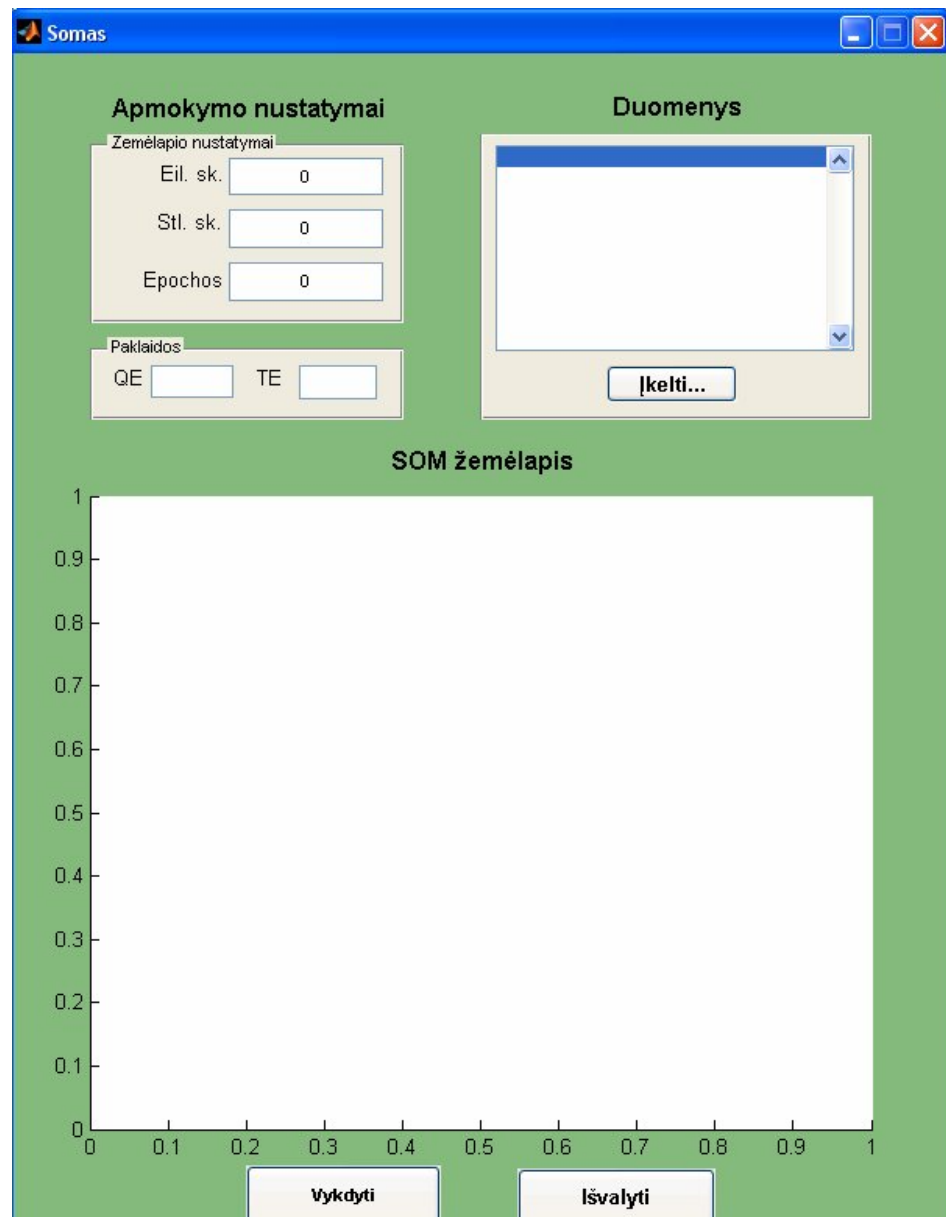
čia $\alpha(t)$ yra mokymo parametras (angl. *learning rate*), o parametras $\sigma(t)$ atitinka kaimynystės N_c eilę. Funkcijos $\alpha(t)$ ir $\sigma(t)$ yra monotoniškai mažėjančios funkcijos. Čia R_c ir R_{ij} yra dvimačiai vektoriai, sudaryti iš M_c ir M_{ij} indeksų (eilučių ir stulpelių numerių), nusakančių vektoriaus X_k neurono nugalėtojo M_c ir perskaičiuojamo neurono M_{ij} vietą SOM tinkle. Mūsų atveju sistemoje buvo panaudota kita kaimynystės funkcija (6) (Dzemyda, Kurasova, Žilinskas, 2008):

$$h_{ij}^c = \frac{\alpha}{a\eta_{ij}^c + 1}, \text{ čia } \alpha = \max\left(\frac{\hat{e} + 1 - e}{\hat{e}}; 0,01\right) \quad (6)$$

čia \hat{e} – pasirinktas epochų skaičius, e – einamosios epochos numeris, η_{ij}^c – kaimynystės tarp M_c ir M_{ij} eilė.

5.2. Sistemos grafinė sąsaja

Tam, kad galėtume naudoti naują sistemą, reikia turėti ne žemesnę nei 7.1 Matlab programos versiją. Paleidus failą „Somas“, atsiras šios sistemos langas (5.3 pav.).

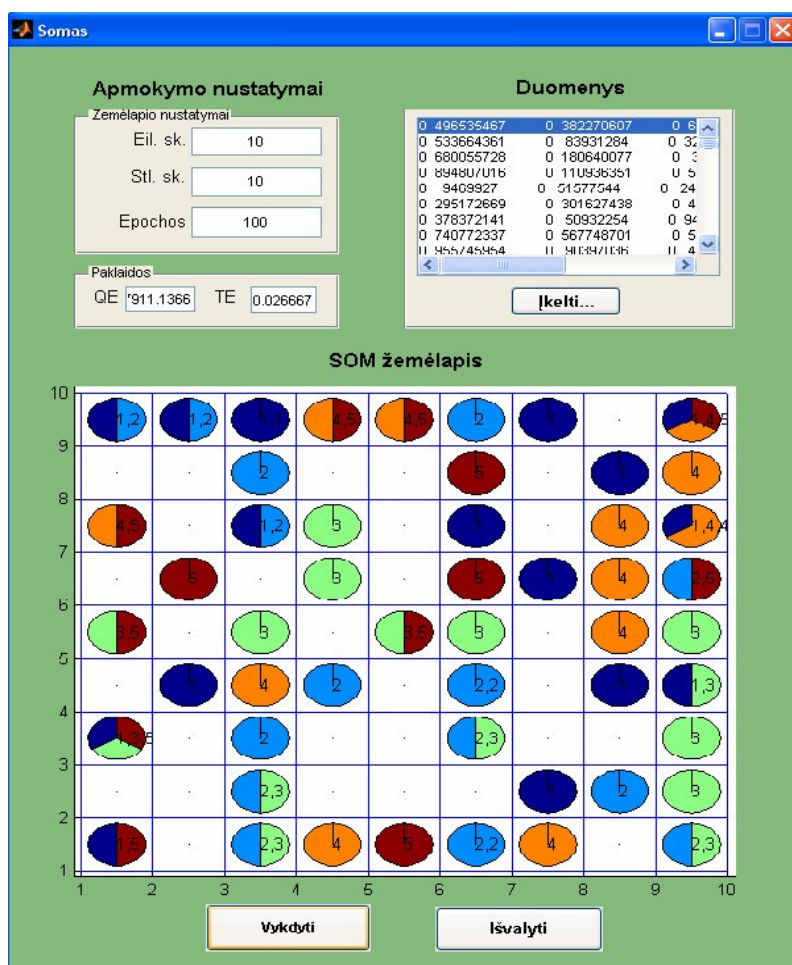


5.3 pav. Naujai sukurta SOM sistema

5.3. SOM vizualizavimas naujoje sistemoje

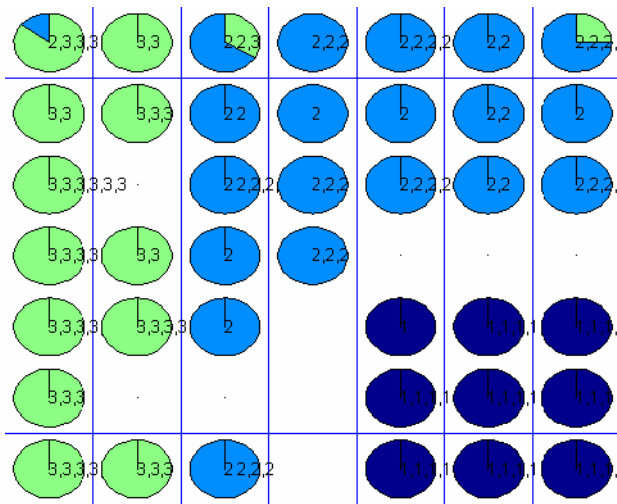
Prieš apmokant tinklą, būtina nurodyti žemėlapių dydį: eilučių ir stulpelių skaičių (Eil. sk. ir Stl. sk.). Taip pat nurodyti epochų skaičių (Epochos), t. y. kiek kartų vektoriai bus paduodami į tinklą. Dešinėje pusėje paspaudus mygtuką „Įkelti...“ yra įkeliami apmokymo duomenys. Duomenys privalo būti tekstiniaame faile. Vektoriai surašyti eilutėmis, o paskutinis stulpelis skirtas vektorių pavadinimams surašyti (klasėms). Įrašius reikiamus duomenis, nuspaudus mygtuką „Vykdyti“ yra apmokomas tinklas. Apmokymo trukmė skirtinga, priklausomai nuo duomenų dydžio.

Apmokius tinklą su pasirinktais duomenimis gaunamas SOM žemėlapis (5.4 pav.). Jo langeliuose skritulinės diagramos, t. y. kelių spalvų (priklausomai nuo klasių kiekio) apskritimai. Kai kurie iš jų suskirstyti į dalis, kiti ištisi – vienspalviai. Ant apskritimo rašomi skaičiai, kurie žymi klasių numerius. Šis vizualizavimo būdas rodo, jog į pasirinktą žemėlapių langelį pakliūna tos klasės vektoriai. Jeigu apskritimas padalintas į kelias dalis, tai reiškia, jog į šį langelį patenka kelių klasių atstovai. Taip pat matome, kurią apskritimo dalį užima vienos klasės atstovai.

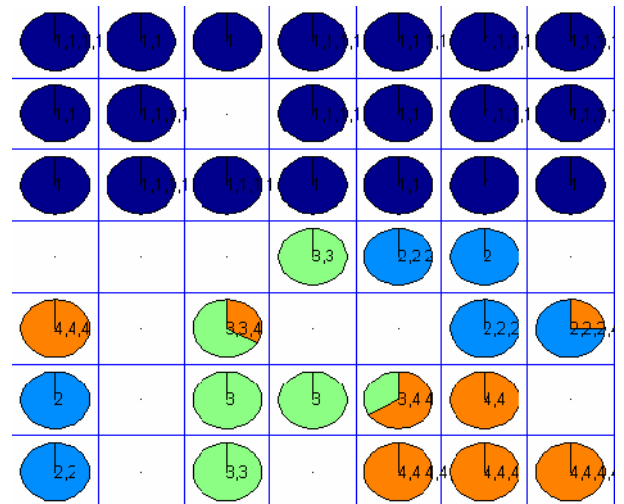


5. 4 pav. Vizualizuoti duomenys naujoje SOM sistemoje

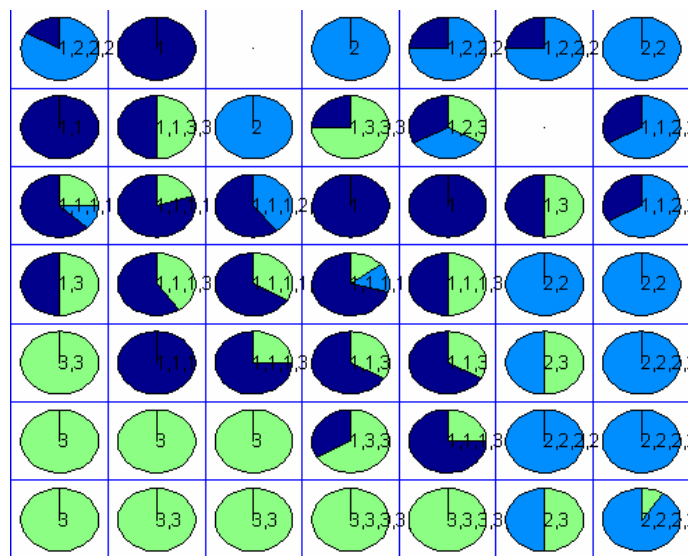
Kaip ir prieš tai nagrinėtų sistemų atveju pasižiūrėsime susidariusius SOM žemėlapius analizuojant irisų, stiklo ir vyno duomenimis. Naują tinklo mokymo algoritmą turinti SOM sistema atvaizduoja irisų duomenis (5.5 pav.), kaip ir kitos sistemos, išskirdama I klasės duomenis, o II ir III klasės duomenys tarpusavyje persipina. Atvaizdavirus stiklo duomenis (5.6 pav.) lygiai taip išsiskiria I klasė, o II, III ir IV klasės susimaišo. Analizuojant vyno duomenis (5.7 pav.), duomenų klasteriai nelabai išsiskiria, nors dalis III klasės duomenų sudaro tam tikrą klasterį, nors kita dalis šios klasės duomenų grupuojasi ir su I, ir II klasės duomenimis.



5.5 pav. Irisų duomenys 8×8 SOM žemėlapyje, gautame nauja sistema



5.6 pav. Stiklo duomenys 8×8 SOM žemėlapyje, gautame nauja sistema



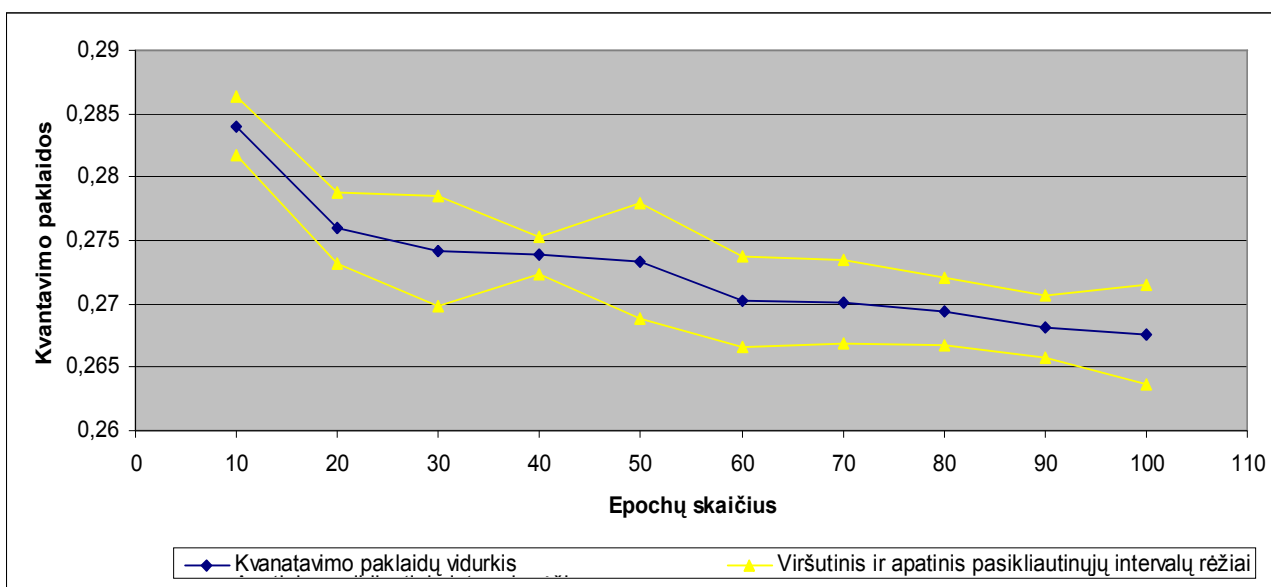
5.7 pav. Vyno duomenys 8×8 SOM žemėlapyje, gautame nauja sistema

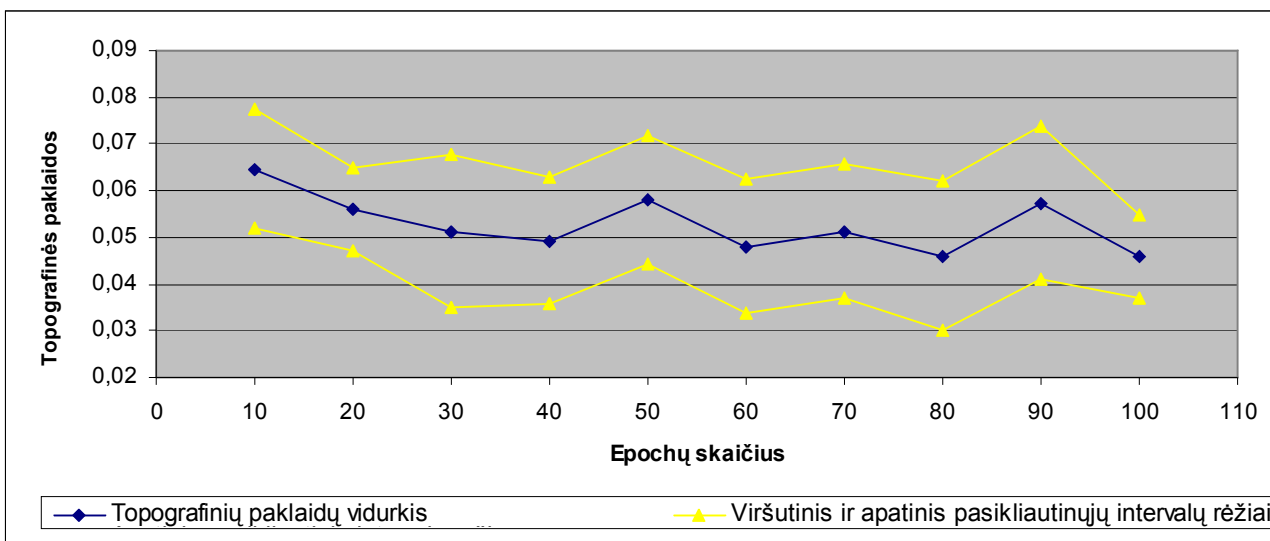
5.4. Eksperimentinio tyrimo rezultatai

5.4.1. Kvantavimo ir topografinės paklaidų priklausomybė nuo epochų skaičiaus

Atlikus tinklo apmokymą, sistema pateikia kvantavimo ir topografinę paklaidas. Norėdami pasižiūrėti, kaip šioje sistemoje priklausomai nuo epochų skaičiaus kinta kvantavimo paklaida buvo atliktas eksperimentinis tyrimas. Sistemoje buvo įkeliami irisų, stiklo ir vyno duomenys, pasirenkamas 10×10 žemėlapis. Bandymo metu epochų skaičius buvo kintantis nuo 10 iki 100, didinant epochas kas 10. Buvo pasirinkta apsiriboti 10 bandymų, kadangi atlikus daugiau rezultatai iš esmės nesikeitė, epochų skaičių imant didesnę rezultatai taip pat žymiai nekito. Tam, kad kaskart būtų apmokas tinklas su tais pačiais duomenimis tik kistų epochų skaičius, buvo pasirenkama ta pati generavimo sėkla (seed). Tai atlieka Matlab komanda `rand('seed', x)`, kai $x = 0, \dots, \infty$.

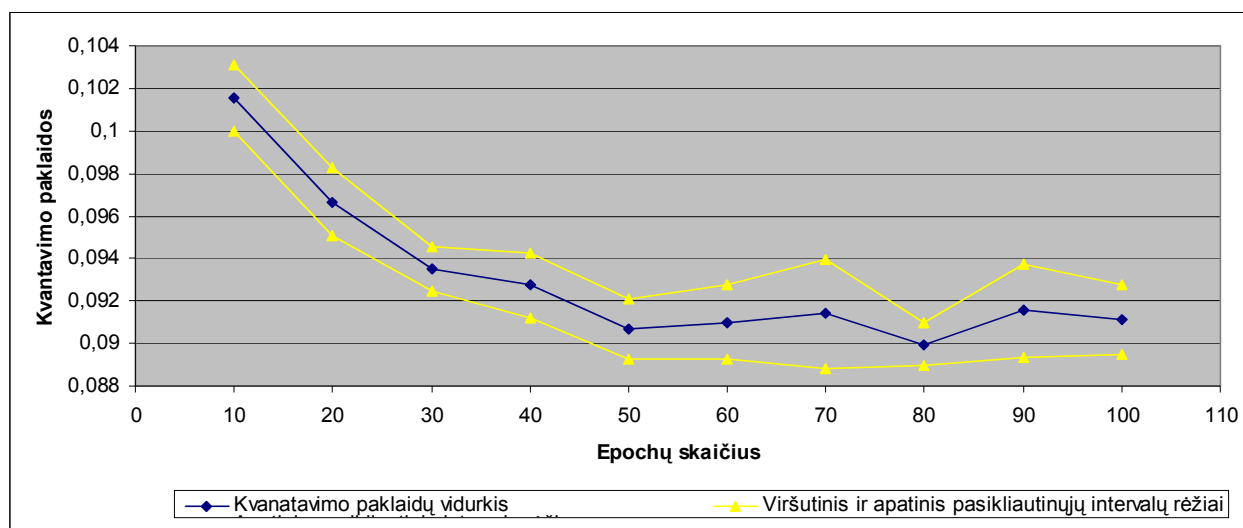
Irisų duomenų atveju didėjant epochų skaičiui, kvantavimo paklaida mažėja, tik ties 50 epochų padidėja (5.8 pav.). Topografinė paklaida labiau svyruoja nei kvantavimo, didėjant epochų skaičiui, tačiau matosi, kad 10-30 epochų dar nekanka norint gauti mažą kvantavimo paklaidą. 5.8 paveiksle pateiktas ne tik paklaidų vidurkiai, bet ir jų pasikliautinieji intervalai. Buvo atliktas pasikliautiniųjų intervalų skaičiavimas, tam kad pasižiūrėti galimus kvantavimo ir topografinės paklaidos vidurkio svyravimus. Tam panaudota MS Excel funkcija `confidence`. Mėlyna kreivė rodo kvantavimo ir topografinės paklaidos kitimą, geltona pasikliautiniųjų intervalų režius.

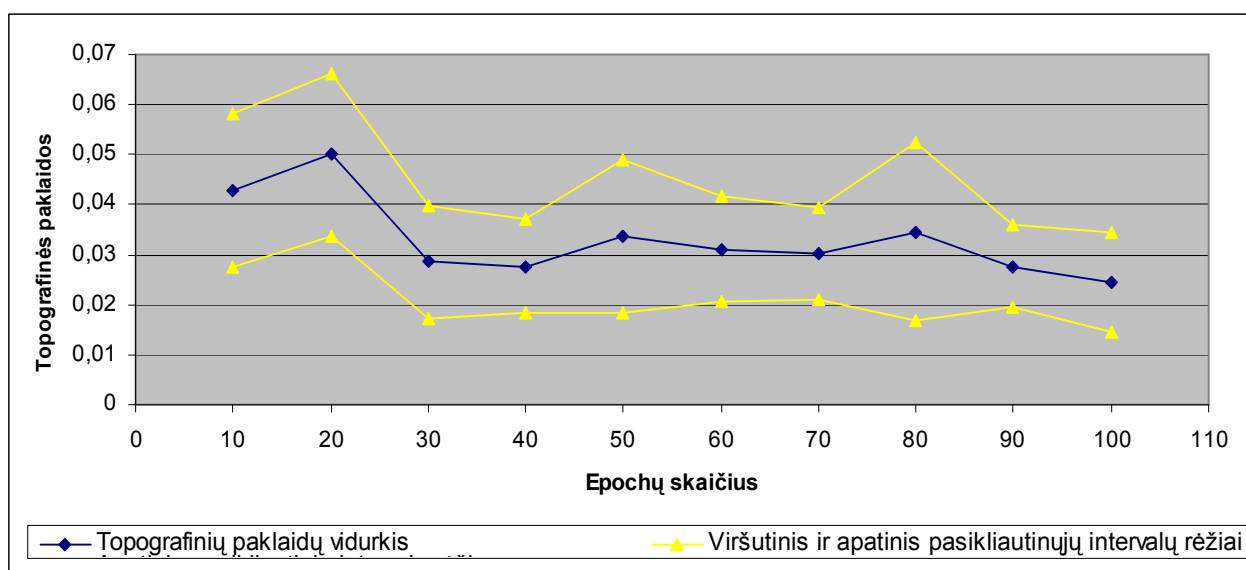




5.8 pav. Irisų duomenų kvantavimo ir topografinės paklaidų priklausomybė nuo epochų skaičiaus

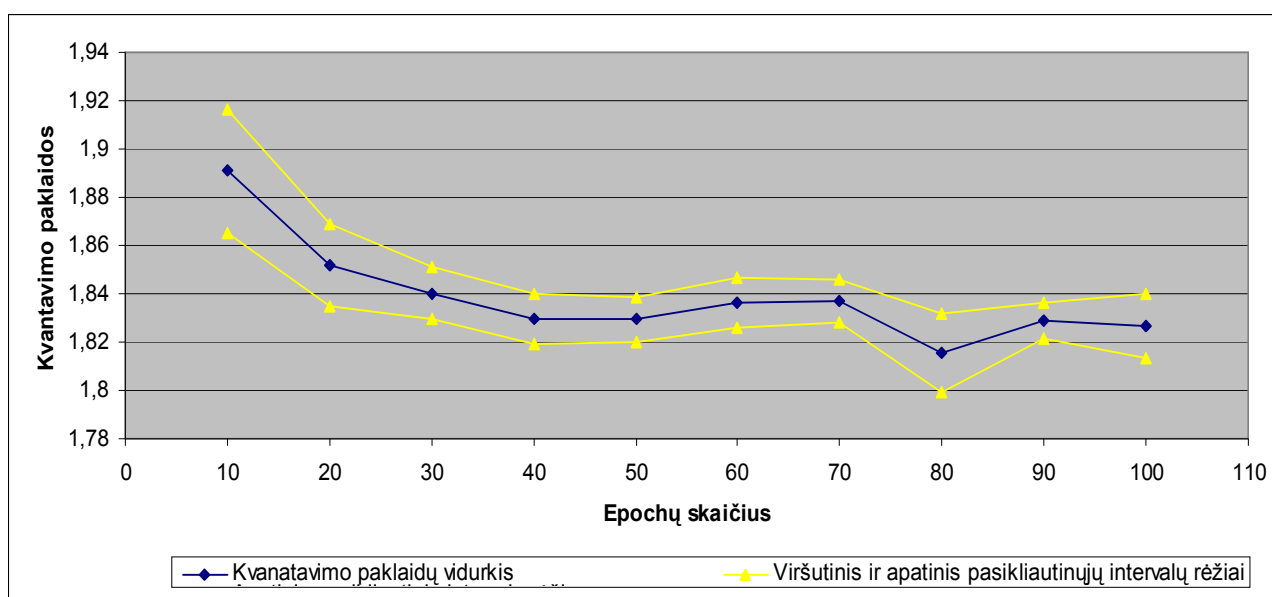
Stiklo duomenų kvantavimo paklaida taip pat mažėja didėjant epochų skaičiui, didesni svyravimai atsiranda, pasirinkus 70, 90 epochas (5.9 pav.). Topografinė paklaida svyruoja, tačiau šioji mažėjimo tendenciją išvelgti galima.

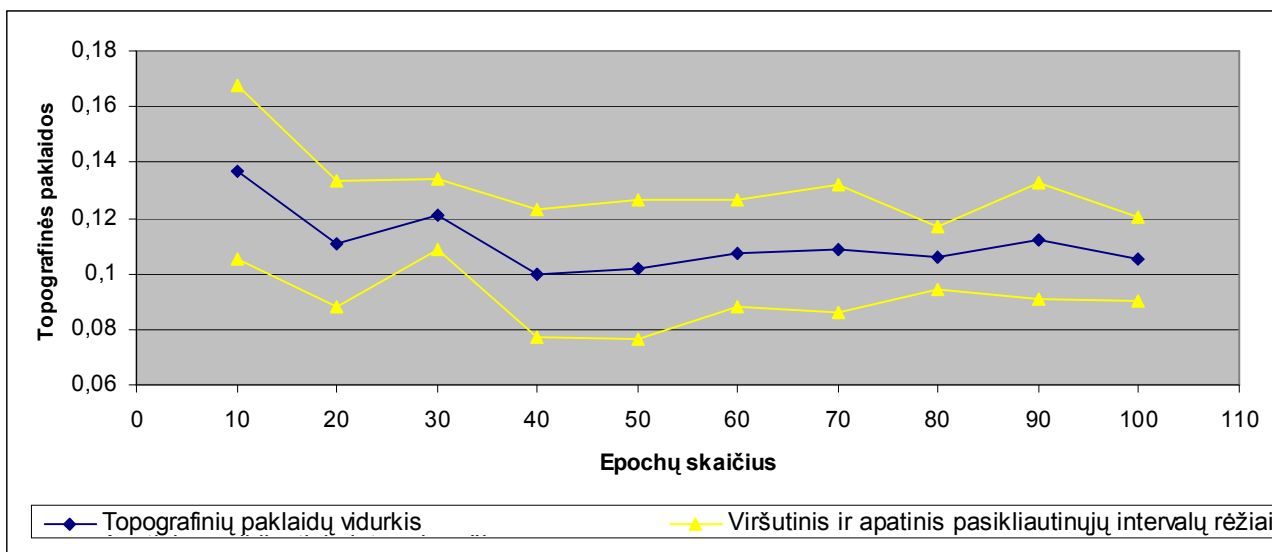




5.9 pav. Stiklo duomenų kvantavimo ir topografinės paklaidų priklausomybė nuo epochų skaičiaus

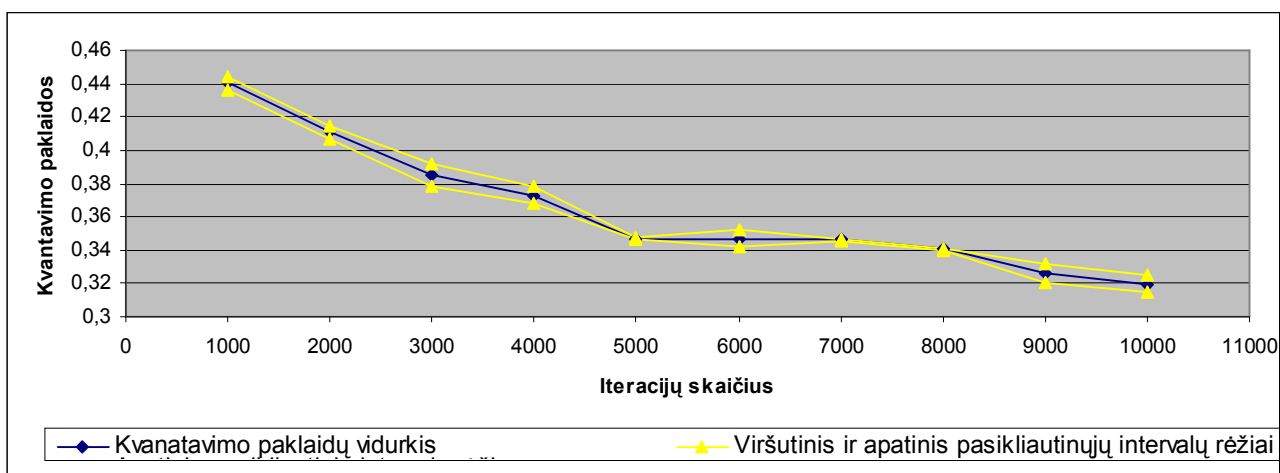
Atlikus bandymus su vyno duomenis kvantavimo paklaidos svyruoja maždaug nuo 1,78 iki 1,92. Didžiausi paklaidos kitimai pasirinkus 80 epochų, vėliau kvantavimo paklaida kinta nežymiai (5.10 pav.).

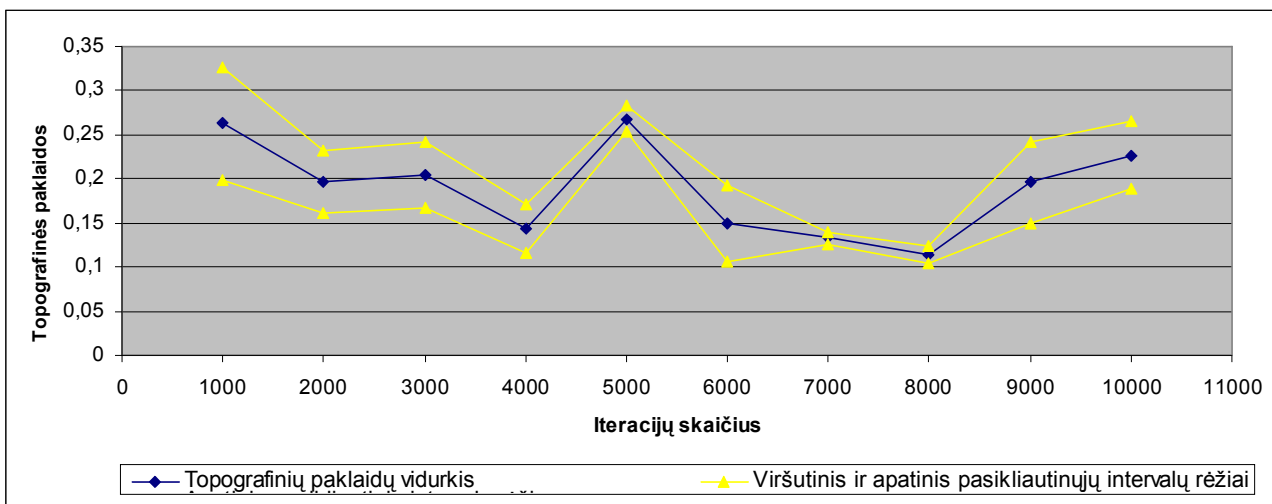




5.10 pav. Vyno duomenų kvantavimo ir topografinės paklaidų priklausomybė nuo epochų skaičiaus

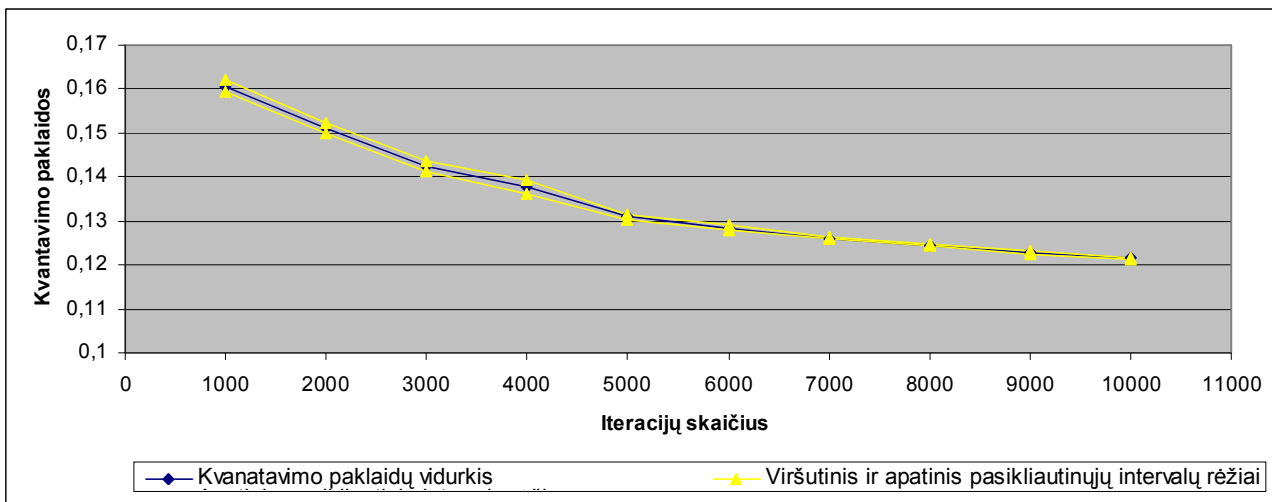
Buvo atliktas tyrimas ir sistemoje NeNet siekiant pažiūrėti kaip kvantavimo paklaida kinta priklausomai nuo mokymo žingsnių (iteracijų) skaičiaus. Šiame tyriame iteracijos buvo parenkamos nuo 1000 iki 10000, didinant kas 1000. Žemėlapiu dydis 10×10 , kaimynystės funkcija pasirinkta burbuliuko (Bubble). Duomenys naudoti tie patys. Atlikus tyrimus su irisų duomenimis (5.11 pav.) skirtingai negu naujai sukurtoje sistemoje čia svyravimai yra kur kas mažesni. Nedideli kvantavimo paklaidos svyravimai atsiranda tik pasirinkus 6000 iteracijų. Pasikliautinieji intervalai labai siauri. Topografinės paklaidos svyravimai yra didesni negu naujai sukurtoje sistemoje.

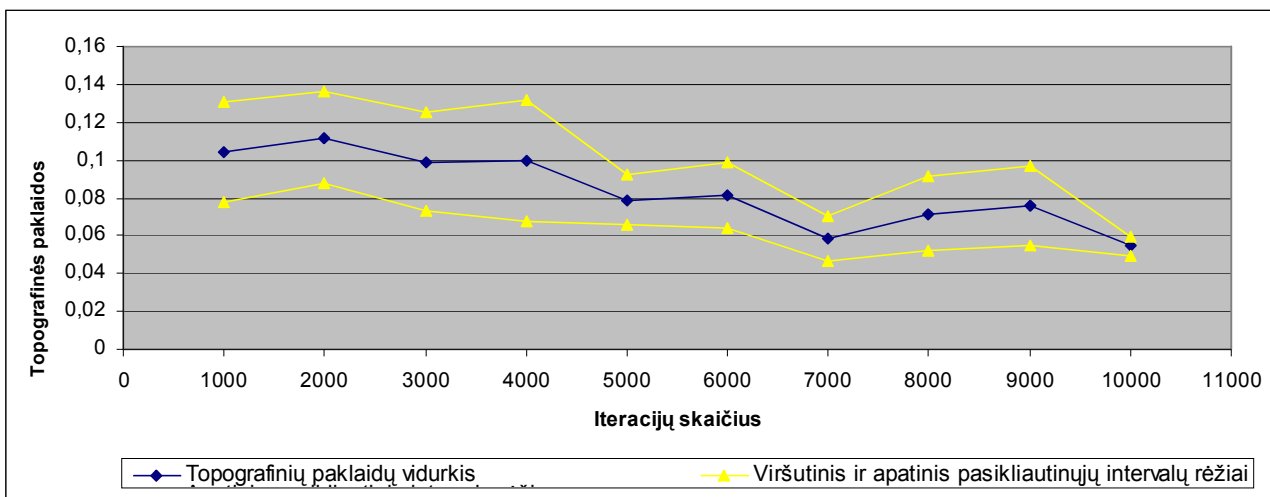




5.11 pav. Irisų duomenų kvantavimo ir topografinės paklaidų priklausomybė nuo iteracijų skaičiaus sistemoje NeNet

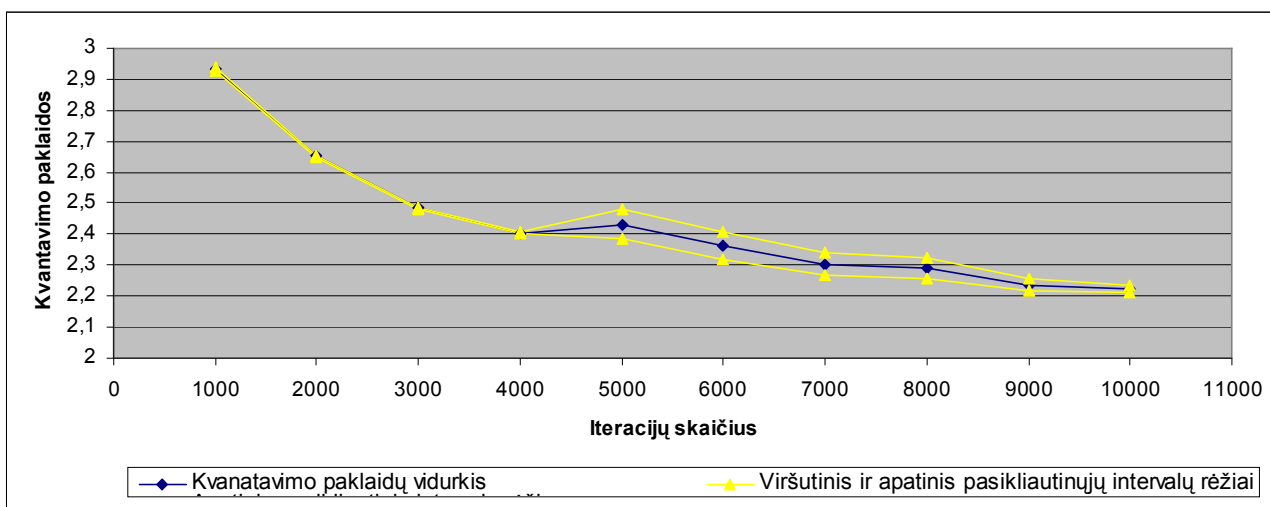
Analizuojant stiklo duomenis, taip pat kaip ir irisų duomenų atveju kvantavimo paklaidų svyravimų beveik nėra (5.12 pav.), tolygiai didėjant iteracijoms, kvantavimo ir topografinė paklaidos mažėja. Skirtingai negu irisų duomenų atveju topografinė paklaidą mažai svyruoja.

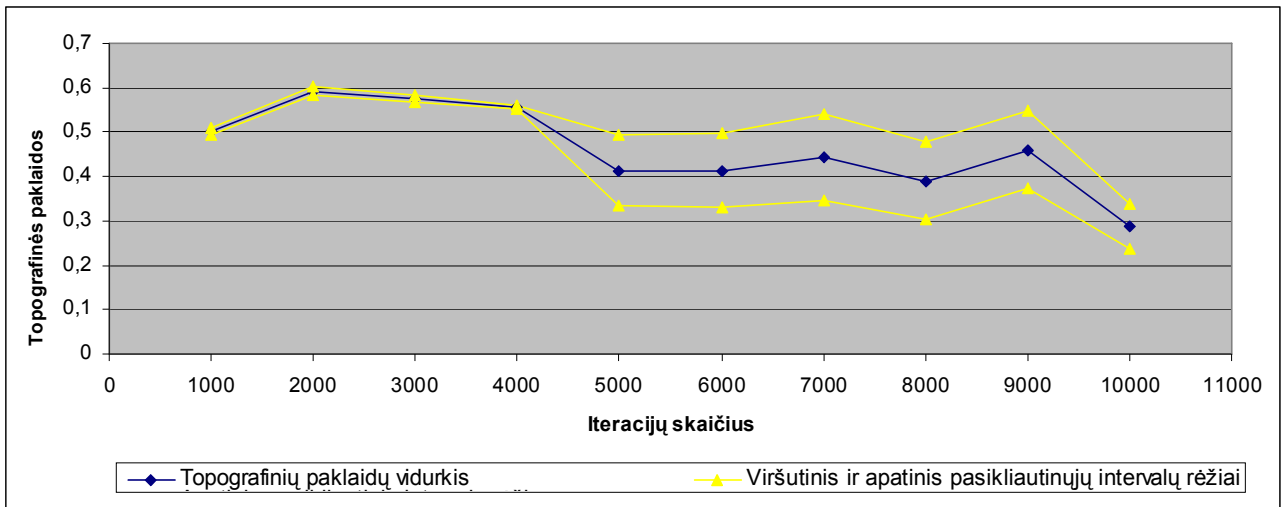




5.12 pav. Stiklo duomenų kvantavimo ir topografinės priklausomybė nuo iteracijų skaičiaus sistemoje NeNet

Vyno duomenų kvantavimo paklaida tolygiai mažėja nuo 1000 iki 4000 iteracijų (5.13 pav.), vėliau atsiranda nedideli svyravimai. Topografinė paklaida nuo 1000 iki 4000 iteracijų nežymiai svyruoja, vėliau atsiranda didesni svyravimai.





5.13 pav. Vyno duomenų kvantavimo ir topografinės priklausomybė nuo iteracijų skaičiaus sistemoje NeNet

5.4.2. Mokymo taisyklės modifikavimas

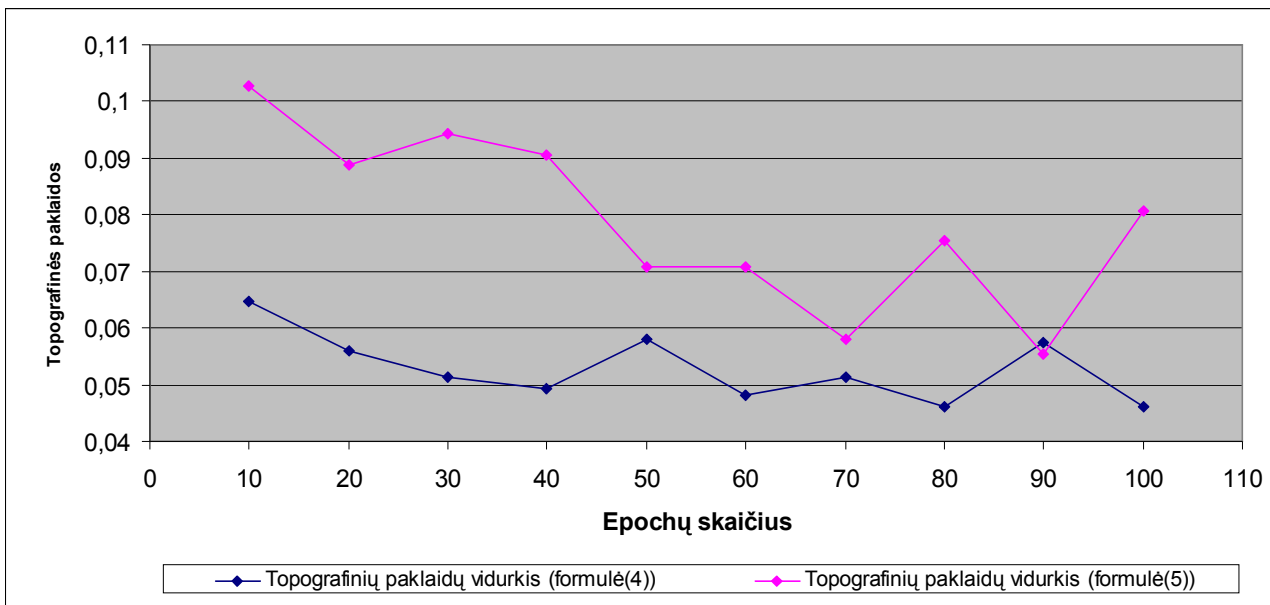
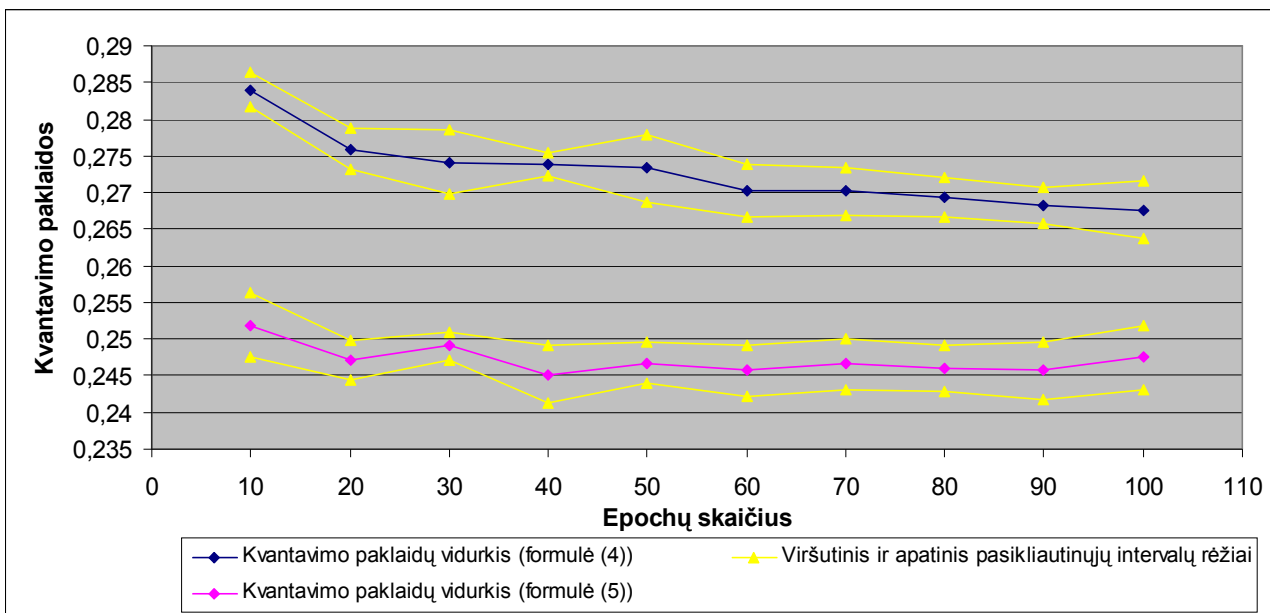
Taip pat buvo atliktas tyrimas, kuriame buvo pakeista kaimynystės formulė (4), siekiant sumažinti prieš tai gautas kvantavimo paklaidas. Po atliktų formulės (4) redagavimų, ji buvo pakeista į formulę:

$$h_{ij}^c = \frac{\alpha}{\sqrt{a(\eta_{ij}^c)^2 + 0,1}}, \text{ kur } \alpha = \max\left(\frac{\hat{e} + 0,1 - e}{\hat{e}}; 0,01\right) \quad (5)$$

Čia \hat{e} – pasirinktas epochų skaičius, e – einamosios epochos numeris, η_{ij}^c – kaimynystės tarp M_c ir M_{ij} eilė.

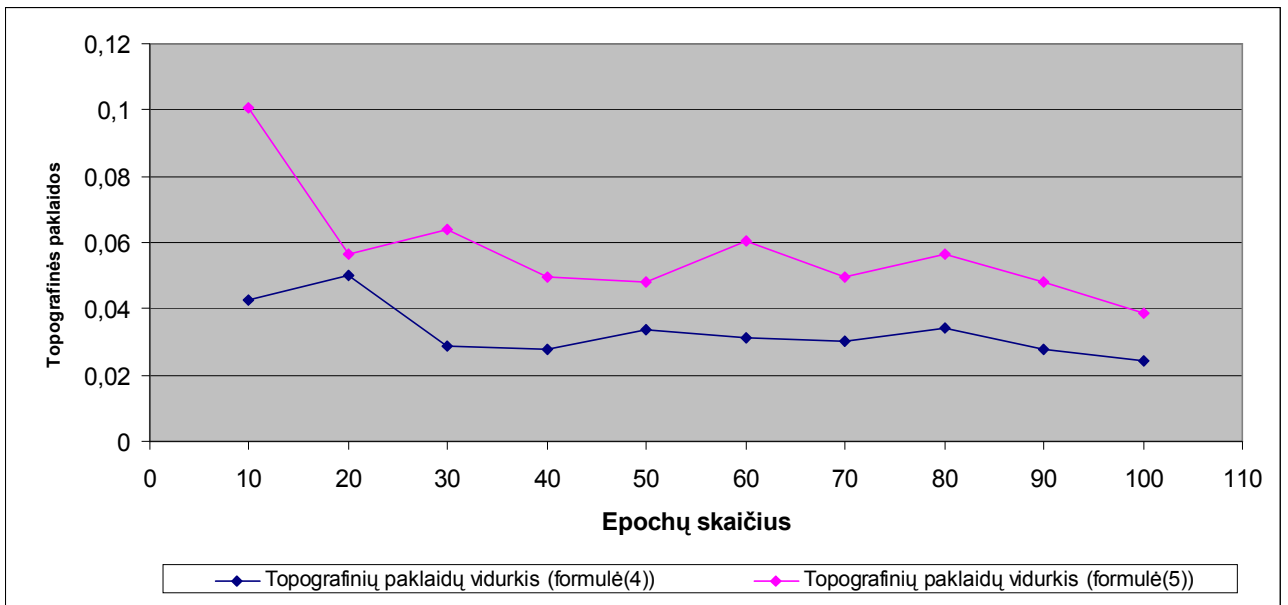
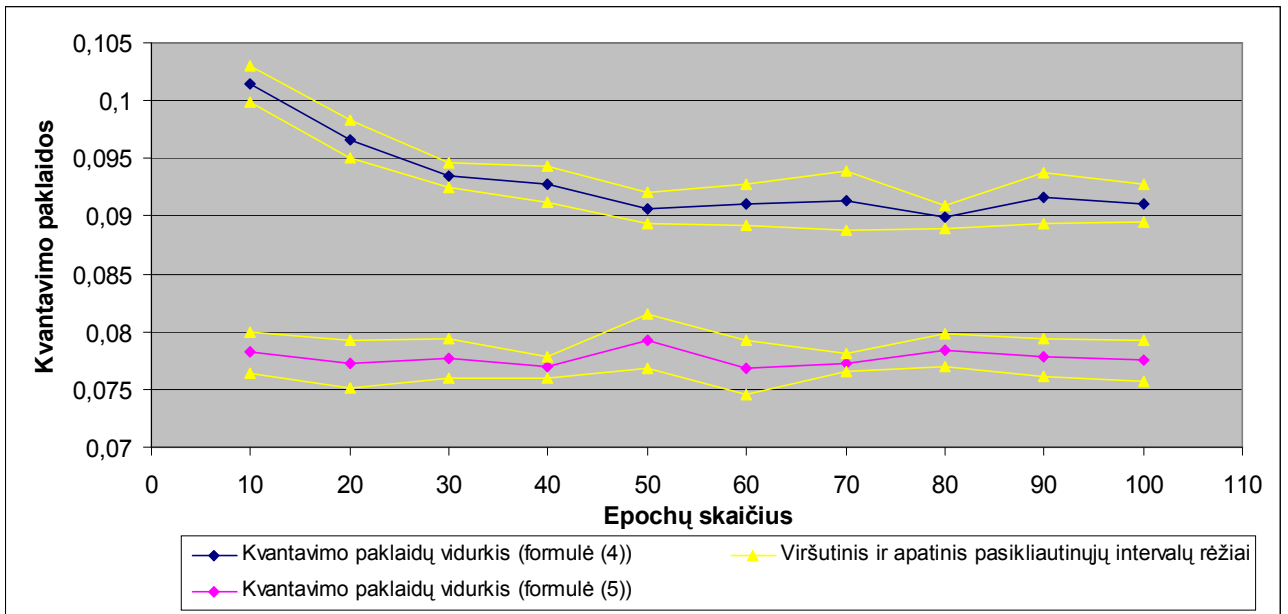
Po šių pakeitimų kvantavimo paklaidos lyginant su ankstesniu tyrimu sumažėjo, bet topografinės paklaidos padidėjo. Buvo atlikti analogiški bandymai naujai sukurtoje sistemoje, tik pakeitus kaimynystės formulę į (5).

Diagramoje (5.14) atvaizduotos gautos kvantavimo ir topografinės paklaidos naudojant formulę (4) ir formulę (5) irisų duomenų atveju. Mėlyna kreivė rodo kvantavimo paklaidos vidurkius skaičiuojant pagal formulę (4), apatinė (violetinė) pagal formulę (5). Matome, jog kvantavimo paklaidos sumažėjo apytiksliai 0,04. Tačiau topografinė paklaida padidėjo, naudojant naują kaimynystės formulę (5), be to pasikliautinieji intervalai persipina, todėl jie grafikuose nebuvo pavaizduoti.



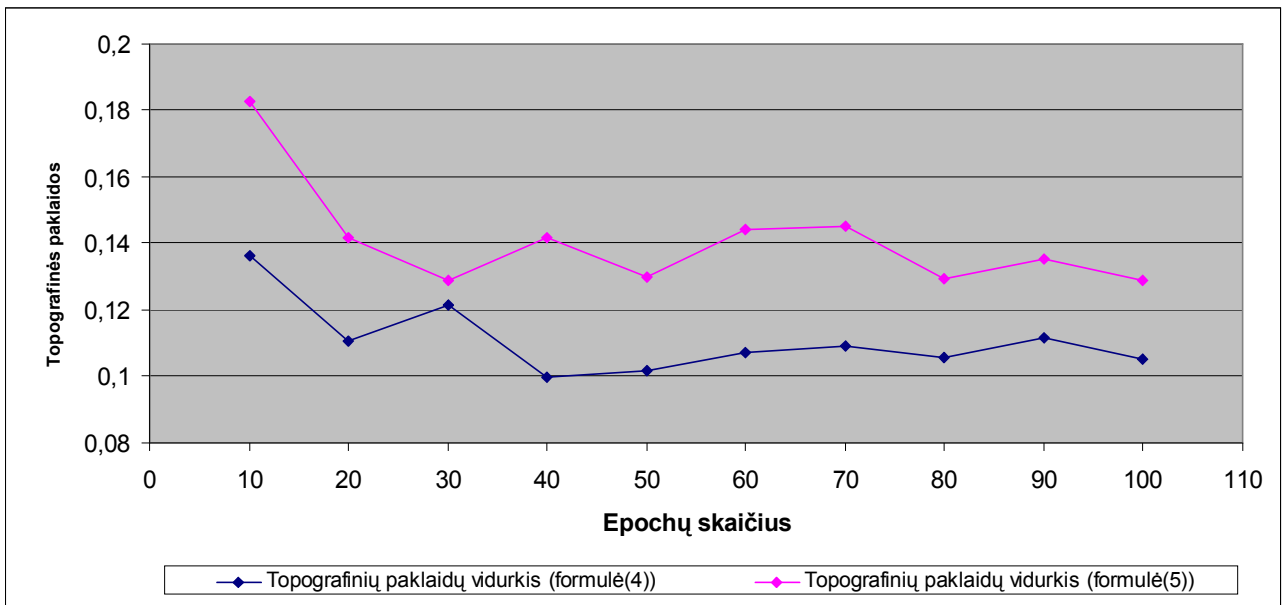
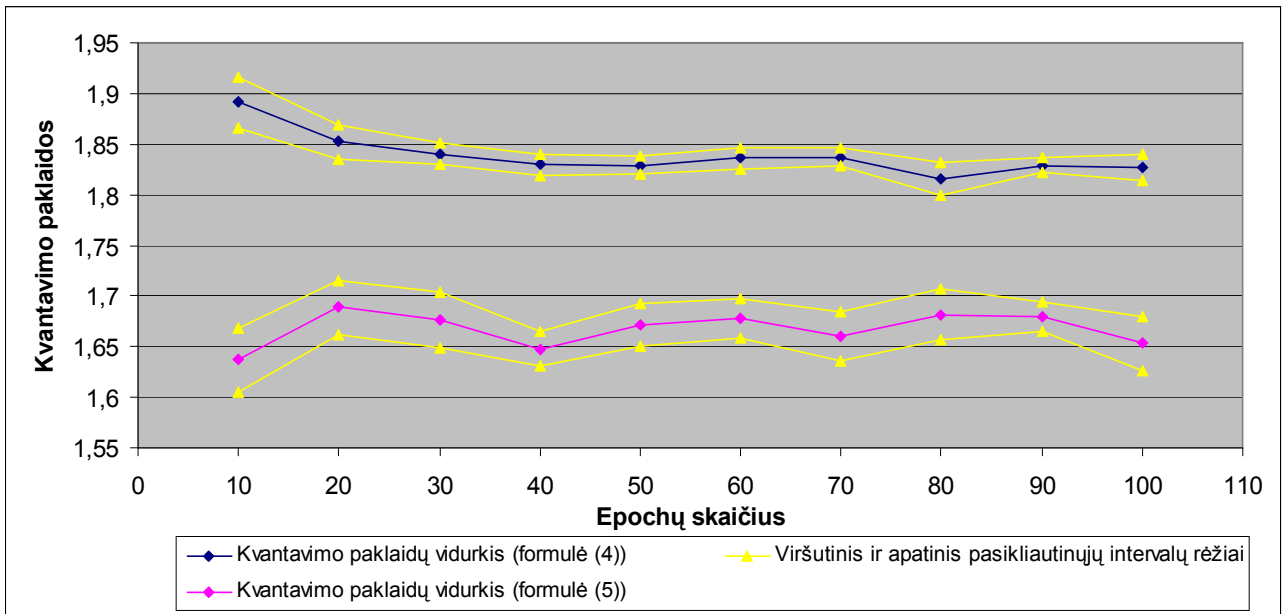
5.14 pav. Irisų duomenų kvantavimo ir topografinės paklaidos priklausomai nuo mokymo taisyklės

Stiklo duomenų atveju lygiai taip pat kvantavimo paklaidos sumažėjo (apytiksliai 0,02), o topografinės padidėjo.



5.15 pav. Stiklo duomenų kvantavimo ir topografinės paklaidos priklausomai nuo mokymo taisyklės

Vyno duomenų atveju taip pat kvantavimo paklaidos sumažėjo (apytiksliai 0,02), o topografinės padidėjo.



5.16 pav. Vyno duomenų kvantavimo ir topografinės paklaidos priklausomai nuo mokymo taisyklės

Matome, jog atlikus formulės pakeitimus kvantavimo paklaidos su visais duomenimis pagerėjo, tačiau topografinė paklaidos padidėjo vidutiniškai nuo 0,02 iki 0,06. Vienareikšmiškai teigti, jog formulė (4) yra geresnė nei formulė (3) negalime, būtini detalesni tyrimai

6 Išvados

Magistro darbą sudaro šios pagrindinės dalys: saviorganizuojančių neuroninių tinklų (žemėlapių) sistemų NeNet, SOM-Toolbox, Databionic ESOM, Viscovery SOMine analizė; sukurtos SOM sistemos, turinčios naują vizualizavimo būdą, analizė; gautų žemėlapių kokybės tyrimas (kvantavimo ir topografinių paklaidų priklausomybė nuo atliekamų epochų ar mokymo žingsnių skaičiaus). Tyrimai atlikti analizuojant tris daugiamačių duomenų aibes: irisų, stiklų ir vyno.

Atlikus tyrimus padarytos šios išvados:

1. Norint gauti daugiau informacijos apie duomenų klasterius, susidariusius SOM žemėlapyje, būtina naudoti kelias sistemas, kuriose realizuotos skirtingos tinklo mokymo taisyklės bei įvairūs žemėlapių vizualizavimo būdai.
2. Visos nagrinėtos sistemos analizuotus duomenis suklasterizavo panašiai, išskirdamos į tuos pačius klasterius, tik vienoje jie labiau išryškėja, kitoje prasčiau. Patogiausiai ir lengviausiai sukurti SOM žemėlapių NeNet sistema, todėl ji tinka pradedantiems tyrinėti SOM tinklus, bet kur kas daugiau galimybių ir profesionalesnė yra SOM-Toolbox sistema. Joje vartotojas pasinaudojus daugiau nei 200 funkcijų gali paruošti įvairius žemėlapius, gauti įvairių skaičiavimų rezultatus. Neseniai sukurta Databionic ESOM sistema patogi norint vizualizuoti didelius duomenų rinkinius, kuriuos galima atskirti skirtingais būdais, pritaikyti pagal vartotojo poreikius. Grafinė sąsaja patraukliausia yra Databionic ESOM sistema. Naudojant sistemą Viscovery SOMine galima nagrinėti SOM tinklus gana išsamiai. Šioje sistemoje daug skirtingų parametrų keitimų galimybių, tačiau valdymas ganėtinai sudėtingas, be to jos licenzija yra mokama, tačiau 30 dienų galima naudotis nemokamai.
3. Naujai sukurta SOM sistema gali tapti puikiu pagalbininku pradedančiam tyrėjui, kadangi jos valdymas yra labai paprastas, tačiau būtina „Matlab“ sistema ir minimalios žinios, kaip ja naudotis. Sistemoje realizuotas naujas vizualizavimo būdas, suteikiantis galimybę matyti skirtingų klasių duomenų, patekusių į tą patį SOM neuroną, santykį. Tai didelis privalumas lyginant su kitomis sistemomis, kadangi kitos nagrinėtos sistemos šios galimybės neturi. Sistema turi tik vieną nedidelį apribojimą – ji yra pritaikyta duomenims, kurių klasių skaičius ne didesnis nei 5, vizualizuoti, tačiau tą trukumą galima nesunkiai pašalinti.
4. Atlikti tyrimai parodė, jog didėjant epochų skaičiui, kvantavimo paklaida mažėja tiek naujai sukurtoje SOM sistemoje, tiek NeNet sistemoje. Dėl skirtingų epochų parinkimo metodų palyginti su kitomis sistemomis ganėtinai sunku.

5. Modifikavus mokymo taisyklę, realizuotą naujoje sistemoje, nustatyta, kad kvantavimo paklaida sumažėja, tačiau topografinė paklaida padidėja, be to pasikliautinų intervalų režiiai gana dideli, todėl rezultatai nėra patikimi. Būtinai išsamesni tyrimai.

7 Literatūros sąrašas

1. ASUNCION, A.; NEWMAN D. J. (2007). UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science [interaktyvus]. [žiūrėta 2009 m. gegužės 20 d.]. Prieiga per internetą: <<http://www.ics.uci.edu/~mllearn/MLRepository.html>>.
2. DZEMYDA, G.; KURASOVA, O. (2002). Comparative Analysis of the Graphical Result Presentation in the SOM Software. *Informatica*, Vol. 13(3), p. 275–286.
3. DZEMYDA G., KURASOVA O., ŽILINSKAS J. (2008) Daugiamatčių duomenų vizualizavimo metodai, 108-127 p.
4. HASSINEN, P.; ELOMAA, J.; RÖNKKÖ, J.; HALME, J.; HODJU, P. (1999). Neural Networks Tool – Nenet [interaktyvus] [žiūrėta 2009 m. gegužės 20 d.]. Prieiga per internetą: <<http://koti.mbnet.fi/~phodju/nenet/Nenet/General.html>>.
5. KOHONEN, T.(2001). Self-organizing Maps, 3rd ed., Springer series in information sciences. Berlin: Springer-Verlag, 506 p. ISBN 3540679219.
6. KURASOVA O., STEFANOVIČ P. (2009) Saviorganizuojančių neuroninių tinklų sistemų lyginamoji analizė. *Informacijos mokslai*. T. 50, p. 334–339. [interaktyvus] [žiūrėta 2010 m. kovo 4 d.]. Prieiga per internetą: <http://www.leidykla.eu/fileadmin/Informacijos_mokslai/50/334-339.pdf>.
7. MicroStat analytics Ltd (2006) [interaktyvus] [žiūrėta 2010 m. kovo 4 d.]. Prieiga per internetą: <<http://www.microstat-analytics.com/en/methods.php>>.
8. Neural Networks Toolbox 6 [interaktyvus] [žiūrėta 2010 m. kovo 17 d.]. Prieiga per internetą: <http://www.mathworks.com/access/helpdesk/help/pdf_doc/nnet/nnet.pdf>.
9. SMITH L. (2008), An Introduction to Neural Networks [interaktyvus] [žiūrėta 2010 m. kovo 5 d.]. Prieiga per internetą: <<http://www.cs.stir.ac.uk/~lss/NNIntro/InvSlides.html>>.
10. STERGIOU C., SIGANOS D. Neural Networks [interaktyvus] [žiūrėta 2010 m. kovo 5 d.]. Prieiga per internetą: <http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html>.
11. ULTSCH, A.; SIEMON, H. (1989). Exploratory Data Analysis: Using Kohonen Networks on Transputers. Technical Report 329, Univ. of Dortmund, Dortmund, Germany.
12. ULTSCH, A.; MOERCHEN, F. (2005): ESOM-Maps: Tools for Clustering, Visualization, and Classification with Emergent SOM, Technical Report Dept. of Mathematics and Computer Science, University of Marburg, Germany, No. 46, [interaktyvus] [žiūrėta 2009 m. gegužės 20 d.]. Prieiga per internetą: <<http://databionic-esom.sourceforge.net>>.

13. VESANTO, J.; HIMBERG, J.; ALHONIEMI E.; PARHANKANGAS, J. (2005) SOM Toolbox for Matlab 5 (2005). [interaktyvus] [žiūrėta 2009 m. gegužės 20 d.] Prieiga per internetą: <<http://www.cis.hut.fi/projects/somtoolbox/about.shtml>>.
14. Viscosity SOMine 5.0 (2010). [interaktyvus] [žiūrėta 2010 m. kovo 14 d.] Prieiga per internetą: <<http://www.viscosity.net/self-organizing-maps>>.
15. Viscosity SOMine 5.0 demo (2010). [interaktyvus] [žiūrėta 2010 m. kovo 14 d.] Prieiga per internetą: <http://www.viscosity.net/download/demo01/demo_SOMine_all.html>.

Saviorganizuojančių neuroninių tinklų (SOM) sistemų lyginamoji analizė

Šiame darbe pateikti ir aprašyti biologinio ir dirbtinio neurono modeliai. Didžiausias dėmesys skiriamas vieno tipo neuroniniams tinklams – saviorganizuojantiems žemėlapiams (SOM). Darbe pateiktas jų apmokymas, taip pat pagrindinių sąvokų (epocha, kaimynystės eilė, unifikuotų atstumų matrica ir kt.), susijusių su SOM neuroniniais tinklais (žemėlapiams), apibrėžimai. Buvo nagrinėtos keturios saviorganizuojančių neuroninių tinklų sistemos: NeNet, SOM-Toolbox, DataBionic ESOM, Viscovery SOMine ir Matlab įrankiai „nntool“, „nctool“, kurie naudojami SOM tinklams sukurti ir apmokyti. Pateikiamos sistemų naudojimosi instrukcijos, norint gauti paprasčiausią SOM žemėlapi. Matlab aplinkoje sukurta ir darbe aprašyta naują vizualizavimo būdą turinti sistema „Somas“, pateiktas jos išskirtinumas ir naudojimosi instrukcija. Sistemoje „Somas“ realizuota kita mokymo funkcija nei kitose minėtose sistemose. Pagrindinis analizuotų sistemų tikslas yra suskirstyti duomenis į klasterius pagal jų panašumą ir pateikti juos SOM žemėlapyje. Sistemos viena nuo kitos skiriasi duomenų pateikimu, mokymo taisyklėmis, vizualizavimo galimybėmis, todėl čia aptariami sistemų panašumai ir skirtumai. Nagrinėti susidarę SOM žemėlapiai ir gautos kvantavimo bei topografinės paklaidos, analizuojant tris duomenų aibes: irisų, stiklo ir vyno. Kvantavimo ir topografinės paklaidos yra kiekybiniai vaizdo kokybės įverčiai. Padarytos išvados apie susidariusius klasterius tiriamuose duomenyse. Naudojant naują sistemą „Somas“ ir sistemą „NeNet“ buvo atliktas tyrimas, siekiant pasižiūrėti, kaip kinta kvantavimo ir topografinė paklaidos, didėjant epochų arba iteracijų skaičiui. Taip pat sistemoje „Somas“ buvo ieškoma geresnės kaimynystės funkcijos išraiškos, su kuria paklaidos taptų mažesnės.

Summary

The comparative analysis of the self-organizing map software

In this master thesis, biologic and artificial neuron models have been described. The focus is self-organizing maps (SOM). The self-organizing maps are one of types of artificial neural networks. SOM training as well as the main concepts which need to explain SOM networks (epochs, neighbourhood size, u-matrix and etc.) have been described. Four systems of self-organizing maps: NeNet, SOM-Toolbox, DataBionic ESOM, Viscovery SOMine, and Matlab tools “nntool” and “nctool” have been analyzed. In the thesis, a system use guide has been presented to make a simple SOM map. A new system “Somas” that has a new visualisation way has been developed in Matlab. The system has been described, its oneness has been emphasized, and a use guide is presented. The main target of the SOM systems is data clustering and their graphical presentation on the self-organizing map. The SOM systems are different one from other in their interfaces, the data pre-processing, learning rules, visualization manners, etc. Similarities and differences of the systems have been highlighted here. The experiments have been carried out with three data sets: iris, glass and wine. The SOM maps, obtained by each system, have been described and some conclusions on the clusters have been drawn. The quantization and topographic errors have been analyzed to estimate the quality of the maps obtained. An investigation has been carried out in the new system “Somas” and system “NeNet” in order to look how quantization and topographic errors vary after changing the number of learning epochs or iterations. The better neighbourhood function that gives lower errors has also been searched.

1 priedas: duomenys.

<i>Irisų duomenys</i>					<i>Stiklo duomenys</i>						
5,1	3,5	1,4	0,2	1	0,4328	0,4376	1	0,2523	0,3518	0,0097	1
4,9	3	1,4	0,2	1	0,2836	0,4752	0,8018	0,3333	0,5214	0,0773	1
4,7	3,2	1,3	0,2	1	0,2208	0,4211	0,7906	0,3894	0,5679	0,0628	1
4,6	3,1	1,5	0,2	1	0,2858	0,3729	0,8218	0,3115	0,5	0,0918	1
5	3,6	1,4	0,2	1	0,2752	0,382	0,8062	0,296	0,5839	0,0886	1
5,4	3,9	1,7	0,4	1	0,2112	0,3098	0,804	0,4143	0,5643	0,1031	1
4,6	3,4	1,4	0,3	1	0,2757	0,3865	0,8018	0,2648	0,5857	0,0934	1
5	3,4	1,5	0,2	1	0,2814	0,3639	0,804	0,2368	0,6125	0,0918	1
4,4	2,9	1,4	0,2	1	0,3525	0,4977	0,7973	0,3364	0,4054	0,0902	1
4,9	3,1	1,5	0,1	1	0,2809	0,3414	0,8018	0,3333	0,5679	0,0918	1
5,4	3,7	1,5	0,2	1	0,2002	0,2992	0,7706	0,3956	0,6054	0,1079	1
4,8	3,4	1,6	0,2	1	0,2845	0,3113	0,8151	0,3053	0,5714	0,0966	1
4,8	3	1,4	0,1	1	0,2081	0,3233	0,7639	0,3458	0,6196	0,1111	1
4,3	3	1,1	0,1	1	0,2779	0,3203	0,7929	0,3053	0,6071	0,087	1
5,8	4	1,2	0,2	1	0,2845	0,2827	0,7996	0,3178	0,6214	0,0934	1
5,7	4,4	1,5	0,4	1	0,2836	0,3128	0,7884	0,2928	0,6125	0,0934	1
5,4	3,9	1,3	0,4	1	0,2937	0,2932	0,8174	0,271	0,5893	0,0982	1
5,1	3,5	1,4	0,3	1	0,4745	0,5459	0,8575	0,1869	0,2768	0,0242	1
5,7	3,8	1,7	0,3	1	0,3494	0,4767	0,8307	0,2773	0,4125	0,0097	1
5,1	3,8	1,5	0,3	1	0,2722	0,3444	0,7884	0,4361	0,5214	0,087	1
5,4	3,4	1,7	0,2	1	0,2788	0,3143	0,7906	0,3738	0,525	0,087	1
5,1	3,7	1,5	0,4	1	0,3736	0,6075	0,8352	0	0,3946	0,0048	1
4,6	3,6	1	0,2	1	0,2726	0,3083	0,8062	0,3115	0,5321	0,095	1
5,1	3,3	1,7	0,5	1	0,2792	0,3128	0,7951	0,3302	0,5732	0,0998	1
4,8	3,4	1,9	0,2	1	0,2656	0,3985	0,7795	0,2679	0,5429	0,0805	1
5	3	1,6	0,2	1	0,2849	0,3383	0,7884	0,2866	0,5696	0,1047	1
5	3,4	1,6	0,4	1	0,2976	0,3729	0,7751	0,3489	0,5054	0,095	1
5,2	3,5	1,5	0,2	1	0,266	0,3218	0,7751	0,324	0,5768	0,0902	1
5,2	3,4	1,4	0,2	1	0,2867	0,2752	0,784	0,3551	0,5964	0,0918	1
4,7	3,2	1,6	0,2	1	0,2937	0,3534	0,7773	0,3084	0,5446	0,0966	1
4,8	3,1	1,6	0,2	1	0,2867	0,2887	0,7929	0,3146	0,5839	0,0982	1
5,4	3,4	1,5	0,4	1	0,2774	0,3173	0,7795	0,2648	0,6179	0,0902	1
5,2	4,1	1,5	0,1	1	0,2897	0,3188	0,7751	0,2928	0,5643	0,0982	1
5,5	4,2	1,4	0,2	1	0,2801	0,2767	0,7728	0,3396	0,6393	0,0966	1
4,9	3,1	1,5	0,2	1	0,2932	0,2947	0,7884	0,3271	0,5607	0,0918	1
5	3,2	1,2	0,2	1	0,1984	0,385	0,7684	0,2866	0,5232	0,0902	1
5,5	3,5	1,3	0,2	1	0,3486	0,4752	0,7862	0,3209	0,3571	0,0821	1
4,9	3,6	1,4	0,1	1	0,2994	0,3023	0,7751	0,3302	0,5625	0,1031	1
4,4	3	1,3	0,2	1	0,482	0,5233	0,8508	0,0561	0,35	0,0177	1
5,1	3,4	1,5	0,2	1	0,482	0,5233	0,8508	0,0561	0,35	0,0177	1
5	3,5	1,3	0,3	1	0,2976	0,3098	0,7795	0,2586	0,575	0,1031	1
4,5	2,3	1,3	0,3	1	0,2809	0,2977	0,7617	0,2835	0,6054	0,095	1
4,4	3,2	1,3	0,2	1	0,2915	0,3729	0,755	0,324	0,5268	0,095	1
5	3,5	1,6	0,6	1	0,4807	0,4511	0,8552	0,134	0,3482	0,0274	1
5,1	3,8	1,9	0,4	1	0,2946	0,3008	0,7639	0,2804	0,5607	0,0998	1
4,8	3	1,4	0,3	1	0,3446	0,415	0,7751	0,3302	0,3821	0,0886	1

5,1	3,8	1,6	0,2	1	0,331	0,3699	0,7506	0,2773	0,5196	0,0918	1
4,6	3,2	1,4	0,2	1	0,6813	0,4902	0,8241	0,1308	0,3143	0,0032	1
5,3	3,7	1,5	0,2	1	0,4864	0,3729	0,8396	0,1558	0,3893	0,0209	1
5	3,3	1,4	0,2	1	0,3437	0,4286	0,7461	0,2928	0,4054	0,095	1
7	3,2	4,7	1,4	2	0,529	0,4496	0,8285	0,0685	0,3464	0,0145	1
6,4	3,2	4,5	1,5	2	0,356	0,3714	0,7416	0,3084	0,4554	0,0966	1
6,9	3,1	4,9	1,5	2	0,3042	0,406	0,6392	0,2804	0,5411	0,0886	1
5,5	2,3	4	1,3	2	0,3169	0,3624	0,6325	0,3084	0,5429	0,0886	1
6,5	2,8	4,6	1,5	2	0,291	0,3729	0,6258	0,3115	0,5661	0,0821	1
5,7	2,8	4,5	1,3	2	0,2871	0,2586	0,6036	0,3115	0,6946	0,0902	1
6,3	3,3	4,7	1,6	2	0,0439	0,3398	0,7728	0,2586	0,5661	0,0998	1
4,9	2,4	3,3	1	2	0,3112	0,3218	0,7751	0,3115	0,5607	0,0966	1
6,6	2,9	4,6	1,3	2	0,2805	0,4135	0,833	0,2741	0,5679	0,095	1
5,2	2,7	3,9	1,4	2	0,2805	0,4	0,8151	0,2804	0,5321	0,0918	1
5	2	3,5	1	2	0,3468	0,4316	0,8062	0,2555	0,5054	0,0225	1
5,9	3	4,2	1,5	2	0,3784	0,4632	0,7973	0,3209	0,3411	0,0193	1
6	2,2	4	1	2	0,464	0,418	0,8597	0,1838	0,3536	0,037	1
6,1	2,9	4,7	1,4	2	0,4881	0,5173	0,8486	0,1526	0,275	0	1
5,6	2,9	3,6	1,3	2	0,464	0,4135	0,833	0,19	0,3929	0,029	1
6,7	3,1	4,4	1,4	2	0,432	0,4451	0,7996	0,2586	0,3839	0,0145	1
5,6	3	4,5	1,5	2	0,4552	0,3489	0,8129	0,1807	0,4304	0,0306	1
5,8	2,7	4,1	1	2	0,4552	0,3489	0,8129	0,1807	0,4482	0,0306	1
6,2	2,2	4,5	1,5	2	0,4552	0,3594	0,7973	0,19	0,4268	0,037	1
5,6	2,5	3,9	1,1	2	0,5202	0,388	0,7973	0,1651	0,3893	0,0193	1
5,9	3,2	4,8	1,8	2	0,1752	0,4932	0,5969	1	0,0143	0,2705	2
6,1	2,8	4	1,3	2	0,3512	0,3008	0,412	0,4891	0,5143	0,0966	2
6,3	2,5	4,9	1,5	2	0,4636	0,1248	0,4187	0,3956	0,5446	0,0757	2
6,1	2,8	4,7	1,2	2	0,4548	0,0451	0,3808	0,3956	0,6482	0,0934	2
6,4	2,9	4,3	1,3	2	0,3749	0,2872	0	0,4237	0,7036	0,0612	2
6,6	3	4,4	1,4	2	0,2419	0,3203	0	0,4798	0,7268	0,1562	2
6,8	2,8	4,8	1,4	2	0,3859	0,382	0	0,4579	0,575	0,0757	2
6,7	3	5	1,7	2	0,5505	0,4075	0	0,4019	0,4304	0,0515	2
6	2,9	4,5	1,5	2	0,0882	0,3444	0	0,8567	0,1196	1	2
5,7	2,6	3,5	1	2	0,0904	0,3414	0	0,8505	0,1589	1	2
5,5	2,4	3,8	1,1	2	0,4074	0,3985	0	0,3458	0,4357	0,0531	2
5,5	2,4	3,7	1	2	0,414	0,3188	0,3586	0,5857	0,4232	0,1224	2
5,8	2,7	3,9	1,2	2	0,4407	0,3368	0,0735	0,3801	0,6393	0,0209	2
6	2,7	5,1	1,6	2	0,3468	0,4917	0,5323	0,3956	0,4571	0	3
5,4	3	4,5	1,5	2	0,3608	0,4602	0,5367	0,2804	0,5268	0	3
6	3,4	4,5	1,6	2	0,3134	0,5609	0,4989	0,4143	0,4589	0	3
6,7	3,1	4,7	1,5	2	0,3235	0,5053	0,4878	0,4268	0,5107	0	3
6,3	2,3	4,4	1,3	2	0,0808	0,5519	0,3875	0,3894	0,8464	0	3
5,6	3	4,1	1,3	2	0,3393	0,6406	0,1737	0,4517	0,4804	0	3
5,5	2,5	4	1,3	2	0,3516	0,5143	0	0,5607	0,5232	0	3
5,5	2,6	4,4	1,2	2	0,3749	0,5759	0	0,0841	0,6554	0	3
6,1	3	4,6	1,4	2	0	1	0	0,0156	1	0	3
5,8	2,6	4	1,2	2	0,3174	0,5398	0,7261	0,6012	0,2571	0,2351	4
5	2,3	3,3	1	2	0,4969	0,6211	0,49	0,5514	0,0804	0,1224	4
5,6	2,7	4,2	1,3	2	0,5487	0,7609	0,4076	0,3178	0,1107	0,0499	4
5,7	3	4,2	1,2	2	0,2186	0,4737	0,3964	0,4673	0,5875	0	4
5,7	2,9	4,2	1,3	2	0,2138	0,6195	0	0,6511	0,6196	0	4
6,2	2,9	4,3	1,3	2	0,223	0,5218	0	0,7788	0,6518	0,0064	4
5,1	2,5	3	1,1	2	0,2651	0,6045	0	0,5327	0,5732	0	4

5,7	2,8	4,1	1,3	2	0,2493	0,5759	0	0,5265	0,6214	0	4
6,3	3,3	6	2,5	3	0,1888	0,5128	0	0,7445	0,6393	0,0129	4
5,8	2,7	5,1	1,9	3	0,1936	0,4722	0	0,7009	0,6107	0,0225	4
7,1	3	5,9	2,1	3	0,2687	0,597	0	0,6386	0,6196	0	4
6,3	2,9	5,6	1,8	3	0,1826	0,5489	0	0,7383	0,5875	0,0064	4
6,5	3	5,8	2,2	3	0,2169	0,6436	0	0,6916	0,5786	0,0081	4
7,6	3	6,6	2,1	3	0,1725	0,6647	0	0,6106	0,6589	0	4
4,9	2,5	4,5	1,7	3	0,2362	0,1835	0	0,2804	0,9589	0,4348	4
7,3	2,9	6,3	1,8	3	0,1752	0,6195	0	0,6636	0,6982	0	4
6,7	2,5	5,8	1,8	3	0,2384	0,612	0	0,5296	0,5893	0	4
7,2	3,6	6,1	2,5	3	0,2204	0,6346	0	0,6168	0,6232	0	4
6,5	3,2	5,1	2	3	0,2709	0,6346	0	0,4704	0,5679	0	4
6,4	2,7	5,3	1,9	3	0,2327	0,6331	0	0,4922	0,5893	0	4
6,8	3	5,5	2,1	3	0,3143	0,5504	0	0,4766	0,5446	0,2271	4
5,7	2,5	5	2	3	0,2305	0,5474	0	0,7632	0,5429	0	4
5,8	2,8	5,1	2,4	3	0,223	0,5128	0	0,8069	0,5	0,0129	4
6,4	3,2	5,3	2,3	3	0,2502	0,6301	0	0,5296	0,5804	0	4
6,5	3	5,5	1,8	3	0,417	0,5459	0	0,5389	0,6446	0	4
7,7	3,8	6,7	2,2	3	0,2353	0,5489	0	0,514	0,6786	0	4
7,7	2,6	6,9	2,3	3	0,2616	0,5263	0	0,5576	0,6339	0	4
6	2,2	5	1,5	3							
6,9	3,2	5,7	2,3	3							
5,6	2,8	4,9	2	3							
7,7	2,8	6,7	2	3							
6,3	2,7	4,9	1,8	3							
6,7	3,3	5,7	2,1	3							
7,2	3,2	6	1,8	3							
6,2	2,8	4,8	1,8	3							
6,1	3	4,9	1,8	3							
6,4	2,8	5,6	2,1	3							
7,2	3	5,8	1,6	3							
7,4	2,8	6,1	1,9	3							
7,9	3,8	6,4	2	3							
6,4	2,8	5,6	2,2	3							
6,3	2,8	5,1	1,5	3							
6,1	2,6	5,6	1,4	3							
7,7	3	6,1	2,3	3							
6,3	3,4	5,6	2,4	3							
6,4	3,1	5,5	1,8	3							
6	3	4,8	1,8	3							
6,9	3,1	5,4	2,1	3							
6,7	3,1	5,6	2,4	3							
6,9	3,1	5,1	2,3	3							
5,8	2,7	5,1	1,9	3							
6,8	3,2	5,9	2,3	3							
6,7	3,3	5,7	2,5	3							
6,7	3	5,2	2,3	3							
6,3	2,5	5	1,9	3							
6,5	3	5,2	2	3							
6,2	3,4	5,4	2,3	3							
5,9	3	5,1	1,8	3							

Vyno duomenys

14,23	1,71	127	2,8	5,64	1,04	1
13,2	1,78	100	2,65	4,38	1,05	1
13,16	2,36	101	2,8	5,68	1,03	1
14,37	1,95	113	3,85	7,8	0,86	1
13,24	2,59	118	2,8	4,32	1,04	1
14,2	1,76	112	3,27	6,75	1,05	1
14,39	1,87	96	2,5	5,25	1,02	1
14,06	2,15	121	2,6	5,05	1,06	1
14,83	1,64	97	2,8	5,2	1,08	1
13,86	1,35	98	2,98	7,22	1,01	1
14,1	2,16	105	2,95	5,75	1,25	1
14,12	1,48	95	2,2	5	1,17	1
13,75	1,73	89	2,6	5,6	1,15	1
14,75	1,73	91	3,1	5,4	1,25	1
14,38	1,87	102	3,3	7,5	1,2	1
13,63	1,81	112	2,85	7,3	1,28	1
14,3	1,92	120	2,8	6,2	1,07	1
13,83	1,57	115	2,95	6,6	1,13	1
14,19	1,59	108	3,3	8,7	1,23	1
13,64	3,1	116	2,7	5,1	0,96	1
14,06	1,63	126	3	5,65	1,09	1
12,93	3,8	102	2,41	4,5	1,03	1
13,71	1,86	101	2,61	3,8	1,11	1
12,85	1,6	95	2,48	3,93	1,09	1
13,5	1,81	96	2,53	3,52	1,12	1
13,05	2,05	124	2,63	3,58	1,13	1
13,39	1,77	93	2,85	4,8	0,92	1
13,3	1,72	94	2,4	3,95	1,02	1
13,87	1,9	107	2,95	4,5	1,25	1
14,02	1,68	96	2,65	4,7	1,04	1
13,73	1,5	101	3	5,7	1,19	1
13,58	1,66	106	2,86	6,9	1,09	1
13,68	1,83	104	2,42	3,84	1,23	1
13,76	1,53	132	2,95	5,4	1,25	1
13,51	1,8	110	2,35	4,2	1,1	1
13,48	1,81	100	2,7	5,1	1,04	1
13,28	1,64	110	2,6	4,6	1,09	1
13,05	1,65	98	2,45	4,25	1,12	1
13,07	1,5	98	2,4	3,7	1,18	1
14,22	3,99	128	3	5,1	0,89	1
13,56	1,71	117	3,15	6,13	0,95	1
13,41	3,84	90	2,45	4,28	0,91	1
13,88	1,89	101	3,25	5,43	0,88	1
13,24	3,98	103	2,64	4,36	0,82	1
13,05	1,77	107	3	5,04	0,88	1
14,21	4,04	111	2,85	5,24	0,87	1
14,38	3,59	102	3,25	4,9	1,04	1
13,9	1,68	101	3,1	6,1	0,91	1

14,1	2,02	103	2,75	6,2	1,07	1
13,94	1,73	108	2,88	8,9	1,12	1
13,05	1,73	92	2,72	7,2	1,12	1
13,83	1,65	94	2,45	5,6	1,24	1
13,82	1,75	111	3,88	7,05	1,01	1
13,77	1,9	115	3	6,3	1,13	1
13,74	1,67	118	2,6	5,85	0,92	1
13,56	1,73	116	2,96	6,25	0,98	1
14,22	1,7	118	3,2	6,38	0,94	1
13,29	1,97	102	3	6	1,07	1
13,72	1,43	108	3,4	6,8	0,89	1
12,37	0,94	88	1,98	1,95	1,05	2
12,33	1,1	101	2,05	3,27	1,25	2
12,64	1,36	100	2,02	5,75	0,98	2
13,67	1,25	94	2,1	3,8	1,23	2
12,37	1,13	87	3,5	4,45	1,22	2
12,17	1,45	104	1,89	2,95	1,45	2
12,37	1,21	98	2,42	4,6	1,19	2
13,11	1,01	78	2,98	5,3	1,12	2
12,37	1,17	78	2,11	4,68	1,12	2
13,34	0,94	110	2,53	3,17	1,02	2
12,21	1,19	151	1,85	2,85	1,28	2
12,29	1,61	103	1,1	3,05	0,906	2
13,86	1,51	86	2,95	3,38	1,36	2
13,49	1,66	87	1,88	3,74	0,98	2
12,99	1,67	139	3,3	3,35	1,31	2
11,96	1,09	101	3,38	3,21	0,99	2
11,66	1,88	97	1,61	3,8	1,23	2
13,03	0,9	86	1,95	4,6	1,19	2
11,84	2,89	112	1,72	2,65	0,96	2
12,33	0,99	136	1,9	3,4	1,06	2
12,7	3,87	101	2,83	2,57	1,19	2
12	0,92	86	2,42	2,5	1,38	2
12,72	1,81	86	2,2	3,9	1,16	2
12,08	1,13	78	2	2,2	1,31	2
13,05	3,86	85	1,65	4,8	0,84	2
11,84	0,89	94	2,2	3,05	0,79	2
12,67	0,98	99	2,2	2,62	1,23	2
12,16	1,61	90	1,78	2,45	1,33	2
11,65	1,67	88	1,92	2,6	1,36	2
11,64	2,06	84	1,95	2,8	1	2
12,08	1,33	70	2,2	1,74	1,07	2
12,08	1,83	81	1,6	2,4	1,08	2
12	1,51	86	1,45	3,6	1,05	2
12,69	1,53	80	1,38	3,05	0,96	2
12,29	2,83	88	2,45	2,15	1,15	2
11,62	1,99	98	3,02	3,25	1,16	2
12,47	1,52	162	2,5	2,6	1,16	2
11,81	2,12	134	1,6	2,5	0,95	2
12,29	1,41	85	2,55	2,9	1,23	2
12,37	1,07	88	3,52	4,5	1,04	2
12,29	3,17	88	2,85	2,3	1,42	2
12,08	2,08	97	2,23	3,3	1,27	2

12,6	1,34	88	1,45	2,45	1,04	2
12,34	2,45	98	2,56	2,8	0,8	2
11,82	1,72	86	2,5	2,06	0,94	2
12,51	1,73	85	2,2	2,94	1,04	2
12,42	2,55	90	1,68	2,7	0,86	2
12,25	1,73	80	1,65	3,4	1	2
12,72	1,75	84	1,38	3,3	0,88	2
12,22	1,29	92	2,36	2,7	0,86	2
11,61	1,35	94	2,74	2,65	0,96	2
11,46	3,74	107	3,18	2,9	0,75	2
12,52	2,43	88	2,55	2	0,9	2
11,76	2,68	103	1,75	3,8	1,23	2
11,41	0,74	88	2,48	3,08	1,1	2
12,08	1,39	84	2,56	2,9	0,93	2
11,03	1,51	85	2,46	1,9	1,71	2
11,82	1,47	86	1,98	1,95	0,95	2
12,42	1,61	108	2	2,06	1,06	2
12,77	3,43	80	1,63	3,4	0,7	2
12	3,43	87	2	1,28	0,93	2
11,45	2,4	96	2,9	3,25	0,8	2
11,56	2,05	119	3,18	6	0,93	2
12,42	4,43	102	2,2	2,08	0,92	2
13,05	5,8	86	2,62	2,6	0,73	2
11,87	4,31	82	2,86	2,8	0,75	2
12,07	2,16	85	2,6	2,76	0,86	2
12,43	1,53	86	2,74	3,94	0,69	2
11,79	2,13	92	2,13	3	0,97	2
12,37	1,63	88	2,22	2,12	0,89	2
12,04	4,3	80	2,1	2,6	0,79	2
12,86	1,35	122	1,51	4,1	0,76	3
12,88	2,99	104	1,3	5,4	0,74	3
12,81	2,31	98	1,15	5,7	0,66	3
12,7	3,55	106	1,7	5	0,78	3
12,51	1,24	85	2	5,45	0,75	3
12,6	2,46	94	1,62	7,1	0,73	3
12,25	4,72	89	1,38	3,85	0,75	3
12,53	5,51	96	1,79	5	0,82	3
13,49	3,59	88	1,62	5,7	0,81	3
12,84	2,96	101	2,32	4,92	0,89	3
12,93	2,81	96	1,54	4,6	0,77	3
13,36	2,56	89	1,4	5,6	0,7	3
13,52	3,17	97	1,55	4,35	0,89	3
13,62	4,95	92	2	4,4	0,91	3
12,25	3,88	112	1,38	8,21	0,65	3
13,16	3,57	102	1,5	4	0,6	3
13,88	5,04	80	0,98	4,9	0,58	3
12,87	4,61	86	1,7	7,65	0,54	3
13,32	3,24	92	1,93	8,42	0,55	3
13,08	3,9	113	1,41	9,4	0,57	3
13,5	3,12	123	1,4	8,6	0,59	3
12,79	2,67	112	1,48	10,8	0,48	3
13,11	1,9	116	2,2	7,1	0,61	3
13,23	3,3	98	1,8	10,52	0,56	3

12,58	1,29	103	1,48	7,6	0,58	3
13,17	5,19	93	1,74	7,9	0,6	3
13,84	4,12	89	1,8	9,01	0,57	3
12,45	3,03	97	1,9	7,5	0,67	3
14,34	1,68	98	2,8	13	0,57	3
13,48	1,67	89	2,6	11,75	0,57	3
12,36	3,83	88	2,3	7,65	0,56	3
13,69	3,26	107	1,83	5,88	0,96	3
12,85	3,27	106	1,65	5,58	0,87	3
12,96	3,45	106	1,39	5,28	0,68	3
13,78	2,76	90	1,35	9,58	0,7	3
13,73	4,36	88	1,28	6,62	0,78	3
13,45	3,7	111	1,7	10,68	0,85	3
12,82	3,37	88	1,48	10,26	0,72	3
13,58	2,58	105	1,55	8,66	0,74	3
13,4	4,6	112	1,98	8,5	0,67	3
12,2	3,03	96	1,25	5,5	0,66	3
12,77	2,39	86	1,39	9,9	0,57	3
14,16	2,51	91	1,68	9,7	0,62	3
13,71	5,65	95	1,68	7,7	0,64	3
13,4	3,91	102	1,8	7,3	0,7	3
13,27	4,28	120	1,59	10,2	0,59	3
13,17	2,59	120	1,65	9,3	0,6	3
14,13	4,1	96	2,05	9,2	0,61	3

2 priedas: mokslinē publikācija.