

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ INŽINERIJOS STUDIJŲ PROGRAMA

KĘSTUTIS ŠIDLAUSKAS

VIRTUALAUS OBJEKTO VALDYMO SISTEMOS SMEGENŲ  
KOMPIUTERIO SĄSAJOS TYRIMAS

Magistro baigiamasis darbas

Darbo vadovas  
prof. dr. R. Damaševičius

KAUNAS, 2013

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ STUDIJŲ PROGRAMA

KĘSTUTIS ŠIDLAUSKAS

VIRTUALAUS OBJEKTO VALDYMO SISTEMOS SMEGENŲ  
KOMPIUTERIO SAŠAJOS TYRIMAS

Magistro baigiamasis darbas

Darbo vadovas  
prof. dr. R. Damaševičius  
2013-05-

Recenzentas  
lekt. dr. M. Patašius  
2013-05-

Atliko:  
IFM–1/2 gr. studentas  
Kęstutis Šidlauskas  
2013-05-

KAUNAS, 2013

## AUTENTIŠKUMO PATVIRTINIMAS

Patvirtinu, kad įteikiamas bakalauro baigiamasis darbas yra:

1. Atliktas mano paties ir nėra pateiktas kitam kursui šiame ar ankstesniuose semestruose;
2. Nebuvo naudotas kitame Institute/Universitete Lietuvoje ir užsienyje;
3. Nenaudoja šaltinių, kurie nėra nurodyti darbe, ir pateikia visą panaudotos literatūros sąrašą.

Kęstutis, Šidlauskas

(parašas)

Data

## **SANTRAUKA**

Šiame darbe nagrinėjama smegenų – kompiuterio sąsajos (BCI) sistema. Taip pat dirbtinių neuroninių tinklų ir atsitiktinių miškų klasifikavimo algoritmų panaudojimas smegenų – kompiuterio sąsajos sistemose. Realizuotas smegenų – kompiuterio sąsajos prototipas. Šis prototipas leidžia valdyti kompiuterio pelę, naudojant elektroencefalogramos arba elektromiogramos skaitytuvą.

Atliktas kompiuterio pelės valdymo, naudojant smegenų – kompiuterio sistemą, tyrimas, vykdant praktines užduotis. Rezultatai palyginti su įprastu būdu valdoma kompiuterio pele. Tyrime naudotas OCZ NIA elektroencefalogramos ir elektromiogramos signalų skaitytuvas.

Palyginta kuris iš naudotų klasifikavimo algoritmų pasiekia didžiausią tikslumą.

Padarytos išvados apie smegenų – kompiuterio sąsajos sistemos prototipo privalumus ir trūkumus.

## **SUMMARY**

This work analyzes the brain – computer interface (BCI) system. Also artificial neural networks and random forest classification algorithms are used in brain – computer interface systems. A prototype of the brain – computer interface was developed. The prototype lets you control your mouse using electromyogram or electroencephalogram reader.

In this work, the practical tasks carried out mouse control study using a brain – computer interface. The results were compared with the normal – controlled computer mouse. The study used OCZ NIA electroencephalogram and electromyogram signal reader.

Compared which of the used algorithms achieves the highest accuracy.

The conclusions were drawn about the BCI prototype.

## TURINYS

Lentelių sąrašas .....	8
Paveikslų sąrašas .....	9
Terminų ir santrumpų žodynas .....	10
Įvadas .....	11
1. literatūros apžvalga ir technologinių sprendimų analizė.....	12
1.1. Egzistuojantys metodai .....	12
1.1.1. Klasifikavimo metodų apžvalga .....	12
1.2. Neuroninių tinklų analizė.....	13
1.2.1. Neuroniniai tinklai .....	13
1.2.2. Neuroninių tinklų apmokymas.....	19
1.2.3. Realizacijos .....	24
1.3. Atsitiktiniai miškai (random forests) .....	24
1.4. Elektroencefalograma .....	24
1.5. Įgyvendinimo problemos .....	25
1.5.1. Architektūros pasirinkimas .....	25
1.5.2. Įgyvendinimo problemos .....	25
1.5.3. Suderinamumo problemos .....	26
1.5.4. Skirtingų įrenginių valdymas.....	26
1.6. Pirmo skyriaus išvados. ....	26
2. Smegenų – kompiuterio sąsajos sistemos reikalavimų specifikacija ir projektas, formalus aprašas .....	27
2.1. Reikalavimų specifikacija.....	27
2.1.1. Sistemos tikslai .....	27
2.1.2. Funkciniai reikalavimai .....	27
2.1.3. Nefunkciniai reikalavimai.....	27
2.2. Architektūros tikslai ir apribojimai.....	28
2.2.1. Tikslai .....	28
2.2.2. Apribojimai .....	28
2.2.3. COTS produktų panaudojimas.....	28
2.2.4. Projektavimo įrankiai.....	28
2.2.5. Įgyvendinimo strategija .....	28
2.3. Projektas, formalus aprašas.....	29
2.3.1. Sistemos statinis vaizdas.....	29
2.3.2. Būsenų diagramos.....	30
2.4. Antro skyriaus išvados.....	34

3. smegenų – kompiuterio sąsajos sistemos tyrimas .....	35
3.1. Eksperimento aprašymas .....	35
3.2. Naudojamas elektroencefalogramos skaitytuvas .....	37
3.3. Rezultatai .....	38
3.3.1. Užduočių įvykdymo rezultatai naudojant pelę .....	38
3.3.2. Klasifikavimo tikslumas su penkiomis komandomis. ....	38
3.3.3. Klasifikavimo tikslumas su dviem komandom, naudojant mintis .....	39
3.3.4. Klasifikavimo tikslumas su dviem komandom, naudojant kaktos raumenis.....	39
3.4. Trečio skyriaus išvados.....	42
4. Rezultatų apibendrinimas ir išvados .....	43
5. Literatūra.....	44

## LENTELIŲ SĄRAŠAS

1.1 lentelė. Klasifikavimo metodų analizės rezultatai .....	13
1.2 lentelė. Neuroninių tinklų palyginimas .....	19
1.3 lentelė. Apmokymo algoritmų palyginimas .....	24
3.1 lentelė. Užduočių įvykdymo laikai, naudojant pelę .....	38
3.2 lentelė. Klasifikavimo tikslumas, apmokant penkias valdymo komandas.....	38
3.3 lentelė. Klasifikavimo tikslumas, apmokant dvi valdymo komandas, naudojant mintis.....	39
3.4 lentelė. Klasifikavimo tikslumas, apmokant dvi komandas, naudojant akių mirksnius .....	39
3.5 lentelė. Užduočių įvykdymo laikai, valdant akių mirksniais .....	39
3.6 lentelė. Užduočių vykdymo laikų skirtumai, kartais.....	40
3.7 lentelė. Pasiektas informacijos perdavimo greitis, bit/min. ....	41



## PAVEIKSLŲ SĄRAŠAS

1.1 pav. Neuronų modelis .....	14
1.2 pav. Vienasluoksnis perceptronas .....	14
1.3 pav. Tiesioginio sklaidimo neuroninis tinklas .....	15
1.4 pav. Rekurentinis neuroninis tinklas .....	16
1.5 pav. Neuroninės-neraiškiosios sistemos struktūra .....	17
1.6 pav. a) Slenkstinė, b) tiesinė, c) Loginio sigmoido, d) Hiperbolinio tangento aktyvavimo funkcija.....	17
1.7 pav. Tikimybinis neuroninis tinklas (PNN) .....	18
1.8 pav. Apmokymo su mokytoju blokinė schema.....	20
1.9 pav. Mokymas su kritiku.....	21
1.10 pav. Neuronų išsidėstymas. Kairėje prieš apmokymą, dešinėje – po 500 iteracijų.....	23
1.11 pav. Bangos stebimos esant įvairioms būsenoms: a) susijaudinęs; b) atsipalaidavęs; c) apsnūdęs; d) miega; e) gilus miegas .....	25
2.1 pav. Sistemos statinis vaizdas .....	29
2.2 pav. Duomenų nuskaitymo būsenų diagrama .....	30
2.3 pav. Iš EEG skaitytuvo gaunamo signalo peržiūrėjimo būsenų diagrama.....	31
2.4 pav. Įrenginio valdymo būsenų diagrama .....	32
2.5 pav. Klasifikatoriaus apmokymo būsenų diagrama .....	33
3.1 pav. Kalkuliatorius .....	35
3.2 pav. Virtuali klaviatūra .....	35
3.3 pav. Paveikslėlis spalvojimui .....	36
3.4 pav. „Memory“ žaidimas .....	37
3.5 pav. Skaitytuvo jutiklių išdėstymas .....	37
3.6 pav. Užduočių įvykdymo laikai, naudojant pelę.....	38
3.7 pav. Užduočių įvykdymo laikai, valdant akių mirksniais.....	40
3.8 pav. Užduočių vykdomo skirtumai .....	41

## TERMINŲ IR SANTRUMPŲ ŽODYNAS

BCI – smegenų ir kompiuterio sąsaja (angl. *Brain Computer Interface*)

EEG – elektroencefalografija (angl. *Electroencephalography*)

NN – neuroninis tinklas, tarpusavyje sujungtų dirbtinių neuronų grupė

OS – operacinė sistema

JRE – java vykdymo aplinka (angl. *Java Runtime Environment*)

JAVA – objektiškai orientuota programavimo kalba, 1991 metais sukurta Sun Microsystems

Elektroencefalograma - elektromagnetinės prigimties bangos, kurios susidaro smegenyse

Elektromiograma – raumenų generuojami elektromagnetiniai signalai

Perceptronas – tiesioginio sklidimo neuroninis tinklas

Smegenų – kompiuterio sąsaja – sąsaja, leidžianti žmogui valdyti įrenginį mintimis

## ĮVADAS

Pastaruoju metu yra didelis susidomėjimas smegenų – kompiuterio sąsaja. Yra įgyvendinta daug projektų, tačiau dauguma jų yra akademinio lygmens ir orientuoti į tyrimus. Kol kas, su šiaudieninėmis technologijomis ir informacijos apdorojimo metodais, betarpiškas bendravimas su kompiuteriu naudojant mintis yra neįmanomas. Tačiau vykdomi tyrimai duomenų analizės ir apdorojimo srityje [1] jau dabar leidžia smegenų bangų, nuskaitomų naudojant elektroencefalogramas (EEG) skaitytuvus, pagalba valdyti nesudėtingus įrenginius arba programas.

Smegenų – kompiuterio sąsaja (BCI) tai valdymo metodas, kuriam nereikia, kad žmogus atliktų kokį nors fizinį veiksmą. BCI sistemos naudoja elektroencefalogramas (EEG) duomenis, gautus iš elektrodų kurie uždėti žmogui ant galvos ir įrašinėja neuronų elektrinį aktyvumą smegenyse [2]. Gauti EEG duomenys apdorojami įvairiais filtravimo metodais ir klasifikuojami, kad išsikirti valdymo komandą.

Smegenų – kompiuterio sąsajos pagrindinė taikymo sritis yra žmonės, turintys fizinę negalią. Žmonės, dėl fizinių sutrikimų negalintys bendrauti su kitais, galėtų tai daryti mintimis, padedami kompiuterio. Negalintys judėti žmonės, galėtų mintimis valdyti invalido vežimėlį, netekę galūnės – protezą.

### **Darbo tikslas ir uždaviniai**

Šiame darbe atliekamas smegenų – kompiuterio sąsajos sistemos tyrimas, signalų klasifikavimui naudojant daugiasluoksnio perceptrono, balsuojančio perceptrono ir atsitiktinių miškų klasifikavimo algoritmus. Signalų nuskaitymui naudojant „OCZ NIA“ elektroencefalogramos ir elektromiogramos skaitytuvą. Darbe aprašomas autoriaus (bendraautorius: M. Vasiljevas) sukurta smegenų – kompiuterio sąsajos sistema leidžia valdyti virtualius ir realius įrenginius, naudojant elektroencefalogramą arba elektromiogramą.

Darbo uždaviniai:

- Išanalizuoti smegenų – kompiuterio sąsajos problematiką.
- Suprojektuoti smegenų – kompiuterio sąsajos sistemos prototipą.
- Ištirti OCZ NIA įrenginiu nuskaitytų signalų klasifikavimo tikslumą, naudojant daugiasluoksnio perceptrono, balsuojančio perceptrono ir atsitiktinių miškų algoritmus.
- Ištirti smegenų – kompiuterio sąsają atliekant realaus gyvenimo užduotis.

# 1. LITERATŪROS APŽVALGA IR TECHNOLOGINIŲ SPRENDIMŲ ANALIZĖ

## 1.1. Egzistuojantys metodai

Smegenų – kompiuterio sąsajos vienas pagrindinių bruožų – srities sudėtingumas. Žmogaus smegenys dar yra mažai ištirtos, kai kurie jose vykstantys procesai kol kas išviso nesuprantami. Smegenyse apdorojami analoginiai signalai, kurie yra perduodami impulsais tarp neuronų. Kompiuteryje apdorojama skaitmeninė informacija. Bandydami apjungti šias sritis į vieną sistemą, susiduriame su daugybe problemų:

- Smegenų bangų nuskaitymas;
- Naudingos informacijos išskyrimas;
- Duomenų persiuntimas į kompiuterį;
- Duomenų apdorojimas, valdymo signalų išskyrimas.

Tačiau šiuolaikinių technologijų pagalba galima išspręsti šias problemas. Smegenų bangoms nuskaityti yra sukurti skaitytuvai, kurie gali iš dalies nufiltruoti ir kai kuriuos triukšmus. Tačiau duomenų apdorojimas ir valdymo signalų išskyrimas dar nėra trivalios užduotys. Joms spręsti yra sukurta daugybė metodų, kurie arba nepakankamai tikslūs, arba tik tam tikrom sąlygom pakankamai tikslūs arba naudoja daug resursų.

### 1.1.1. Klasifikavimo metodų apžvalga

Prieš parenkant klasifikavimo metodus, reikia apžvelgti kuo daugiau šiuolaikinių metodų. Metodų apžvalgoje naudoti šie kriterijai:

- Mokslinių straipsnių kiekis „ScienceDirect“ (adresas: <http://www.sciencedirect.com/>). Rezultatų lentelėje „I“ stulpelis;
- Mokslinių straipsnių kiekis „IEEE Xplore Digital library“ (adresas: <http://ieeexplore.ieee.org/>). Rezultatų lentelėje „II“ stulpelis;
- Mokslinių straipsnių kiekis „ACM Digital library“ (adresas: <http://dl.acm.org/>). Rezultatų lentelėje „III“ stulpelis;
- Mokslinių straipsnių kiekių aritmetinis vidurkis. Rezultatų lentelėje „IV“ stulpelis;
- Metodų rezultatų tikslumas (%), remiantis BBCI (Berlin Brain-Computer Interface) rengtų konkursų rezultatais.

Pagal parinktus kriterijus atliktos klasifikavimo metodų analizės rezultatai pateikti 1.1 lentelėje:

## 1.1 lentelė. Klasifikavimo metodų analizės rezultatai

Metodo pavadinimas	Straipsnių kiekis				Tikslumas	Pastaba
	I	II	III	IV		
Discriminative biorthogonal bases	0	0	0	0	59%	
Tikhonov regularization	3	1	2	2	58%	
Mahalanobis distance	22	6	21	16	86%	
Logistic regression	169	3	32	68	87%	
Restricted Boltzmann Machines	0	0	2	1	-	
Hidden Markov model	57	9	23	30	84%	
Bayesian classifier	26	7	15	16	67%	Kombinuotas kartu su SVM
The Filter Bank Common Spatial Pattern (FBCSP)	1	7	0	3	-	
Neural Networks arba Neuro-Fuzzy Systems	451	129	332	304	84%	Kombinuotu CSSD (Common spatial subspace decomposition) ir FD (Fisher discrimination) metodu išgaunamos 3 savybės ( $f_i = w_i T x_i + b_i$ , $i=1,2,3$ ) iš vieno EEG pavyzdžio, ir naudoja perceptronų neuroninį tinklą
Quadratic classifier	8	1	5	5	83%	
Fisher's linear discriminant	33	5	21	20	82.2%	Apskaičiuoja furje koeficientus 0-7 kiekvienam kanalui (0-2 Hz).
Linear discriminant analysis (LDA)	149	102	96	116	86%	
Independent component analysis (ICA)	174	72	109	118	-	
Principal component analysis (PCA)	178	55	70	101	-	
Support vector machine (SVM)	223	129	114	155	91%	

## 1.2. Neuroninių tinklų analizė

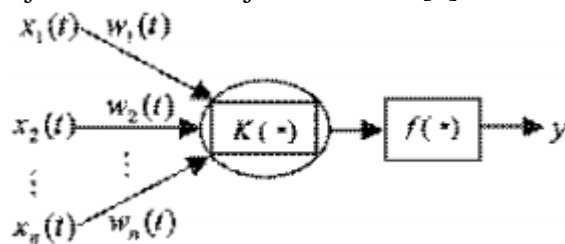
### 1.2.1. Neuroniniai tinklai

#### 1.2.1.1. Neuronų modelis

Neuronas yra informacijos apdorojimo vienetas, kuris yra neuroninio tinklo veikimo pagrindas (pav. 1.1). Neuronų veikimą galima apibūdinti:

1. Neuronas gauna keletą įėjimo reikšmių (tai neuroninio tinklo įėjimo reikšmės arba kitų neuronų išėjimo reikšmės). Kiekviena įėjimo jungtis turi savo perdavimo koeficientą (svorį). Neuronų sužadimo reikšmė formuojama skaičiuojant svorinę įėjimo signalų sumą.

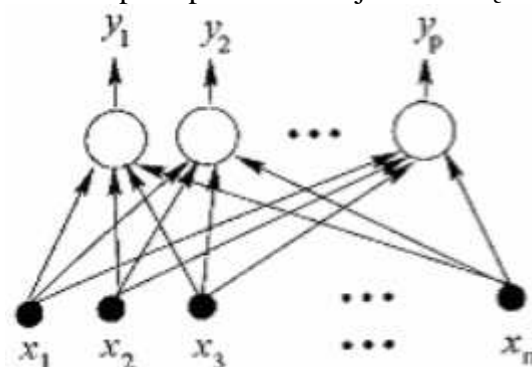
2. Pagal sužadavimo signalą, naudojant aktyvacijos funkciją (neuro no perdavimo funkciją), skaičiuojama neuro no išėjimo reikšmė [4].



1.1 pav. Neuro no modelis [5].

### 1.2.1.2. Perceptronas

Paprasčiausios architektūros yra taip vadinami *tiesioginio sklidimo* neuroniniai tinklai, kuriuose galimos tik vienkryptės į priekį jungtys. Paprasčiausias tokio tipo neuroninis tinklas – *perceptronas*, susidedantis iš vieno sluoksnio  $p$  neuronų, sujungtų su  $n$  įėjimais (pav. 1.2). Neuronų skaičius  $p$  lygus išėjimų skaičiui. Kai kurie autoriai perceptrone naudoja tik vieną neuroną, tada  $p = 1$ .



1.2 pav. Vienasluoksnis perceptronas [3].

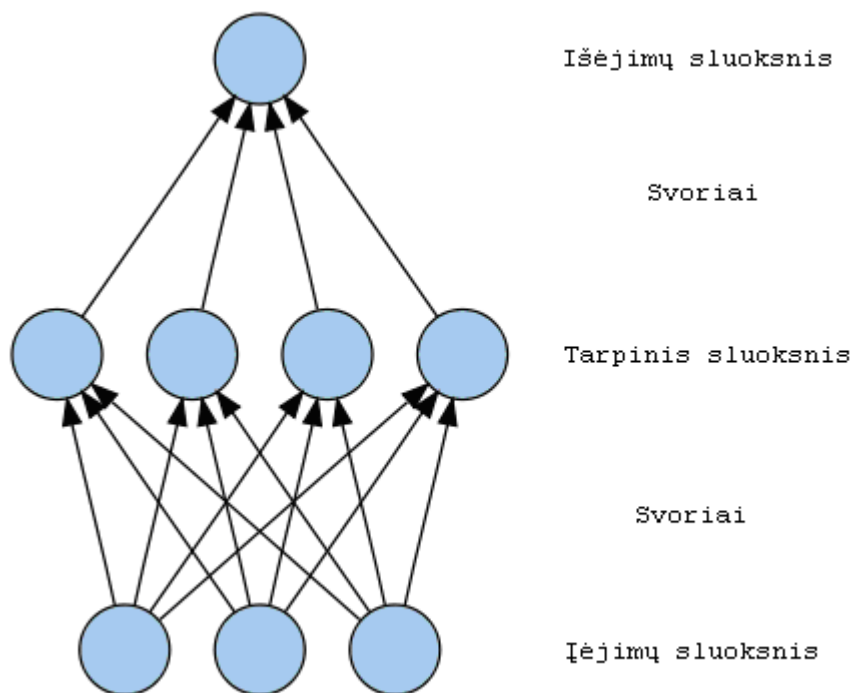
Perceptrone kiekvienas išėjimas  $y_i$  yra įėjimo vektoriaus  $X = (x_1, x_2, \dots, x_n)$  funkcija, kuri skaičiuojama pagal (1.1) formulę.

$$y_i = f(a_i) = f\left(\sum_{k=0}^n w_{ik} x_k\right), i = 1, \dots, p \quad (1.1)$$

čia  $w_{ik}$  jungties iš  $k$ -tosios įėjimo vektoriaus komponentės į  $i$ -tajį išėjimą svoris.

### 1.2.1.3. Daugiasluoksnis tiesioginio sklidimo neuroninis tinklas

Tinklai, turintys daugiau nei vieną neuronų sluoksnį, kuriuose galimi tik ryšiai į priekį iš įėjimų į išėjimus, yra vadinami *daugiasluoksniais perceptronais* [6] arba *daugiasluoksniais tiesioginio sklidimo neuroniniais tinklais*. Kiekvienas toks tinklas sudarytas iš įėjimų aibės, išėjimų neuronų sluoksnio ir paslėptų neuronų sluoksnių tarp įėjimų ir išėjimų (pav. 1.3).



1.3 pav. Tiesioginio sklidimo neuroninis tinklas.

#### 1.2.1.4. Balsuojantis perceptronas

Klasikinio perceptrono apmokymas prasideda nuo nulinio prognozavimo vektoriaus:  $v = 0$ . Jis prognozuoja, kokiai klasei priklauso duotas duomenų pavyzdys  $x$ :  $\hat{y} = \text{sign}(v \cdot x)$ . Jei prognozė skiriasi nuo  $y$  klasės, tai atnaujinamas prognozavimo vektorius:  $v = v + y_i$ . Jei prognozė yra teisinga, tai vektorius nekeičiamas. Tuomet procesas kartojamas su kitais pavyzdžiais. Dažniausiai perceptronas apmokomas kartojant šį procesą tol, kol bus rastas prognozavimo vektorius, su kuriuo visi apmokymui pateikti pavyzdžiai bus teisingai suklasifikuoti.

Balsuojančiame perceptrone [7] apmokant tinklą, kiekvieną kartą, kai padaroma klaidinga prognozė, generuojamas naujas prognozavimo vektorius ir skaičiuojama kiek kiekvienas sugeneruotas vektorius pateikė teisingų prognozių iki pirmos klaidos. Šis teisingų prognozių kiekis naudojamas kaip vektoriaus svoris balsuojant. Klasifikuojant kiekvienas vektorius balsuoja ir kuri klasė gauna daugiausiai balsų, tai klasei ir priskiriami duomenys. Balsavimo perceptrono išėjimas (1.2) formulė.

$$y_i = \text{sgn} \left\{ \sum_{n=0}^N c_n \text{sgn}(w_n x_{i,n}) \right\} \quad (1.2)$$

Čia:  $x_{i,n}$  – įėjimai,  $w_n$  – įėjimų svoriai,  $c_n$  – balsavimo svoris,  $d_i$  – pageidaujamas išėjimas,  $e_i$  – paklaida.

#### 1.2.1.5. Radialinės bazinės funkcijos (RBF) tinklas

Tai neuroninis tinklas, sudarytas iš trijų (įėjimų, išėjimų ir 1 paslėpto) sluoksnių, kuriame radialinė bazinės funkcija naudojama kaip neurono aktyvacijos funkcija.

Radialinės bazinės funkcijos užrašomos taip:  $\phi(x) = \phi(\|x - c\|)$ .

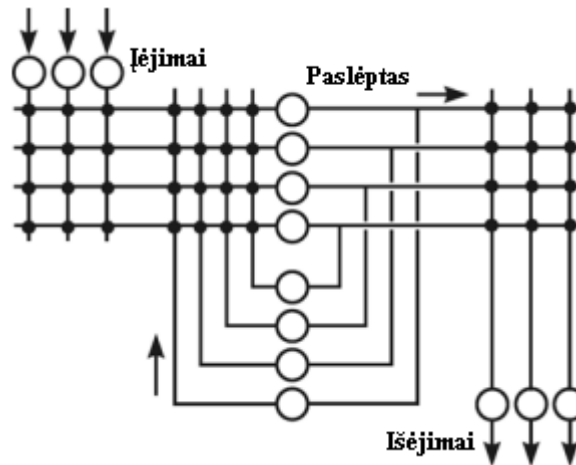
RBFN tinklai – tai dvisluoksnės struktūros tinklai, kur pirmas sluoksnis yra netiesinis atvaizdas, realizuojamas neuronais su radialinėmis bazinėmis funkcijomis, o antras sluoksnis – tiesinis – išėjimo.

RBFN tinkluose paslėptas sluoksnis realizuoja funkcijas, kurios radikaliai keičiasi aplink pasirinktą centrą ir nulines reikšmes įgyja tik prie šio centro. RBFN tinklų efektyvumas dažniausiai priklauso nuo centro parinkimo. Centrai parenkami skirtingai, kartais jie parenkami lygūs įėjimo

taškams, arba nustatomi programos apmokymo metu. Šiame sluoksnyje esančiuose neuronuose nenaudojama įėjimų svorių mechanizmas kaip daugiasluoksniame perceptrone [8].

#### 1.2.1.6. Rekurentiniai neuroniniai tinklai (RNN)

Tai neuroninis tinklas kuriame informacija sklinda dvejomis kryptimis, t.y. neuronai taip pat turi jungtis iš vėlesnių į ankstesnius neuronus (pav. 1.4). Toks tinklas gali būti nestabilus ir turi be galo sudėtingą dinamiką, dėl to realių uždavinių sprendimui dažniausiai naudojami tiesioginio sklidimo neuroniniai tinklai.



1.4 pav. Rekurentinis neuroninis tinklas [9].

#### 1.2.1.7. Neuroninė – neraiškioji architektūra (Neuro – Fuzzy)

Neuro-fuzzy tinklai sujungia taisyklėmis paremtos Fuzzy logikos semantinę aiškumą su neuroninio tinklo mokymo galimybe. Neuro-Fuzzy tinklas susideda iš trijų sluoksnių: įėjimų sluoksnis, Fuzzy taisyklių sluoksnis, išėjimų sluoksnis. Fuzzy taisyklės tinkle atitinka jungčių svorius.

Neuroninė-neraiškioji sistema susideda iš komponentų, kurie sudaro tradicinę neraiškioji sistemą, išskyrus tai, kad skaičiavimai kiekviename žingsnyje vykdomi paslėptų neuroninių sluoksnių pagalba (pav. 1.5).

Sistemą sudaro trys skirtingi sluoksniai [10]:

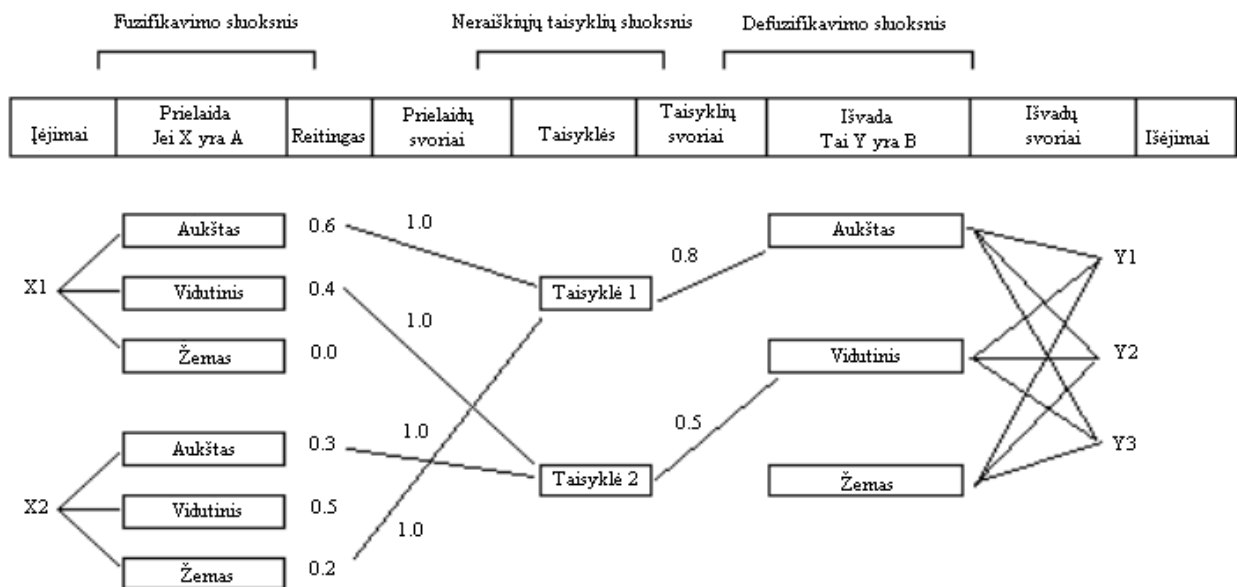
- „Fuzifikavimo“ sluoksnis.
- Neraiškių taisyklių sluoksnis.
- „Defuzifikavimo“ sluoksnis.

„Fuzifikavimo“ sluoksnyje kiekvienas neuronas reprezentuoja neraiškios aibės įėjimo duomenų priklausomybės funkciją. Šis sluoksnis atstovauja taisyklės IF dalį.

Neraiškių taisyklių sluoksnyje aktyvuojama neraiškioji taisyklė ir kiekvienos taisyklės reikšmė išėjime vaizduoja šios taisyklės pradinį svorį, kuris nustatomas apmokymo pabaigoje.

„Defuzifikavimo“ sluoksnyje kiekvienas neuronas atstovauja veiklos taisyklės išvados teiginį ir jo priklausomybės funkcija gali būti sudaryta iš vienos ar dviejų sigmoidinių funkcijų ir linijinės funkcijos. Kiekvieno junginio svoris šiuo atveju yra priklausymo funkcijos „svorio centras“ ir šis parametras yra taip pat gaunamas iš apmokymo.



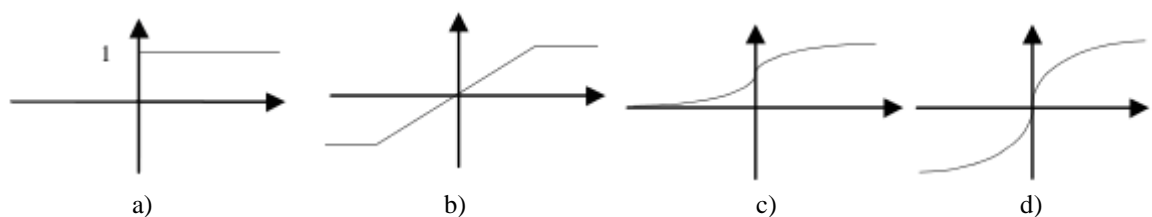


1.5 pav. Neuroninės-neraiškiosios sistemos struktūra.

### 1.2.1.8. Compositional pattern – producing tinklai (CPPN)

Tai Neuroninių tinklų tipas, kuriame kombinuojamos skirtingos neuronų aktyvavimo funkcijos [11].

Dažniausiai naudojamos neuronų aktyvavimo funkcijos:



1.6 pav. a) Slenkstinė, b) tiesinė, c) Loginio sigmoido, d) Hiperbolinio tangento aktyvavimo funkcija.

Slenkstinė (pav. 1.5.a) -  $f(v) = 1$ , kai  $v \geq 0$  ir  $f(v) = 0$ , kai  $v < 0$  [12].

Tiesinė (pav. 1.5. b) -  $f(v, L) = \begin{cases} L, v \geq L \\ v, |v| < L \\ -L, v \leq -L \end{cases}$  [13].

Loginio sigmoido (pav. 1.5. c) -  $f(v) = \frac{1}{1 + e^{-v}}$  [14].

Hiperbolinio tangento (pav. 1.5. d) -  $f(v) = \tanh(v)$  [14].

### 1.2.1.9. Tikimybiniai neuroniniai tinklai (PNN)

Tikimybiniai neuroniniai tinklai yra Bayes sprendimų strategijos įgyvendinimas [15] neuroniniuose tinkluose. Veiksmai organizuojami į daugiasluksnį tiesioginio sklidimo tinklą, susidedantį iš keturių sluoksnių (pav. 1.7):

1. Įėjimų sluoksnius, jame neatliekami jokie skaičiavimai [16];
2. Modelių sluoksnius, jame skaičiuojama funkcija (1.3)

$$p_{i,j} = \frac{1}{(2\pi\sigma_{i,j}^2)^{\frac{d}{2}}} \exp\left(-\frac{\|x - c_{i,j}\|^2}{2\sigma_{i,j}^2}\right) [15], \quad (1.3)$$

čia  $c_{i,j}$  reiškia vektorių,  $\sigma_{i,j}$  yra išlyginimo parametras;

3. Sumavimo sluoksnis, sumuodamas ir skaičiuodamas visų neuronų, priklausančių tai pačiai klasei, vidurkį skaičiuoja maksimalią tikimybę, kad modelis  $x$  bus klasifikuotas kaip  $C_i$  (1.4) formulė.

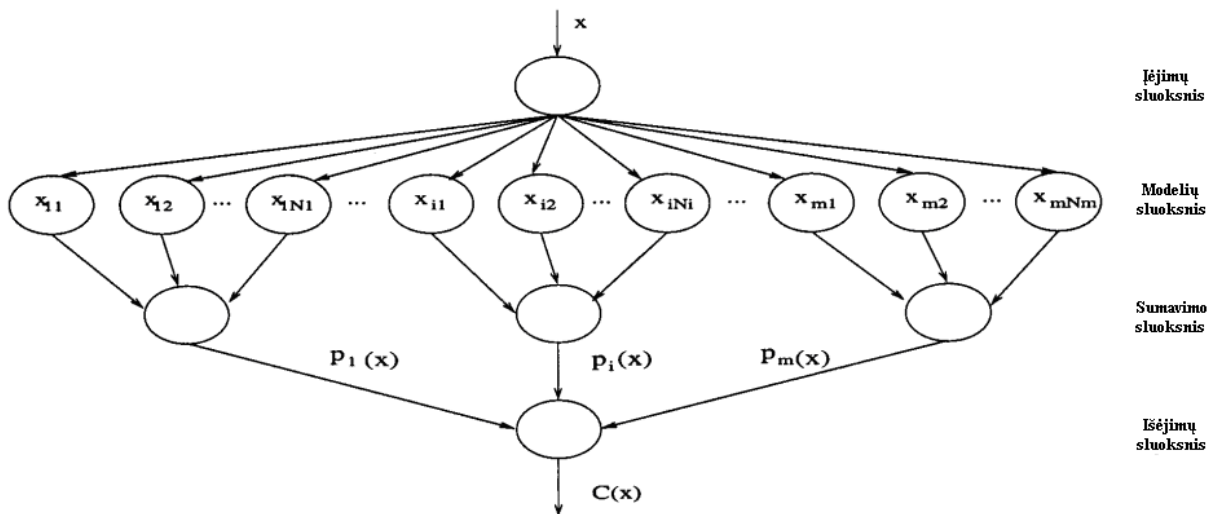
$$f_j(x) = \sum_{i=1}^{M_j} \beta_{i,j} p_{i,j}(x), \quad 1 \leq j \leq K \quad (1.4)$$

čia  $M_j$  nurodo pavyzdžių kiekį klasėje  $j$ ,  $\beta_{i,j}$  maišymo koeficientas, kuris turi tenkinti sąlyga (1.5)

$$\sum_{i=1}^{M_j} \beta_{i,j} = 1, \quad 1 \leq j \leq K \quad (1.5)$$

4. Išėjimų sluoksnyje skaičiuojama (1.6) formulė

$$C_j(x) = \frac{\alpha_j f_j(x)}{\sum_{i=1}^K \alpha_i f_i(x)} \quad (1.6)$$



1.7 pav. Tikimybinis neuroninis tinklas (PNN).

### 1.2.1.10. Neuroninių tinklų palyginimas

1.2 lentelė. Neuroninių tinklų palyginimas.

Pavadinimas	Privalumai	Trūkumai
Perceptronas	Programinės bei techninės realizacijos yra labai paprastos; Paprastas ir greitas mokymo algoritmas.	Primityvūs skirstymo paviršiai leidžia spręsti tik paprastus atpažinimo uždavinius
Daugiasluoksnis tiesioginio sklidimo neuroninis tinklas	vienas iš populiariausių ir efektyviausių DNT tipų; Leidžia išspręsti daugelį užduočių.	Kiekvienos iteracijos metu vyksta tinklo parametrų pokyčiai. Toks funkcionavimas padidina mokymo laiką.
Balsuojantis perceptronas	lengvai realizuojamas; naudoja mažai resursų; pakankamai tikslus lyginant su kitais algoritmais	Nėra pats tiksliausias algoritmas
Radialinės bazinės funkcijos (RBF) tinklas	gali modeliuoti bet kokią nelineinę funkciją naudodami vieną paslėptą sluoksnį; greitai apmokomi.	apmokymo algoritmai ne visada parenka optimalias radialinių funkcijų centrų kombinacijas
Rekurentiniai neuroniniai tinklai		gali būti nestabilus ir turi be galo sudėtingą dinamiką
Neuroninė – neraiškioji architektūra	geba atstovauti žmogaus žinioms būdingus neaiškumus naudojant lingvistinius kintamuosius; dėl natūralių taisyklių atvaizdavimo, lengvas rezultatų interpretavimas; lengvas žinių bazės praplėtimas, pridėdant naujas taisykles; atsparumas sistemos sutrikimams.	negali apibendrinti, arba gali tik atsakyti tai kas yra parašyta taisyklėse; neatsparus tinklo struktūros pokyčiams, tokie pokyčiai pakeičia taisyklių sąrašą; priklauso nuo eksperto, nustatančio logines taisykles.
Tikimybiniai neuroniniai tinklai	Greitas mokymasis; Iš prigimties lygiagreti struktūra; Garantuojamas optimalus klasifikatorius, didėjant apmokymo pavyzdžių rinkiniui (nėra lokalių minimumų problemos); Apmokymo pavyzdžiai gali būti pridėti arba išimti be plataus permokymo	Ne toks bendras kaip backpropagation; Didelis atminties reikalavimas; Lėtas vykdymas.

### 1.2.2. Neuroninių tinklų apmokymas

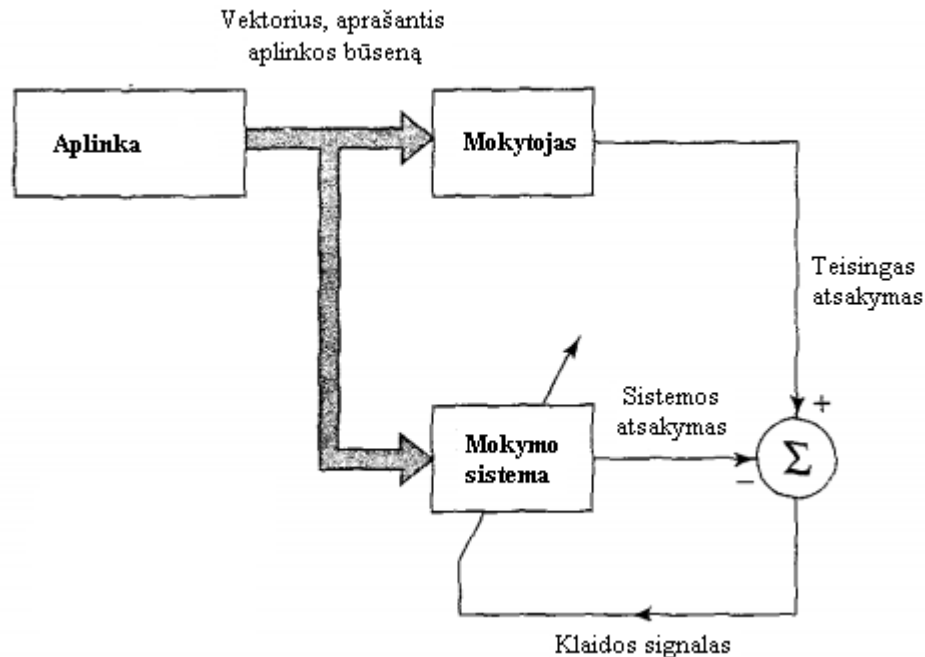
Egzistuoja du pagrindiniai neuroninių tinklų apmokymo tipai: mokymasis su mokytoju ir mokymasis be mokytojo, iš kurių labiau paplitęs mokymasis su mokytoju (egzistuoja ir trečias mokymo tipas: *reinforcement learning*).

Neuroninis tinklas apmokomas naudojant mokymo imtį. Mokyme su mokytoju, mokymo imtis sudaroma iš įėjimo reikšmių kartu su atitinkamomis išėjimo (užduoties) reikšmėmis. Jei tinklas tinkamai apmokytas, jis gali modeliuoti funkciją, siejančią įėjimo bei išėjimo kintamuosius ir gali daryti spėjimą, kai išėjimo reikšmės nežinomos.

Mokymo proceso metu dažniausiai skaičiuojamas neuroninio tinklo darbą įvertinantis parametras. Tai gali būti neuroninio tinklo daroma klaida arba kitoks neuroninį tinklą įvertinantis kriterijus. Toliau, pagal mokymo taisyklę, koreguojami tinklo svoriai ir slenksčių reikšmės taip, kad neuroninis tinklas darytų mažesnę klaidą.

### 1.2.2.1. Mokymasis su mokytoju

Apmokymo su mokytoju blokinė diagrama pavaizduota 1.8 paveiksle. Mokytojas apie aplinką žino įėjimų ir išėjimų forma. Neuroninis tinklas apie aplinką nieko nežino. Mokytojas ir neuroninis tinklas iš aplinkos gauna apmokymo pavyzdį. Mokytojas, pagal turimas žinias, nustato koks turi būti teisingas atsakymas. Neuroninis tinklas gražina savo atsakymą. Tuomet skaičiuojamas skirtumas tarp neuroninio tinklo gražinto atsakymo ir mokytojo atsakymo. Gautas skirtumas siunčiamas neuroniniam tinklui kaip klaidos signalas, tada pagal apmokymo pavyzdį ir klaidos signalą reguliuojami tinklo parametrai. Taip žingsnis po žingsnio neuroninis tinklas apmokomas emuliuoti mokytoją. Tokiu būdu mokytojo žinios perkeliama į neuroninį tinklą.



1.8 pav. Apmokymo su mokytoju blokinė schema [6].

Sistemos tikslumo matavimui galima naudoti klaidų paviršiaus funkciją. Klaidų paviršius yra vidurkinamas pagal visų galimų išėjimų – įėjimų reikšmes. Kiekvienas veiksmas, atliktas kartu su mokytoju, yra atvaizduojamas, kaip taškas, tame paviršiuje. Kad sistema tobulėtų, gautas taškas turi judėti mažiausio paviršiaus taško link. Mokymo su mokytoju sistema tai vertina pagal klaidų paviršiaus gradientą. Paviršiaus gradientas yra vektorius, kiekviename taške rodantis, greičiausią nusileidimą į žemiausią paviršiaus tašką. Jei mokoma pagal pavyzdžius, tai sistema gali iš karto įvertinti gradiento vektorių [6].

### 1.2.2.2. Mokymasis be mokytojo

Jame nenaudojamos užduoties reikšmės. Vietoje įėjimo – išėjimo atvaizdavimo neuroninis tinklas mokomas pats ieškoti koreliacijų arba panašumų tarp mokymo aibės įėjimų [6]. Čia nėra grįžtamojo ryšio, pasakančio, kuris atsakymas yra teisingas. Mokymo be mokytojo algoritmuose nėra mokytojo signalo. Čia turima tik mokymo aibė  $\{x^j, j=1, \dots, N\}$ , susidedanti iš vektorių, priklausančių erdvei  $R^n$ . Metodų tikslas yra kategorizuoti mokymo duomenis arba rasti juose kokius nors reguliarumus ar ypatumus. Mokymo be mokytojo sėkmė glūdi tame, kad parenkamas nuo mokytojo nepriklausomas kriterijus, mokymo metu tas kriterijus optimizuojamas, parenkant tinkamas svorių reikšmes.

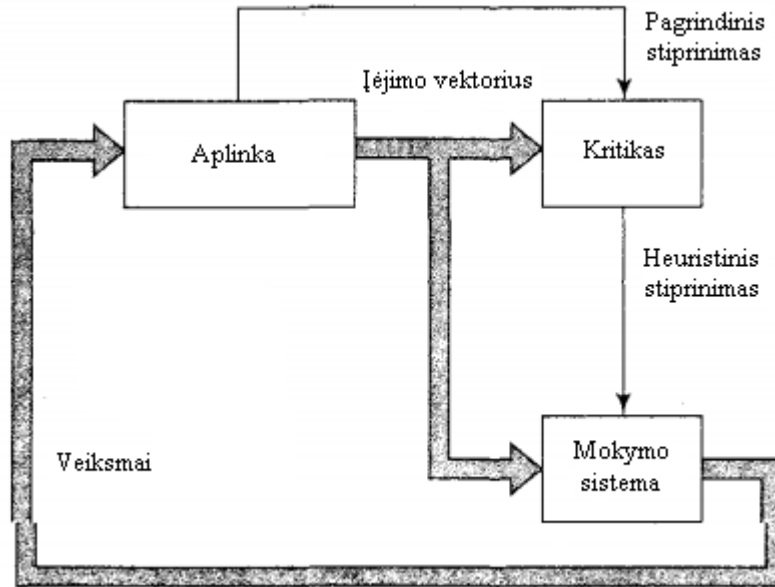
Galimos trys mokymo be mokytojo strategijos, kurios leidžia spręsti atitinkamus trijų tipų duomenų analizės uždavinius:

1. Hebb'o tipo mokymas,
2. varžytinių tipo mokymas,
3. saviorganizuojančio žemėlapių (neuroninio tinklo) mokymas.

### 1.2.2.3. Mokymasis su kritiku

Tai tarpinis tarp mokymo su mokytoju ir mokymu be mokytojo apmokymo būdas. Tai vienas iš metodų, skirtų spręsti situacijas, kai turima mažiau detalesnė informacija, tai yra, kai atgal sklaidžiama mažiau informacijos, nei mokymo su mokytoju metodo atveju. Šis mokymas panašus į mokymą su mokytoju tuo, kad taip pat gaunama grįžtama iš išorės informacija, tačiau ji įvertinanti, o ne apmokanti. Todėl mokymas vadinasi mokymu su kritiku.

Apmokant su kritiku (1.9 pav.) neuroninis tinklas bendrauja su aplinka. Kritikas iš aplinkos gauna pagrindinį stiprinimo signalą ir jį paverčia tikslesniu stiprinimo signalu [6].



1.9 pav. Mokymas su kritiku [6].

### 1.2.2.4. „Klaidos sklidimo atgal“ mokymo algoritmas

Šiame algoritme klaidos sklinda iš išėjimų į įėjimus. „Klaidos sklidimo atgal“ algoritmas skaičiuoja tinklo klaidų gradientą [17]. Algoritmą sudaro du žingsniai:

- įėjimų reikšmių „sklidimas pirmyn“ iš įėjimų į išėjimų sluoksnį, jo metu sugeneruojami išėjimų aktyvavimo reikšmės;
- paklaidos „sklidimas atgal“ iš išėjimų į įėjimų sluoksnį, jo metu sugeneruojami išėjimų ir tarpinių sluoksnių svorių pokyčiai.

Kaip perceptrono mokyme, taip ir „klaidos sklidimo atgal“ algoritme naudojama mokymo su mokytoju strategija.

Naudosime (1.9) dalinę kvadratinių sumų paklaidą  $E^j(W)$ , svoriai bus keičiami pagal (1.10) formulę.

Pirmame žingsnyje įėjimų reikšmėms  $x^j$  apskaičiuojamos išėjimų reikšmės  $y^j$ . Įvertinama paklaidos funkcija  $E^j(W)$  išėjimų sluoksnyje  $L$ . Tuo baigiama pirma fazė.

Antrame žingsnyje, jei paklaidos funkcija  $E^j(W)$  nelygi nuliui, reikia keisti svorių matricą  $\Delta W$ . Pagal (1.11) formulę keičiami visi svoriai  $w_{ik}$  jungčių, kurios jungia  $k$ -tąjį neuroną  $(l - 1)$ -ajame sluoksnyje su  $i$ -tuoju neuronu  $(l)$ -ajame sluoksnyje.

$$E^j(W) = \frac{1}{2} \sum_{i=1}^p (y_i^j - t_i^j)^2, \quad (1.9)$$

$$\Delta w_{ik}^j(m') = -\eta \frac{\partial E^j(m')}{\partial w_{ik}}, \quad (1.10)$$

$$\Delta w_{ik}^j = -\eta \frac{\partial E^j}{\partial w_{ik}}. \quad (1.11)$$

Dalinės išvestinės išreiškiamos taip:

$$\frac{\partial E^j}{\partial w_{ik}} = \frac{\partial E^j}{\partial a_i^j} \frac{\partial a_i^j}{\partial w_{ik}}, \quad (1.12)$$

$$y_i = f_i(a_i) = f_i\left(\sum_{k=0}^{n_{l-1}} w_{ik} y_k\right), \quad i=1, \dots, n_l. \quad (1.13)$$

Iš (1.13) formulės gauname:

$$\frac{\partial a_i^j}{\partial w_{ik}} = y_k^j. \quad (1.14)$$

Naudojant pažymėjimą

$$\delta_i^j = \frac{\partial E^j}{\partial a_i^j}, \quad (1.15)$$

(1.14) ir (1.15) išraiškas įstačius į (1.12) ir (1.11) gauname:

$$\frac{\partial E^j}{\partial w_{ik}} = \delta_i^j y_k^j, \quad (1.16)$$

$$\Delta w_{ik}^j = -\eta \delta_i^j y_k^j, \quad (1.17)$$

$i \in l$  - ajam sluoksniui

$k \in (l-1)$  - ajam sluoksniui

Išėjimų sluoksniui, t.y. kai neuronas  $i \in l$  - ajam sluoksniui, turime:

$$\delta_i^j = \frac{\partial E^j}{\partial a_i^j} = f'(a_i^j)(y_i^j - t_i^j) \quad i \in \text{išėjimų sluoksniui } L, \quad (1.18)$$

Lieka problema, kaip rasti  $\frac{\partial E^j}{\partial a_i^j}$  paslėptiems neuronams, t.y. kai  $i \in l$ -ajam sluoksniui, kai  $l < L$ .

$L$ . Naudojant dalines išvestines bendru atveju galima paršyti:

$$\delta_i^j = \frac{\partial E^j}{\partial a_i^j} = \sum_{s=1}^{n_{l+1}} \frac{\partial E^j}{\partial a_s^j} \frac{\partial a_s^j}{\partial a_i^j}, \quad (1.19)$$

Čia  $n_{l+1}$  žymi neuronų  $(l+1)$ -ajame sluoksnyje skaičių. Išraiška  $\frac{\partial E^j}{\partial a_i^j}$  yra lygi dydžiui  $\delta_s^j$ , apibrėžtam  $l+1$ -ajam sluoksnyje. Atsižvelgiant į (1.19) formulę gauname  $\frac{\partial a_s^j}{\partial a_i^j} = g'(a_i^j) w_{si}$ . Tada paslėptiems  $i$ -tiesiems neuronams:

$$\delta_i^j = f'(a_i^j) \sum_{s=1}^{n_{l+1}} w_{si} \delta_s^j, \quad (1.20)$$

$i \in \text{sluoksniui } l < L$

$s \in \text{sluoksniui } l+1$ .

Iš pradžių reikia apskaičiuoti  $\delta$  reikšmes išėjimų sluoksnyje  $L$  pagal (1.18) formulę. Tada palaipsniui skaičiuoti  $\delta$  reikšmes paslėptiems neuronams tarpiniuose sluoksniuose  $l < L$  naudojant  $(l+1)$ -mo sluoksniu  $\delta$  reikšmes (1.20).

Kai visi svoriai pakeičiami pagal (1.17) formulę, į tinklą pateikiamas sekantis mokymo pavyzdys  $x^j$  ir procedūra kartojama iš naujo. Apmokymas baigiamas, kai pasiekiamas nustatyta paklaidos funkcijos reikšmė, arba nustatytas iteracijų kiekis.

Norint pagreitinti mokymosi procesą, yra koreguojama *delta* taisyklė (1.17), pridendant tam tikrą konstantą:

$$\Delta w_{ik}^j(m') = -\eta \delta_i^j(m') y_k^j(m') + a \Delta w_{ik}^j(m'-1), \quad (1.21)$$

čia  $\alpha$  dažniausiai yra teigiama konstanta  $0 \leq \alpha \leq 1$ , ir yra vadinama *momentum* konstanta. Kai  $\alpha = 0$ , gaunama (1.16) formulė. *Momentum* reikšmė naudojama „klaidos sklidimo atgal“ mokymo algoritme, norint išvengti „užstrigimo“ lokaliame minimume ir pagreitinti tinklo konvergavimo procesą [3].

### 1.2.2.5. Saviorganizuojantys neuroniniai tinklai

Saviorganizuojantys neuroniniai tinklai (žemėlapiai (SOM)) dar vadinami Kohoneno neuroniniais tinklais [18]. Tai yra vienasluoksnis, konkuruojantis neuroninis tinklas, kuris surikiuoja neuronus tam tikra tvarka [19]. Konkuruojantis mokymasis yra adaptyvus procesas, kurio metu neuronai palaipsniui tampa jautrūs skirtingoms įėjimų kategorijoms, pavyzdžių rinkiniams įėjimų erdvėje [20].

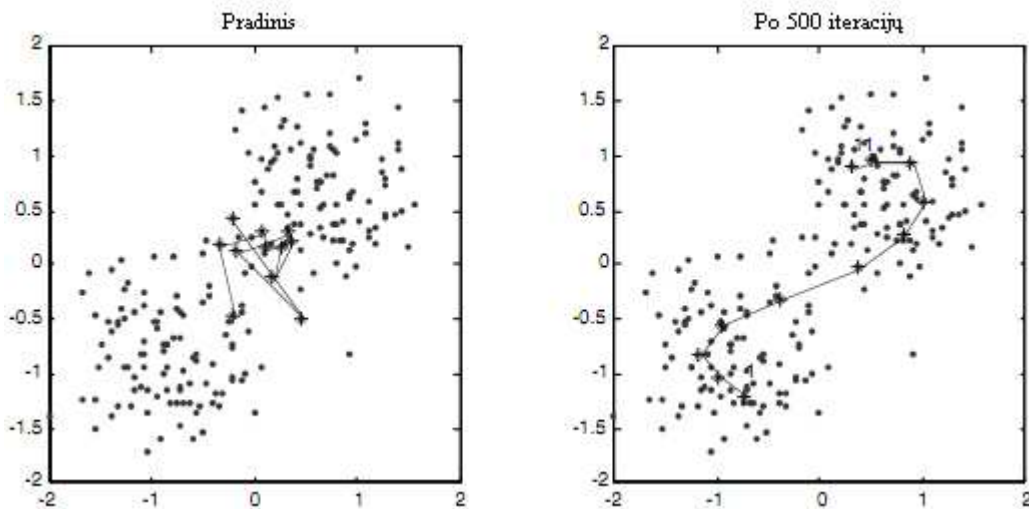
Kiekvienas neuronas turi du įėjimus, kurių svoriai atitinka koordinatės dvimatėje erdvėje. Prieš apmokant tinklą, neuronai, sprendimų aibėje, išdėstomi atsitiktine tvarka (1.10 pav.). Jei paduoti įėjimai yra arti neurono, tai jo išėjime bus 1, rodantis, kad tas neuronas yra laimėtojas atstovauti pasiūstų savybių vektoriui. Likę neuronai gražins 0.

Kiekvieną kartą atsiradus neuronui laimėtojui, laimėtojo ir greta jo esančių neuronų įėjimai koreguojami taip, kad jų įėjimų ryšiai geriau atitiktų paduoto įėjimo vektorių. Kelių neuronų įėjimai bus koreguojami, priklauso nuo  $h$  (1.22), kuris yra mažėjanti funkcija ir nurodo kokių atstumu nuo laimėjusio neurono, bus keičiami neuronų įėjimai.

$$h_{ij} = h(\|r_i - t_j\|; t) \quad (1.22)$$

Čia  $t$  – laikas,  $r_i, r_j$  – dydžių vektoriai.

Praktikoje  $h$  parenkamas didelis, kad pradėdant apmokymą būtų padengta visa sprendimų aibė, o mokymo metu jo reikšmė išlėto mažėja.



**1.10 pav.** Neuronų išsidėstymas. Kairėje prieš apmokymą, dešinėje – po 500 iteracijų [19].

Vienas iš tinklo apsimokymo kokybės įvertinimo kriterijų yra kvantavimo paklaida

$$E_{SOM} = \frac{1}{m} \sum_{j=1}^m \|x_j - m_{c(j)}\|. \quad \text{Tai vidutinis atstumas tarp kiekvieno } n\text{-mačio duomenų vektoriaus}$$

$X^j = (x_1^j, x_2^j, \dots, x_n^j)$  ir jo vektoriaus-nugalėtojo  $m_{c(j)}$ ,  $m$  – analizuojamų vektorių skaičius [3].

### 1.2.2.6. apmokymo algoritmų palyginimas

1.3 lentelė. Apmokymo algoritmų palyginimas.

Algoritmo pavadinimas	Privalumai	Trūkumai
Mokymasis su mokytoju	Visos klasės ir išėjimai yra suprantami žmonėms; Lengvai naudojamas skiriamųjų modelių klasifikavimui ir duomenų regresijai	Gali būti sunku sužymėti apmokymo duomenims klases.
Mokymasis be mokytojo	Kadangi neapibrėžtos klasės, gali atkreipti dėmesį į požymius, kurie nebuvo pastebėti	Neaiški mokymosi kryptis; Gali neaptikti skiriamų požymių duomenų rinkinyje
Mokymasis su kritiku	Grižtamasis ryšys gali būti daugiamačiai; Gerai veikia kai nėra lengva apibrėžti klases	Neaišku koks grįžtamasis ryšys geriausias
„Klaidos sklaidimo atgal“ mokymo algoritmas	Lengva naudoti, tik keli reguliuojami parametrai; Pritaikomas plačiam problemų ratui; Gali suformuoti sudėtingus netiesinius žemėlapius; Labai paplitęs apmokymo metodas	Apmokymas gali būti lėtas; Naujas apmokymas perrašys seną apmokymą, jei seni pavyzdžiai nėra naudojami; Neaptinka naujumo, todėl turi būti apmokomas su žinomais rezultatais; Lokalaus minimumo problema
Saviorganizuojantys neuroniniai tinklai	Paprastas ir lengvai suprantamas algoritmas; Topologinis grupavimas; Algoritmas be mokytojo, kuris gerai veikia su netiesiniais duomenų rinkiniais; Gerai vizualizuoja duomenis.	Reikalauja daug kompiuterio resursų

### 1.2.3. Realizacijos

Dėl Neuroninių tinklų tipų gausos egzistuoja daug bibliotekų realizuojančių šiuos metodus.

Neuroph – šiame įrankyje realizuota perceptrono, daugiasluoksnio perceptrono, neuro-fuzzy tipo neuroniniai tinklai.

Encog – šiame įrankyje realizuota paprasto ir daugiasluoksnio perceptrono, radialinės bazinės funkcijos neuroniniai tinklai.

WEKA – šioje bibliotekoje realizuota daugiasluoksnis ir balsuojantis perceptronai.

### 1.3. Atsitiktiniai miškai (random forests)

Atsitiktinis miškas sudarytas iš kelių medžio tipo klasifikatorių  $\{h(\mathbf{x}, \Theta_k), k=1, \dots\}$ , kur  $\{\Theta_k\}$  yra nepriklausomi, vienodai paskirstyti atsitiktiniai vektoriai. Kiekvienas medis balsuoja kuriai klasei priskirti įėjimo duomenis [21]. Su dauguma duomenų rinkinių, šis algoritmas pasiekia didelį tikslumą ir veikimo greitį.

### 1.4. Elektroencefalograma

Smegenų bangos yra elektromagnetinės prigimties bangos, kurios susidaro smegenyse. Yra žinoma, kad kiekvienas elektrocheminis neuronų impulsas sukuria tam tikro dažnio elektromagnetinį lauką. Jį galima išmatuoti elektroencefalografu (EEG) arba kitais prietaisais. Elektroencefalogramos bangos, pagal dažnių juostas, skirstomos į 5 bangas [23].

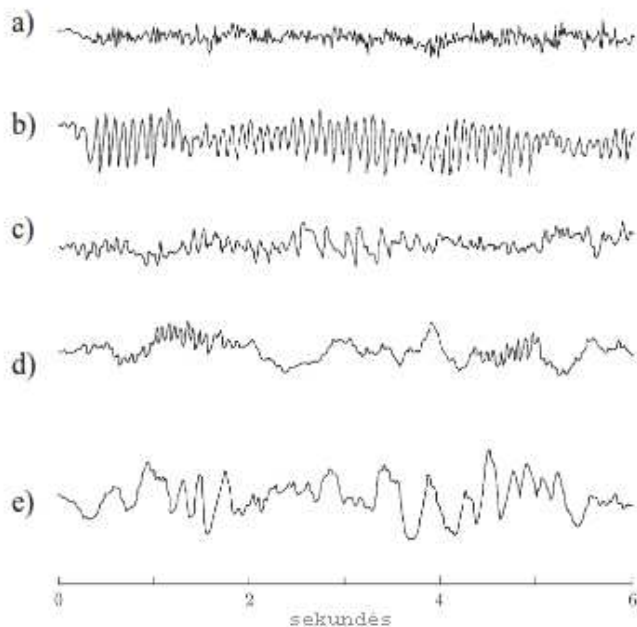
Alfa bangos (8-12 Hz)

Šios bangos atsiranda atsipalaidavus, esant užmerktoms akims bei nusiraminus. Alfa bangos atitinka budrios sąmonės, esant atsipalaidavus, būseną.

Beta bangos (12-30 Hz)



Tai mažos amplitudės bangos asocijuojamos su smegenų žievės aktyvumu.  
Delta bangos (0,5 - 4 Hz)  
Parastai būna gilaus miego fazėje ir turi didelę amplitudę.  
Gama bangos (>30 Hz)  
Gama bangos asocijuojamos su informacijos apdorojimu smegenų žievėje.  
Teta bangos (4 - 8 Hz)  
Atsiranda esant mieguistumui ir tam tikroms miego fazėms [22].



**1.11 pav.** Bangos stebimos esant įvairioms būsenoms: a) susijaudinęs; b) atsipalaidavęs; c) apsnūdęs; d) miega; e) gilus miegas.

## 1.5. Įgyvendinimo problemos

Įgyvendinant smegenų – kompiuterio sąsajos platformą, gali kilti įvairių problemų.

### 1.5.1. Architektūros pasirinkimas

**Problema:** kuriant platformą, kuri skirta produktų kūrimui komponuojant įvairius modulius – būtina architektūra su moduline struktūra.

**Sprendimas:** sistemos pagrindui naudoti modulinę struktūrą.

### 1.5.2. Įgyvendinimo problemos

#### 1.5.2.1. Skirtingi duomenų šaltiniai

**Problema:** egzistuoja daugybė įrenginių, gebančių nuskaityti smegenų bangas ir realiu laiku perduoti į kompiuterį.

**Sprendimas:** platformoje bus realizuotas standartinis nuskaitymo būdas. Realizacijai reikalingi metodai bus pateikti abstrakčioje klasėje, kurią paveldėjus bus galima realizuoti bet kokį duomenų pateikimo būdą.

#### 1.5.2.2. Skirtingi duomenų formatai

**Problema:** skirtingos realizacijos gali duomenis priimti skirtingais formatais, pvz.: vienu atveju gali būti naudojamas masyvas, kitu vektorius. Nors duomenys juose vienodi, bet formatai nesuderinami.

**Sprendimas:** bus naudojami standartiniai duomenų tipai, o kuriant konkretų modulį bus atliekama reikiamas duomenų konvertavimas.

### *1.5.2.3. Skirtingiems metodams reikalingas skirtingas parametrų kiekis*

**Problema:** skirtingiems metodams reikalingas skirtingas parametrų kiekis.

**Sprendimas:** dalis parametrų gali būti parinkti naudojant įvairius algoritmus, kurie gali būti realizuoti kartu su moduliu. Jei parametrų įvedimui reikalingas vartotojo įsikišimas – bus sukurtas parametrų surinkimo modulis, o programuotojui reikės sukurti parametrų įvedimo formą.

### **1.5.3. Suderinamumo problemos**

#### *1.5.3.1. Skirtingų platformų palaikymas*

**Problema:** programa gali būti naudojama skirtingose platformose, todėl svarbus įvairių sistemų palaikymas.

**Sprendimas:** programa realizuojama naudojant java programavimo kalbą, kuri yra nepriklausoma ir veikia skirtingose operacinėse sistemose.

#### *1.5.3.2. Metodu, realizuotų skirtingomis programavimo kalbomis palaikymas*

**Problema:** ne visų algoritmų realizacijos gali būti parašytos java programavimo kalba. O programme svarbu užtikrinti galimybę naudoti kuo daugiau algoritmų.

**Sprendimas:** kiekviena realizacija egzistuos kaip atskiras modulis, kuriame bus realizuota atitinkama sąsaja. Modulio realizacijos ribose bus galima realizuoti kreipimąsi į konkrečią realizaciją.

### **1.5.4. Skirtingų įrenginių valdymas**

**Problema:** siekiant kuo didesnio programos universalumo – būtina užtikrinti galimybę pritaikyti programą įvairiems valdomiems įrenginiams, tiek fiziniams, tiek virtualiems.

**Sprendimas:** kiekvienam valdomam įrenginiui realizuojama atskiras modulis, kuriame bus realizuotas valdymo komandų perdavimas į įrenginį.

## **1.6. Pirmo skyriaus išvados.**

Smegenų – kompiuterio sąsajos sistemoje pagrindinė užduotis - iš gaunamo signalo atpažinti kokią komandą reikia vykdyti. Tam naudojami klasifikavimo algoritmai. Kadangi signalai yra sunkiai atskiriami, o sistema turi veikti realiu laiku, reikalingi algoritmai, kurie pasiekia didelį tikslumą bet naudoja mažai resursų. Šiame skyriuje analizuoti neuroniniai tinklai, neuroninių tinklų apmokymo algoritmai, atsitiktinių miškų klasifikavimo algoritmas. Taip pat smegenų – kompiuterio sąsajos įgyvendinimo problemos. Kadangi elektroencefalogramos signalo dažniai yra žemi, gali kilti problemų dėl sistemos veikimo realiu laiku.

## **2. SMEGENŲ – KOMPIUTERIO SĄSAJOS SISTEMOS REIKALAVIMŲ SPECIFIKACIJA IR PROJEKTAS, FORMALUS APRAŠAS**

### **2.1. Reikalavimų specifikacija**

Smegenų – kompiuterio sąsaja – tai sistema, kuri smegenų generuojamas elektromagnetines bangas verčia į valdymo signalus. Pagrindinė smegenų – kompiuterio sąsajos taikymo sritis – fizinę negalią turintiems žmonėms skirtų įrenginių valdymas. Ši sistema, žmonėms turintiems fizinę negalią turėtų suteikti galimybę valdyti elektroninius įrenginius.

#### **2.1.1. Sistemos tikslai**

Smegenų – kompiuterio sąsajos programa turi atlikti įrenginio valdymą smegenų generuojamomis EEG bangomis. Taip pat sistema gali veikti kaip elektroencefalogramos signalų apdorojimo ir klasifikavimo tyrimo platforma, kuri teikia galimybę pasirinkti skirtingus signalų apdorojimo ir klasifikavimo algoritmus.

Bendru atveju smegenų – kompiuterio sąsajos sistema turi atlikti šiuos veiksmus:

- Nuskaityti EEG signalą iš skaitytuvo;
- Valdyti įrenginį;
- Apdoroti EEG signalą įvairiais signalų apdorojimo metodais;
- Klasifikuoti EEG signalą naudojant įvairius klasifikavimo algoritmus;
- Grafiškai atvaizduoti EEG signalą.

#### **2.1.2. Funkciniai reikalavimai**

- Įkelti duomenis iš failo
- Žiūrėti signalo vaizdą
- Nuskaityti duomenis
- Pasirinkti EEG skaitytuvą
- Nutraukti valdymą
- Valdo įrenginį
- Išsaugoti duomenis faile
- Pasirinkti duomenų filtravimo metodą
- Pasirinkti klasifikavimo algoritmą
- Nustatyti algoritmo parametrus
- Stebėti klasifikavimo kokybę
- Įrenginio būseną
- Siųsti duomenis

#### **2.1.3. Nefunkciniai reikalavimai**

##### **2.1.3.1. Reikalavimai sistemos išvaizdai**

Bendri reikalavimai smegenų – kompiuterio sąsajos sistemos vartotojo sąsajai:

- Lengvai skaitoma sąsaja;
- Neįkyri sąsaja;
- Profesionali išvaizda;
- Sąveikaujanti sąsaja;
- Informuojanti vartotoją apie atliktus veiksmus sąsaja.

##### **2.1.3.2. Reikalavimai panaudojamumui**

Panaudojamumo paprastumas smegenų – kompiuterio sąsajos sistemai kriterijai:

- Paprastas naudotis neurologams ir kitiems mokslininkams;
- Klaidos kainos mažinimas;

- Galimybė nutraukti bet kokį procesą;
- Procesų ir duomenų vizualizacija.

### 2.1.3.3. Reikalavimai vykdymo charakteristikoms

Užduočių vykdymo greičio apribojimai:

- Maksimalus signalo surinkimo laikas: 5s
- Maksimalus klasifikavimo proceso trukmės laikas: 1s

Viršijus nustatytus apribojimus procesas nutraukiamas ir tolimesni procesai vykdomi su duomenimis, kurie buvo apdoroti per tą laiką.

Maksimalus buferio dydis duomenų nuskaitymui – 5 MB. Jei sugeneruojama daugiau duomenų, tai pertekliniai duomenys ignoruojami.

### 2.1.3.4. Reikalavimai veikimo sąlygoms

Sistema turi veikti *windows* 32 bitų operacinėse sistemose: *windows xp, vista, 7*. Norint pilnai panaudoti sistemą, ji turi veikti kartu su prie kompiuterio prijungtu valdomu įrenginiu ir EEG skaitytuvu.

## 2.2. Architektūros tikslai ir apribojimai

### 2.2.1. Tikslai

Pagrindiniai projektuojamos sistemos architektūros tikslai:

- Sistemos architektūra turi būti lanksti. Turi būti galimybė greitai ir nesudėtingai pritaikyti sistemą naujiems valdomiems įrenginiams ir EEG skaitytuvų modeliams.
- Sistema turi būti sudaryta iš atskirų komponentų.
- Architektūra turi tenkinti šiuos laikinius apribojimus:
  - Maksimalus signalo surinkimo laikas: 5s
  - Maksimalus klasifikavimo proceso laikas: 1s
- Projektuojama sistema turi realizuoti šiuos pagrindinius veiksmus:
  - Naudojant smegenų generuojamus EEG signalus valdyti įrenginį.
  - Apdoroti EEG signalus skirtingais signalų apdorojimo metodais.
  - Klasifikuoti EEG signalus skirtingais klasifikavimo algoritmais.
  - Atvaizduoti signalų vaizdus grafiškai

### 2.2.2. Apribojimai

Sistema realizuojama JAVA programavimo kalba.

### 2.2.3. COTS produktų panaudojimas

Realizuojant sistemą numatoma panaudoti USBReader.CLibrary, EEG skaitytuvo ir sistemos komunikacijai per USB sąsają realizuoti.

### 2.2.4. Projektavimo įrankiai

Projektavimui buvo naudoti šie įrankiai:

- Visual Paradigm – UML diagramoms braižyti.
- Microsoft Word – architektūros specifikacijai dokumentuoti.
- NetBeans – programavimo darbams.

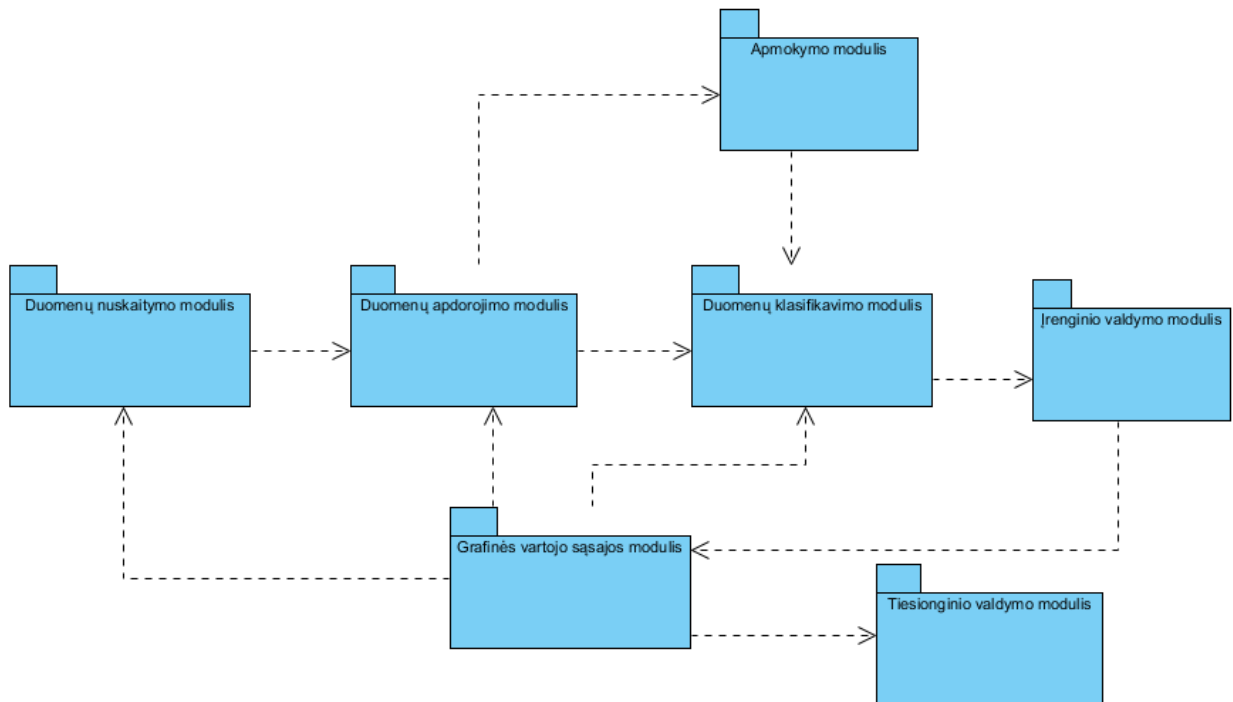
### 2.2.5. Įgyvendinimo strategija

Kompiuterio – smegenų sąsajos sistemai realizuoti naudojama komponentinio projektavimo metodika ir *bottom-up* principas. Atskiri sistemos komponentai realizuojami atskirai, nepriklausomai nuo kitų komponentų. Šie komponentai ištestuojami. Baigus ištestuoti visus komponentus, jie integruojami į vieną sistemą. Atlikus integraciją atliekamas visos sistemos testavimas.

## 2.3. Projektas, formalus aprašas

### 2.3.1. Sistemos statinis vaizdas

Pateikiamas sistemos statinis vaizdas (2.1 pav.).



2.1 pav. Sistemos statinis vaizdas.

Sistema yra suskaidyta į septynis pagrindinius modulius.

Duomenų nuskaitymo modulis realizuoja kompiuterio ir EEG nuskaitymo įrenginio komunikaciją per USB sąsają. Šiame modulyje sąsajos su skirtingais EEG skaitytuvais realizuojamos kaip atskiri moduliai.

Duomenų apdorojimo modulis realizuoja duomenų apdorojimo metodus, reikalingus EEG signalo apdorojimui. Šiame modulyje, skirtingi signalų apdorojimo metodai realizuojami kaip atskiri moduliai.

Apmokymo modulio pagalba įrašomi duomenų rinkiniai reikalingi apmokyti klasifikavimo metodus.

Duomenų klasifikavimo modulyje realizuota EEG duomenų klasifikavimo metodai. Šiame modulyje, atskiri metodai realizuojami kaip atskiri moduliai.

Įrenginio valdymo modulis realizuoja įrenginių valdymą naudojant duomenų klasifikavimą. Šiame modulyje, sąsaja su kiekvienu įrenginiu realizuojama kaip atskiras modulis.

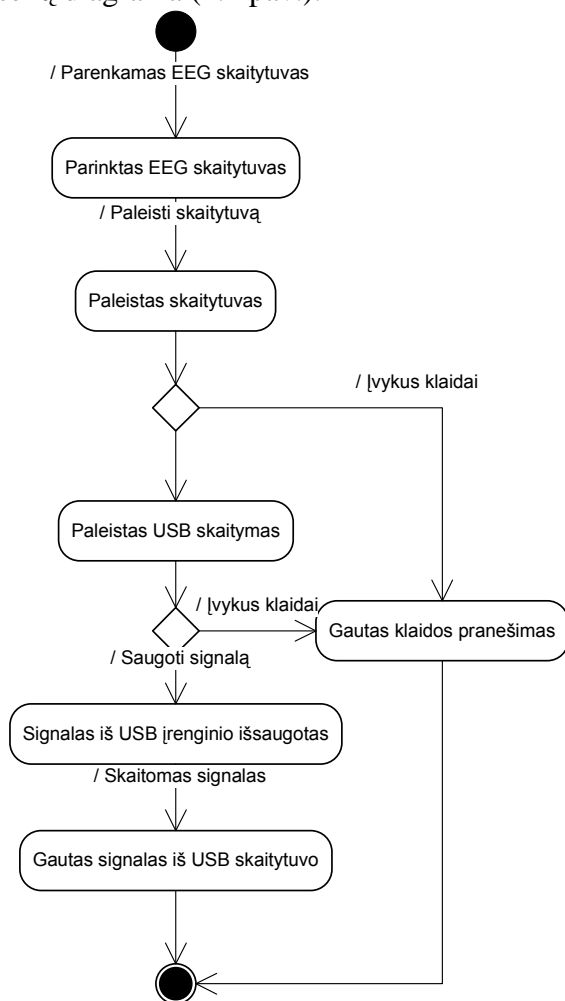
Tiesioginio valdymo modulis realizuoja roboto valdymą naudojantis klaviatūra.

Grafinės vartotojo sąsajos modulyje realizuojamas sąsajos su kiekvienu sistemos moduliu bei papildomas funkcionalumas, numatytas panaudos atvejų modelyje.

## 2.3.2. Būsenų diagramos

### 2.3.2.1. Duomenų nuskaitymo būsenų diagrama.

Duomenų nuskaitymo būsenų diagrama (2.2 pav.).

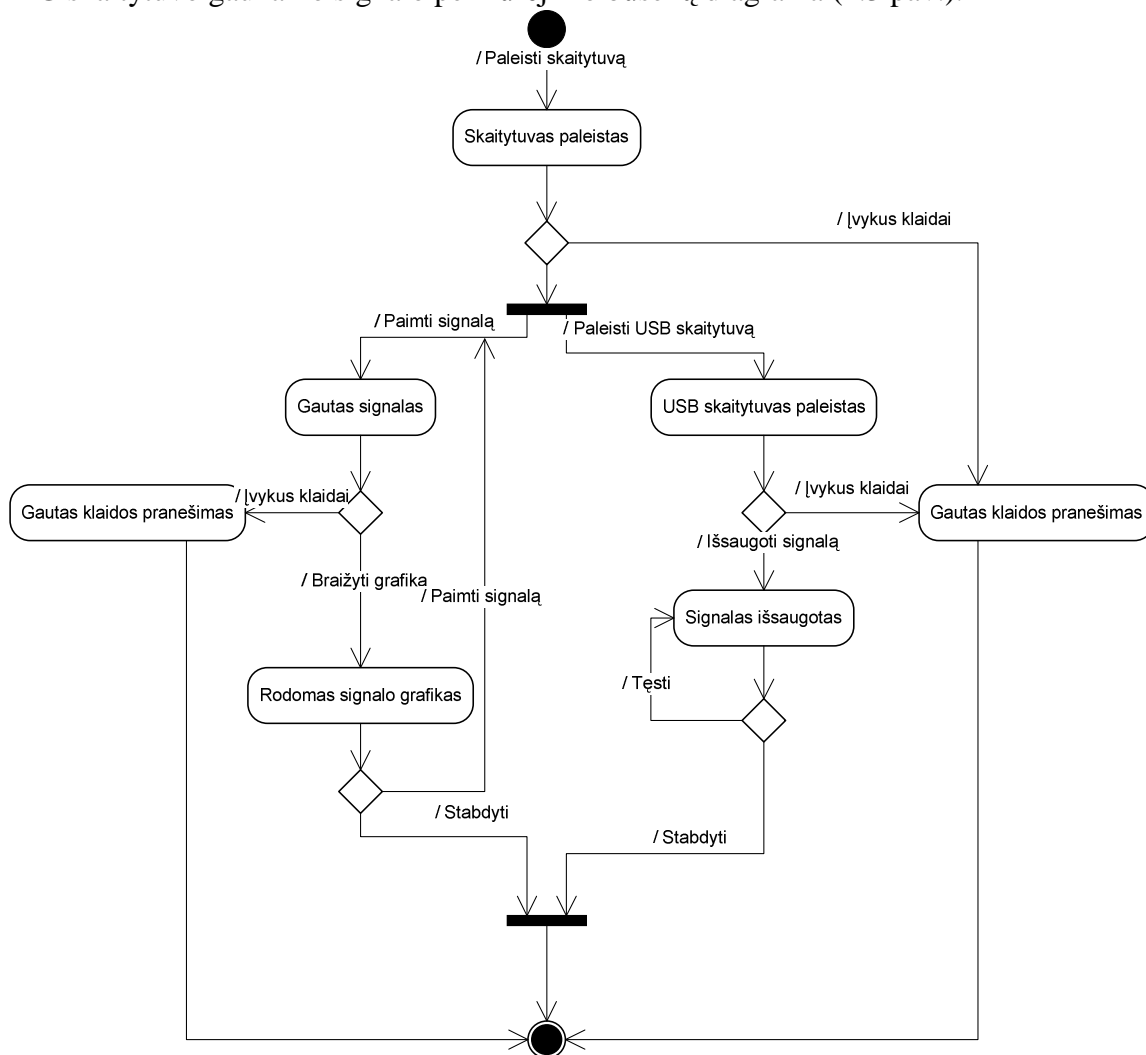


2.2 pav. Duomenų nuskaitymo būsenų diagrama.

EEG signalui nuskaityti parinkamas skaitytuvas, paleidžiama skaitymo modulis, paleidžiamas duomenų skaitymas iš USB ir išsaugomas signalas. Jei paleidžiant skaitymą arba skaitant signalą įvyksta klaida, tai skaitymas iškart nutraukiamas.

### 2.3.2.2. Iš EEG skaitytuvo gaunamo signalo peržiūrėjimo būsenų diagrama

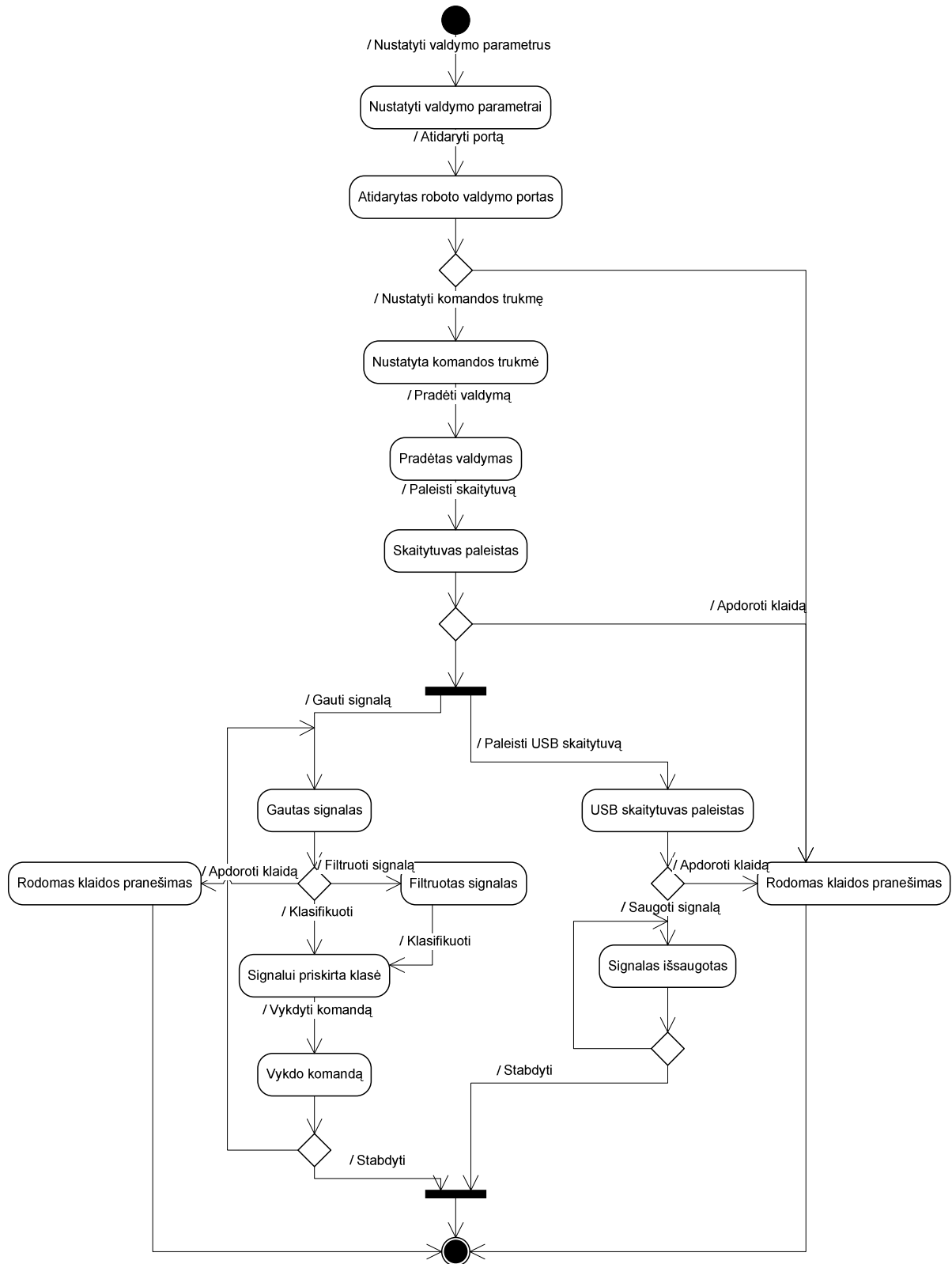
Iš EEG skaitytuvo gaunamo signalo peržiūrėjimo būsenų diagrama (2.3 pav.).



2.3 pav. Iš EEG skaitytuvo gaunamo signalo peržiūrėjimo būsenų diagrama.

Elektroencefalogramos signalui peržiūrėti pirma paleidžiamas skaitytuvas, kadangi signalą reikia, paleidus cikle, pastoviai skaityti iš USB, tai skaitymas ir vaizdavimas vykdomi atskirose gijose. Signalų skaitymo gijoje pastoviai skaitomas ir išsaugomas signalas, tol kol negaunama stabdymo signalas arba neįvyksta klaida. Signalų vaizdavimo gijoje pastoviai skaitomas išsaugotas signalas ir braižomas grafikas, kol negaunamas stabdymo signalas arba neįvyksta klaida.

**2.3.2.3. Įrenginio valdymo būsenų diagrama**  
 Įrenginio valdymo būsenų diagrama (2.4 pav.).



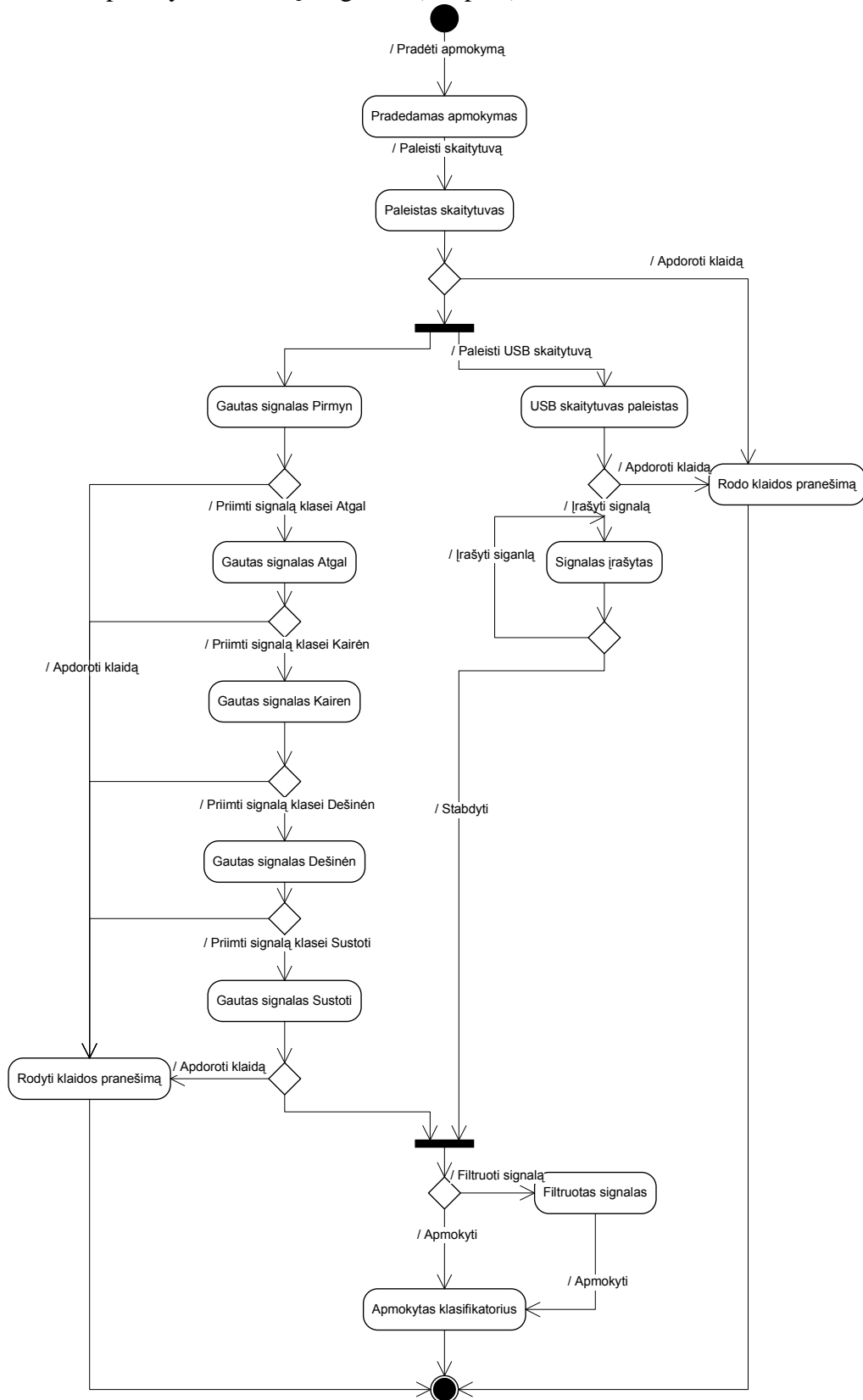
**2.4 pav.** Įrenginio valdymo būsenų diagrama.

Prieš pradėdant valdyti įrenginį, reikia nustatyti valdymo ir valdomo įrenginio parametrus ir paleisti EEG skaitytuvą. Signalų nuskaitymas ir įrenginio valdymas vykdomas atskirose gijose. Signalų skaitymo gijoje paleidžiamas USB skaitytuvas ir cikle skaitomas ir saugomas signalas. Valdymo gijoje, nustatytais laiko intervalais, nuskaitymas išsaugotas signalas, jei parinkta naudoti filtravimo metodus tai filtruoja, klasifikuoja ir vykdo komandą.



### 2.3.2.4. Klasifikatoriaus apmokymo būsenų diagrama

Klasifikatoriaus apmokymo būsenų diagrama (2.5 pav.).



2.5 pav. Klasifikatoriaus apmokymo būsenų diagrama.

Apmokant klasifikatorių, EEG signalų įrašymo metu, kuriamos dvi gijos. Vienoje gijoje iš EEG skaitytuvo nuskaitytas signalas ir išsaugomas, kitoje parenkama, kuriai klasei priklauso signalas. Kai visi signalai įrašyti, tada apmokomas klasifikatorius. Komandų kiekis priklauso nuo parinkto valdomo įrenginio.

## **2.4. Antro skyriaus išvados**

Šiame skyriuje pateikta reikalavimai smegenų – kompiuterio sąsajos sistemai, tikslai, apribojimai, sistemos statinis vaizdas ir pagrindinės būsenų diagramos.

### 3. SMEGENŲ – KOMPIUTERIO SAŠAJOS SISTEMOS TYRIMAS

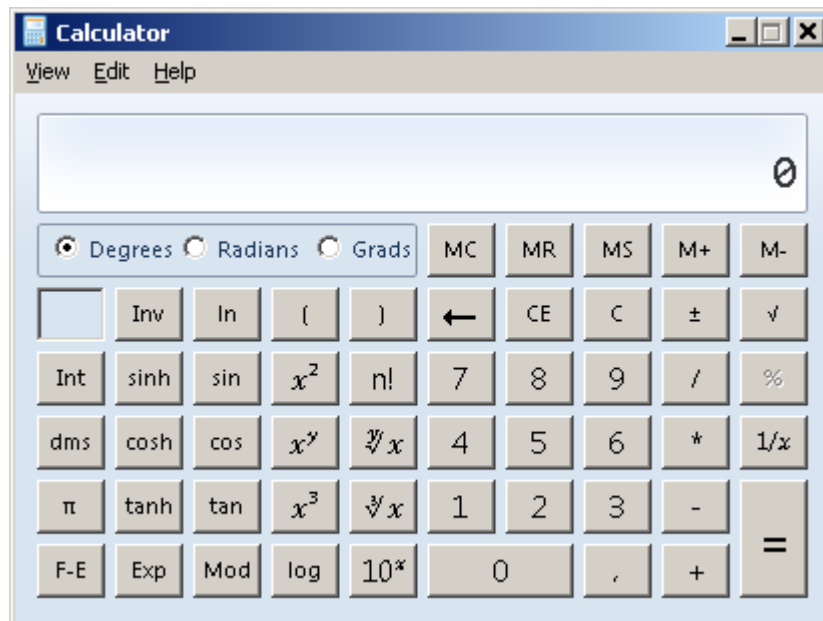
#### 3.1. Eksperimento aprašymas

Eksperimentas atliekamas bandant suvaldyti virtualų įrenginį. Kaip virtualus įrenginys naudojama pelė. Klasifikuojamų signalų trukmė: 0,5s. Klasifikatorius apmokomas trim būdais:

- Mintimis įsivaizduojant, kad atliekami kažkokie veiksmai, klasifikatorius apmokomas atskirti penkias valdymo komandas: judėti į viršų, apačią, kairę, dešinę ir paspausti dešinę klavišą.
- Mintimis įsivaizduojant, kad atliekami kažkokie veiksmai, klasifikatorius apmokomas atskirti dvi valdymo komandas: pradėti vykdymą ir stabdyti vykdymą.
- Naudojant akių mirksėjimą, klasifikatorius apmokomas atskirti dvi valdymo komandas: pradėti vykdymą ir stabdyti vykdymą.

Šiam eksperimentui parinktos užduotys taip, kad atspindėtų dirbant ir pramogaujant atliekamus veiksmus. Eksperimento metu atliekamos keturios užduotys:

- Kalkuliatoriumi (3.1 pav.) suskaičiuoti:  $(19,27 * 3762 / 3075 - 76)^2 + 81 = 2829,35965504$ . Šioje formulėje skaičiai parinkti taip, kad daugumai jų įvesti pelė reikėtų judinti ir horizontaliai ir vertikalčiai.



3.1 pav. Kalkuliatorius.

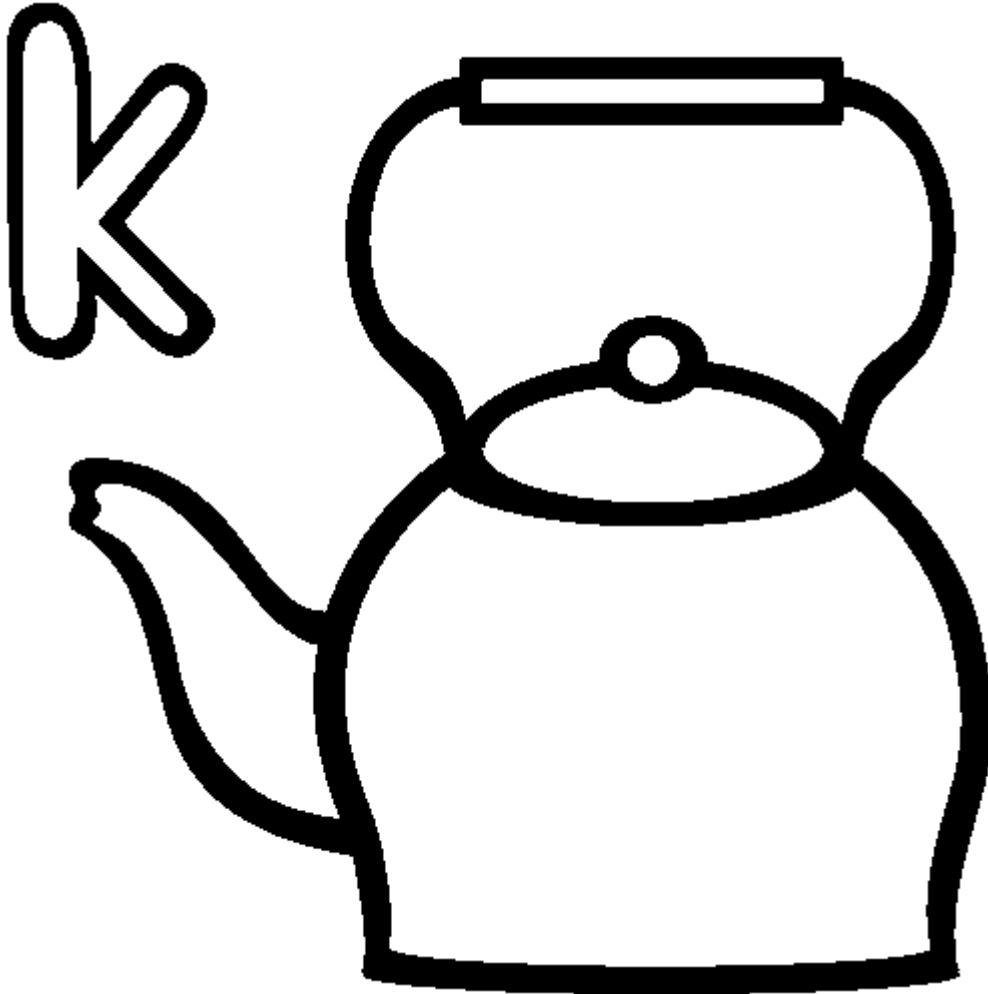
- Naudojant virtualią klaviatūrą (3.2. pav.) įvesti tekstą:

„Mūsų sukurtoje sistemoje naudojamas nesudėtingas ir patogus klasifikavimo būdas – komandų klasifikavimas pagal signalo lygį.“



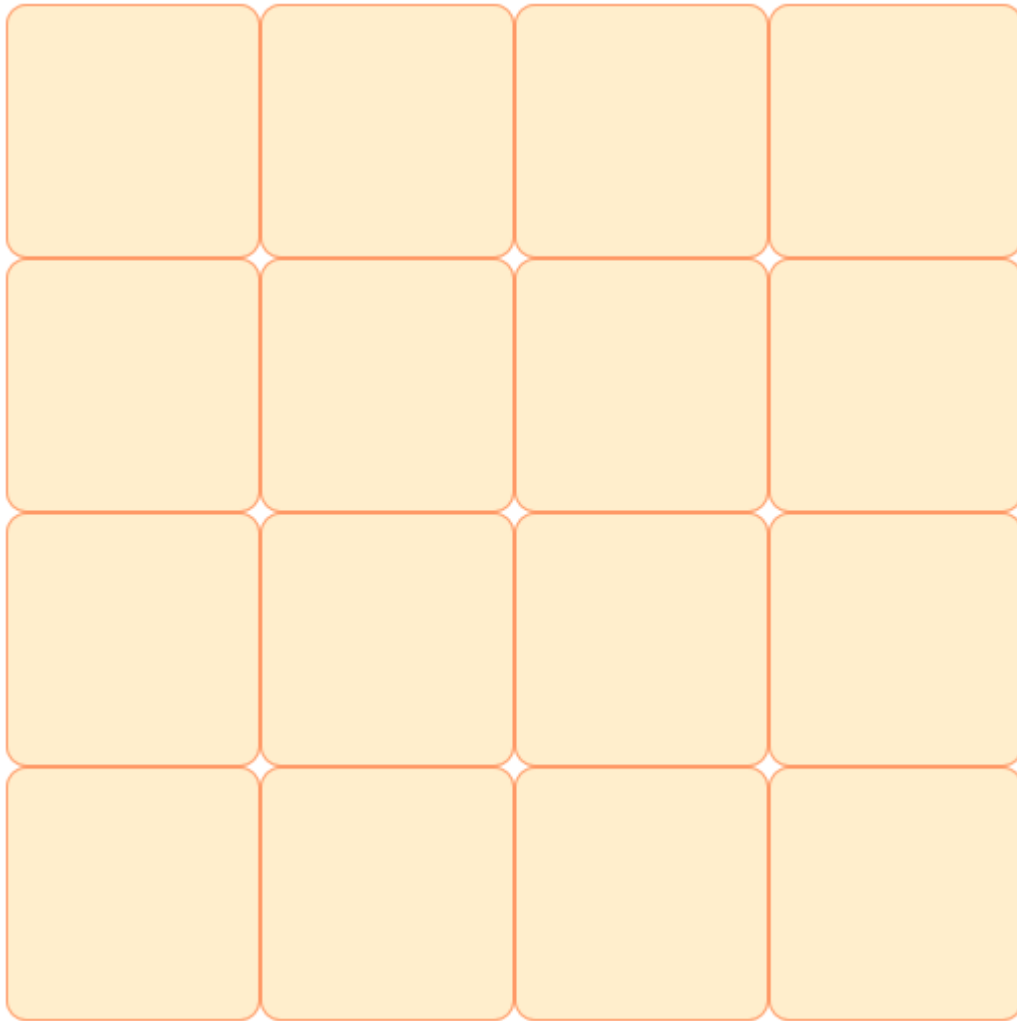
3.2 pav. Virtuali klaviatūra.

- c. Elektroninėje parduotuvėje <http://www.kilobaitas.lt> išsirinkti nešiojamą kompiuterį, planšetę, MP3 grotuvą. Kiekvieną produktą pasirinkus, atidaryti produkto aprašymą ir įdėti į prekių krepšelį. Pasirinkus visus produktus, peržiūrėti krepšelį.
- d. Atidaryti puslapį <http://www.thecolor.com/Coloring/Kettle.aspx> ir nuspalvinti paveikslą (3.3 pav.) tai, kad kiekvieno lauko spalva skirtųsi nuo su tuo lauku besiribojančių laukų spalvų.



3.3 pav. Paveikslėlis spalvojimui.

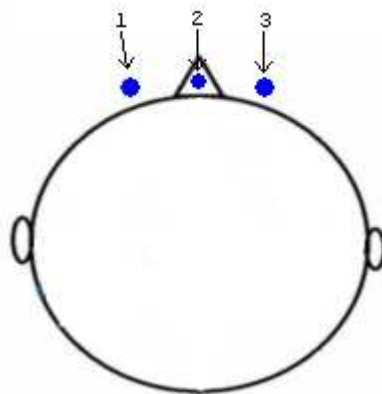
- e. Žaidimas „Memory“ (3.4. pav.). Atidaryti puslapį <http://mypuzzle.org/find-the-pair> ir pasirinkus aštuonias poras sudėlioti paveikslėlius.



**3.4 pav.** „Memory“ žaidimas.

### **3.2. Naudojamas elektroencefalogramos skaitytuvas**

Ekspimento metu naudojamas OCZ NIA (Neuro Impuls Actuator) elektroencefalogramos ir elektromiogramos skaitytuvas. Jame išviso yra trys jutikliai, kurie dedami ant kaktos (3.5 pav.). Skaitytuvas gražina skirtumą tarp pirmo ir trečio jutiklio, o antras jutiklis naudojamas kaip žemė.



**3.5 pav.** Skaitytuvo jutiklių išdėstymas.

### 3.3. Rezultatai

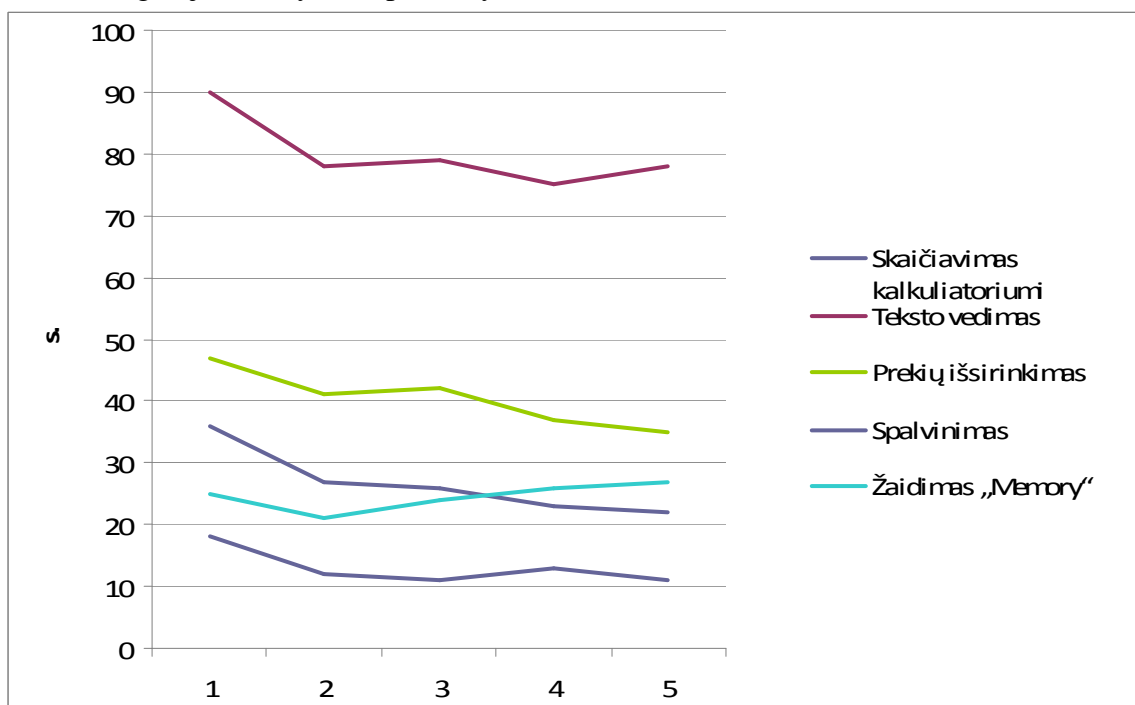
#### 3.3.1. Užduočių įvykdymo rezultatai naudojant pelę

Užduočių įvykdymo laikai sekundėmis, naudojant pelę (3.1 lentelė, 3.6 pav.).

3.1 lentelė. Užduočių įvykdymo laikai, naudojant pelę

Bandymo numeris	Skaičiavimas kalkuliatoriumi	Teksto vedimas	Prekių išsirinkimas	Spalvinimas	Žaidimas „Memory“
1.	36	90	47	18	25
2.	27	78	41	12	21
3.	26	79	42	11	24
4.	23	75	37	13	26
5.	22	78	35	11	27
Vidutinis vykdymo laikas	26,8	80	40,4	13	24,6
dispersija	5,540758	5,787918	4,669047	2,915476	2,302173

Kai užduotys yra nuspėjamos pastebimas apsimokymas ir kiekvieno bandymo metu vykdymo laikas gerėja, vyksta apsimokymas. Kai užduotis nėra nuspėjama ir kiekvieną kartą keičiasi, tai vykdymo laikas negerėja ir nevyksta apsimokymo.



3.6 pav. Užduočių įvykdymo laikai, naudojant pelę.

#### 3.3.2. Klasifikavimo tikslumas su penkiomis komandomis.

Klasifikavimo tikslumas, klasifikatorių apmokant atskirti penkias valdymo komandas (3.2 lentelė). Valdymui naudojamos mintys.

3.2 lentelė. Klasifikavimo tikslumas, apmokant penkias valdymo komandas

Klasifikavimo algoritmas	Tikslumas, %
Atsitiktinis miškas	24,33
Atsitiktinis miškas (apmokant kaip binarinį)	21,66
Balsuojantis perceptronas	19
Daugiasluoksnis perceptronas	19
Daugiasluoksnis perceptronas (apmokant kaip binarinį)	19

Kadangi geriausias pasiektas tikslumas tėra 24,33% tai užduočių vykdyti nepavyko.

### 3.3.3. Klasifikavimo tikslumas su dviem komandom, naudojant mintis

Klasifikavimo tikslumas, klasifikatorių apmokant atskirti dvi valdymo komandas (3.3 lentelė). Valdymui naudojamos mintys.

**3.3 lentelė.** Klasifikavimo tikslumas, apmokant dvi valdymo komandas, naudojant mintis

Klasifikavimo algoritmas	Tikslumas, %
Atsitiktinis miškas	44
Atsitiktinis miškas (apmokant kaip binarinį)	40,66
Balsuojantis perceptronas	36,66
Daugiasluoksnis perceptronas	33,33
Daugiasluoksnis perceptronas (apmokant kaip binarinį)	33,33

Kadangi geriausias pasiektas tikslumas tėra 44% tai užduočių vykdyti nepavyko.

### 3.3.4. Klasifikavimo tikslumas su dviem komandom, naudojant kaktos raumenis

Klasifikavimo tikslumas, klasifikatorių apmokant atskirti dvi valdymo komandas (3.4 lentelė). Valdymui naudojama kairės ir dešinės akies mirksėjimai.

**3.4 lentelė.** Klasifikavimo tikslumas, apmokant dvi komandas, naudojant akių mirksnius

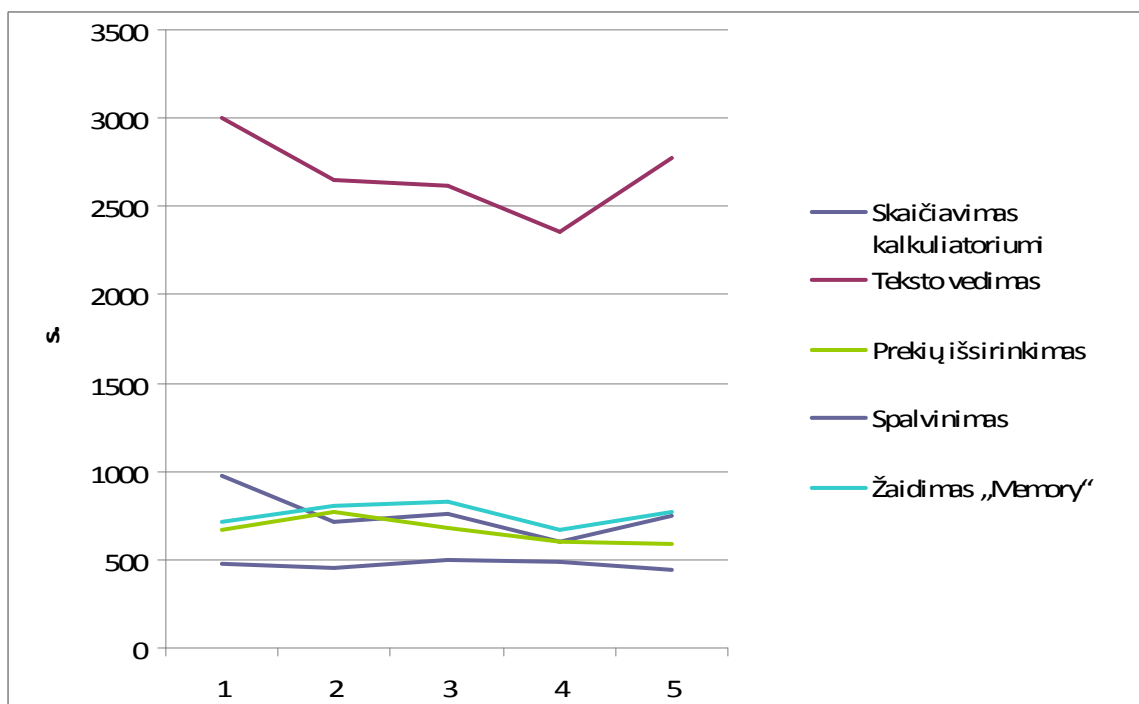
Klasifikavimo algoritmas	Tikslumas, %
Atsitiktinis miškas	87,33
Atsitiktinis miškas (apmokant kaip binarinį)	85,33
Balsuojantis perceptronas	63,33
Daugiasluoksnis perceptronas	69,33
Daugiasluoksnis perceptronas (apmokant kaip binarinį)	68

Pakankamą tikslumą pavyko pasiekti tik naudojant atsitiktinio miško algoritmą (tiek apmokant kaip binarinį, tiek vieną klasifikatorių apmokant visų komandų). Užduočių vykdymo laikai sekundėmis, naudojant atsitiktinio miško klasifikatorių ir jį apmokius atpažinti dvi komandas (3.5 lentelė, 3.7 pav.).

**3.5 lentelė.** Užduočių įvykdymo laikai, valdant akių mirksniais

Bandymo numeris	Skaičiavimas kalkuliatoriumi	Teksto vedimas	Prekių išsirinkimas	Spalvinimas	Žaidimas „Memory“
1.	977	3004	669	471	713
2.	714	2645	773	457	808
3.	761	2616	679	501	828
4.	602	2361	600	491	670
5.	753	2774	586	444	773
Vidutinis vykdymo laikas	761,4	2680	661,4	472,8	758,4
dispersija	136,2068	234,9968	74,62774	23,49894	65,94164

Kadangi skaičiavimas kalkuliatoriumi buvo pirma atliekama užduotis, tai jos atlikimo laiką labiausiai įtakoja mokymasis valdyti pelę, naudojant smegenų – kompiuterio sąsają.



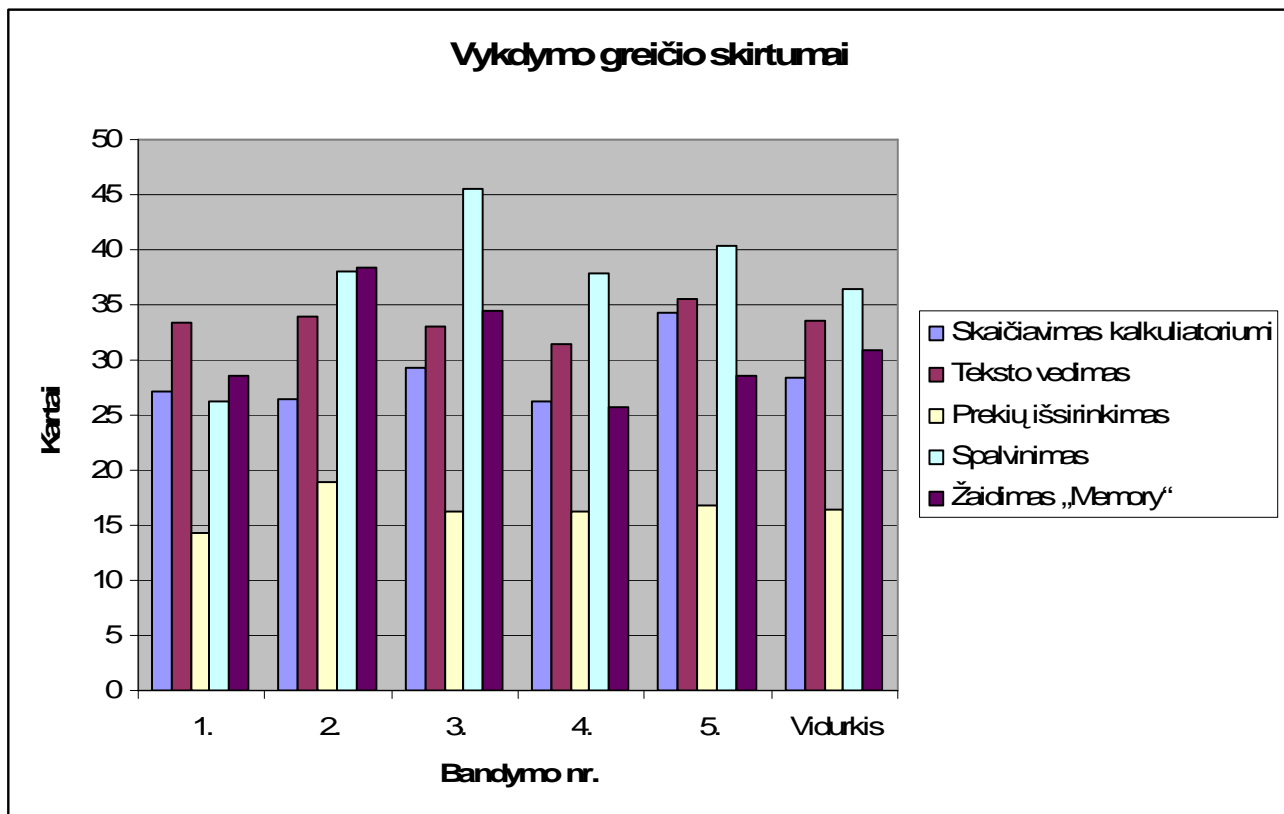
3.7 pav. Užduočių įvykdymo laikai, valdant akių mirksniais.

Kiek kartų skiriasi užduočių vykdymo laikas naudojant smegenų – kompiuterio sąsają ir įprastą valdymo būdą, pateikta 3.6 lentelėje ir 3.8 paveiksle.

3.6 lentelė. Užduočių vykdymo laikų skirtumai, kartais.

Bandymo numeris	Skaičiavimas kalkuliatoriumi	Teksto vedimas	Prekių išsirinkimas	Spalvinimas	Žaidimas „Memory“
1.	27,13889	33,37778	14,23404	26,16667	28,52
2.	26,44444	33,91026	18,85366	38,08333	38,47619
3.	29,26923	33,11392	16,16667	45,54545	34,5
4.	26,17391	31,48	16,21622	37,76923	25,76923
5.	34,22727	35,5641	16,74286	40,36364	28,62963
Vidurkis	28,41045	33,5	16,37129	36,36923	30,82927





3.8 pav. Užduočių vykdymo skirtumai.

Informacijos perdavimo greitis (Information transfer rate) skaičiuojamas pagal formulę (3.1) [24]

$$ITR = \frac{B}{t} \quad (3.1)$$

čia  $t$  – vieno veiksmo atlikimo laikas;

$B$  – bitai / operacijai, apskaičiuojama pagal formulę (3.2) [24]

$$B = \log_2 N + P \log_2 P + (1 - P) \log_2 \frac{1 - P}{N - 1} \quad (3.2)$$

čia  $N$  – veiksmų kiekis,  $P$  – klasifikatoriaus tikslumas.

Vykdam užduotis pasiektas informacijos perdavimo greitis pateiktas 3.7 lentelėje.

3.7 lentelė. Pasiektas informacijos perdavimo greitis, bit/min.

Bandymo numeris	Skaičiavimas kalkuliatoriumi	Teksto vedimas	Prekių išsirinkimas	Spalvinimas	Žaidimas „Memory“
1.	5,174239	13,3435	2,33912	3,785543	7,478588
2.	7,080156	15,15458	2,024413	3,901512	4,600499
3.	6,64288	15,32258	2,30467	3,558864	7,119153
4.	8,397395	16,9775	2,608119	3,631346	7,136099
5.	6,713455	14,44984	2,670429	4,015745	6,898103
Vidurkis	6,801625	15,0496	2,38935	3,778602	6,646488

Vykdam prekių išsirinkimo užduotį, gautas mažas informacijos perdavimo greitis todėl, kad reikia laukti kol naršyklė atsiūs puslapį ir su pele vaikščioti po visą ekraną. Spalvinant taip pat reikėjo vaikščioti po didelį ekrano plotą, todėl gavosi mažas informacijos perdavimo greitis. Vedant tekstą, su pele reikėjo vaikščioti po santykinai mažą plotą, taip pat dėl didesnių klavišų pelės judėjimo žingsnis buvo padidintas, todėl gavosi didelis informacijos perdavimo greitis.

### 3.4. Trečio skyriaus išvados

Šiame skyriuje atliktas smegenų – kompiuterio sąsajos tyrimas. Tyrimui naudota OCZ NIA elektroencefalogramos ir elektromiogramos skaitytuvas. Klasifikavimui naudota daugiasluoksnis perceptronas, balsuojantis perceptronas ir atsitiktinio miško klasifikavimo algoritmai. Buvo bandoma mintimis išskirti penkias ir dvi valdymo komandas. Kaktos raumenų pagalba buvo bandoma išskirti dvi valdymo komandas. Geriausias klasifikavimo tikslumas su penkiom komandom, naudojant mintis – 24,33%. Geriausias klasifikavimo tikslumas su dviem komandom, naudojant mintis – 44%. Geriausias klasifikavimo tikslumas su dviem komandom, naudojant kaktos raumenis – 87,33%.

Valdant mintimis nebuvo pasiektas pakankamas tikslumas, dėl to buvo daroma daug klaidų ir užduotys nebuvo įvykdytos. Su kaktos raumenimis, naudojant atsitiktinio miško klasifikavimo algoritmą pavyko pasiekti pakankamą tikslumą ir įvykdyti praktines užduotis. Geriausias pasiektas informacijos perdavimo greitis: 16,97 bit/min., kitų autorių pasiekti rezultatai yra 5 – 25 bit/min. [25].

Sąsajos privalumai:

- Lengvai suprantama vartotojui
- Vartotojas padaręs klaidą, gali pasitaisyti

Trūkumai:

- Reikia apmokyti klasifikatorių
- Išskiriama mažai valdymo klasių, todėl valdymas yra lėtas.

#### 4. REZULTATŲ APIBENDRINIMAS IR IŠVADOS

Šiame magistriniame darbe buvo:

- Sukurtas smegenų – kompiuterio sąsajos prototipas, kuriame naudojami daugiasluoksnis perceptronas, balsuojantis perceptronas ir atsitiktinis miškas. Realizuota kompiuterio pelės valdymas. Naudojant šią sąsają žmonės su fizine negalia galės bendrauti su aplinka (pvz.: siųsti elektroninius laiškus, pakviesti pagalbos), pagerinti savo gyvenimo kokybę.
- Patikrintas OCZ NIA skaitytuvu nuskaityto signalo klasifikavimo tikslumas. Geriausias klasifikavimo tikslumas su penkiom komandom, naudojant mintis – 24,33%. Geriausias klasifikavimo tikslumas su dviem komandom, naudojant mintis – 44%. Geriausias klasifikavimo tikslumas su dviem komandom, naudojant kaktos raumenis – 87,33%. Pakankamas klasifikavimo tikslumas buvo pasiektas tik naudojant atsitiktinio miško algoritmą ir valdymui naudojant kaktos raumenis. Iš tirtų klasifikavimo algoritmų geriausiai veikė atsitiktinio miško algoritmas.
- Ištirta smegenų – kompiuterio sąsajos sistema atliekant praktines užduotis. Praktinės užduotys buvo atliekamos valdant kompiuterio pelę. Elektroencefalograma nuskaityta, naudojant OCZ NIA skaitytuvą. Bandant užduotis įvykdyti naudojant mintis, dėl per mažo klasifikavimo tikslumo, buvo daroma daug klaidingų veiksmų ir nepavyko įvykdyti užduočių. Užduotis pavyko įvykdyti tik naudojant atsitiktinio miško klasifikavimo algoritmą, o valdymui naudojant kaktos raumenis. Atliekant eksperimentą pastebėta, kad po to kai mirktelima akimi, signalui reikia iki 1 s. Nusistovėti.

Smegenų – kompiuterio sąsajos privalumai ir trūkumai:

- Privalumai: lengvai suprantama vartotojui. Vartotojas padaręs klaidą, gali pasitaisyti.
- Trūkumai: reikia apmokyti klasifikatorių. Išskiriama mažai valdymo klasių, todėl valdymas lėtas.

## 5. LITERATŪRA

- [1] **Selim A.E., Wahed M.A., Kadah Y.M.** Machine learning methodologies in brain-computer interface systems. *Proc. of Biomedical Engineering Conference, CIBEC 2008*, pp.1-5, 2009.
- [2] **I. Martisius, M. Vasiljevas, K. Sidlauskas, R. Turcinas, I. Plauska, R. Damasevicius.** Design of a Neural Interface Based System for Control of Robotic Devices . In T. Skersys, R. Butleris, and R. Butkiene (Eds.), *ICIST 2012 : Proceedings of the 18th international conference on Information and Software Technologies*, Kaunas, Lithuania, September 13-14, 2012. Springer-Verlag, CCIS 319, pp. 297-311, 2012.
- [3] **Viktor Medvedev** 2007. „Tiesioginio Sklidimo Neuroninių Tinklų Taikymo Daugiamatiams Duomenims Vizualizuoti Tyrimai“. Daktaro disertacija. [Žiūrėta 2011-11-10] prieiga internete: <[http://www.mii.lt/files/disert\\_08\\_medvedev.pdf](http://www.mii.lt/files/disert_08_medvedev.pdf)>
- [4] Jain, A.K.; Jianchang Mao; Mohiuddin, K. M., "Artificial neural networks: a tutorial," *Computer* , vol.29, no.3, pp.31,44, Mar 1996 doi: 10.1109/2.485891
- [5] **Jin, L.V.; Chen Guo; Zhi-peng Shen; Min Zhao; Yuan Zhang**, "Summary of Artificial Neuron Model Research," *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE* , vol., no., pp.677,682, 5-8 Nov. 2007 doi: 10.1109/IECON.2007.4459952
- [6] **Haykin, S.** *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1998.
- [7] **Freund Y., Schapire R. E.** Large Margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277-296, 1999.
- [8] **P. Venkatesan and S. Anitha**, "Application of radial basis function neural network for diagnosis of diabetes mellitus", *Current Science*, vol. 91, pp 1195 – 1999, 2006.
- [9] **Cruse Holk**, *Neural Networks as Cybernetic Systems*, 2nd and revised edition. Brains, Minds and Media, Bielefeld, Germany, 2006.
- [10] **Nauck, D.; Klawonn, F.; Kruse, R.** 1996. *Foundations of Neuro--Fuzzy Systems*. Wiley, Chichester
- [11] **Ghaderi, K.; Akhlgian, F.; Moradi, P.**, "A new digital image watermarking approach based on DWT-SVD and CPPN-NEAT," *Computer and Knowledge Engineering (ICCKE), 2012 2nd International eConference on* , vol., no., pp.12,17, 18-19 Oct. 2012 doi: 10.1109/ICCKE.2012.6395344
- [12] **Guang-Bin Huang; Qin-Yu Zhu; Mao, K.Z.; Chee-Kheong Siew; Saratchandran, P.; Sundararajan, N.**, "Can threshold networks be trained directly?," *Circuits and Systems II: Express Briefs, IEEE Transactions on* , vol.53, no.3, pp.187,191, March 2006 doi: 10.1109/TCSII.2005.857540
- [13] **M. Barbarosou, N.G. Maratos**, "TRAINING OF PERCEPTRON NEURAL NETWORK USING PIECEWISE LINEAR ACTIVATION FUNCTION" *Journal of Applied Electromagnetism*;Jun 2011, Vol. 13 Issue 1, p10
- [14] **Simon Haykin**, *Neural Networks: A Comprehensive Foundation (2nd Edition)*. Prentice Hall PTR Upper Saddle River, NJ, USA ©1998
- [15] **Hazrati, Mehrnaz Kh.; Erfanian, Abbas**; , "An on-line BCI for control of hand grasp sequence and holding using adaptive probabilistic neural network," *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE* , vol., no., pp.1009-1012, 20-25 Aug. 2008 doi: 10.1109/IEMBS.2008.4649326 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4649326&isnumber=4649055>
- [16] **Specht, Donald F.**, "Probabilistic neural networks for classification, mapping, or associative memory," *Neural Networks, 1988., IEEE International Conference on* , vol., no., pp.525,532 vol.1, 24-27 July 1988 doi: 10.1109/ICNN.1988.23887
- [17] **Paul J. Werbos (1994)**. *The Roots of Backpropagation. From Ordered Derivatives to Neural Networks and Political Forecasting*. New York, NY: John Wiley & Sons, Inc.

- [18] **Kohonen, Teuvo (1982).** "Self-Organized Formation of Topologically Correct Feature Maps". *Biological Cybernetics* 43 (1): 59-69.
- [19] **Hu Y.H.** Handbook of neural network signal processing. CRC Press, 2002, ISBN: 0-8493-2359-2
- [20] **S. Kaski, (1997).** Data Exploration Using Self-Organizing Maps PhD thesis, Helsinki University of Technology, Department of Computer Science and Engineering, [žiūrėta 2013-05-02]. Prieiga per internetą <<http://www.cis.hut.fi/~sami/thesis/>>
- [21] **Leo Breiman,** "Random Forests", *Machine Learning*, 45(1):5-32, 2001.
- [22] **Kirmiziansan, E.; Bayraktaroglu, Z.; Gurvit, H.; Keskin, Y.; Emre, M.; Demiralp, T.** (2006). "Comparative analysis of event-related potentials during Go/NoGo and CPT: Decomposition of electrophysiological markers of response inhibition and sustained attention". *Brain Research* 1104 (1): 114-128. doi:10.1016/j.brainres.2006.03.010. PMID 16824492
- [23] **Elgendi, M.; Vialatte, F.; Cichocki, A.; Latchoumane, C.; Jaesung Jeong; Dauwels, J.,** "Optimization of EEG frequency bands for improved diagnosis of Alzheimer disease," *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, vol., no., pp.6087,6091, Aug. 30 2011-Sept. 3 2011  
doi: 10.1109/IEMBS.2011.6091504
- [24] **Graimann, B., Allison, B., and Pfurtscheller, G.** 2011. *Brain-Computer Interfaces: Revolutionizing Human-Computer Interaction.* Springer Verlag, Berlin.
- [25] **Wolpaw, J.R., Birbaumer, N., Heetderks, W.J., McFarland, D.J., Peckham, P.H., Schalk, G., Donchin, E., Quatrano, L.A., Robinson, C.J., Vaughan, T.M.** Brain-computer interface technology: A review of the first international meeting. *IEEE Trans. on Rehabilitation Engineering*, 8, 2 (2000), 164--173.