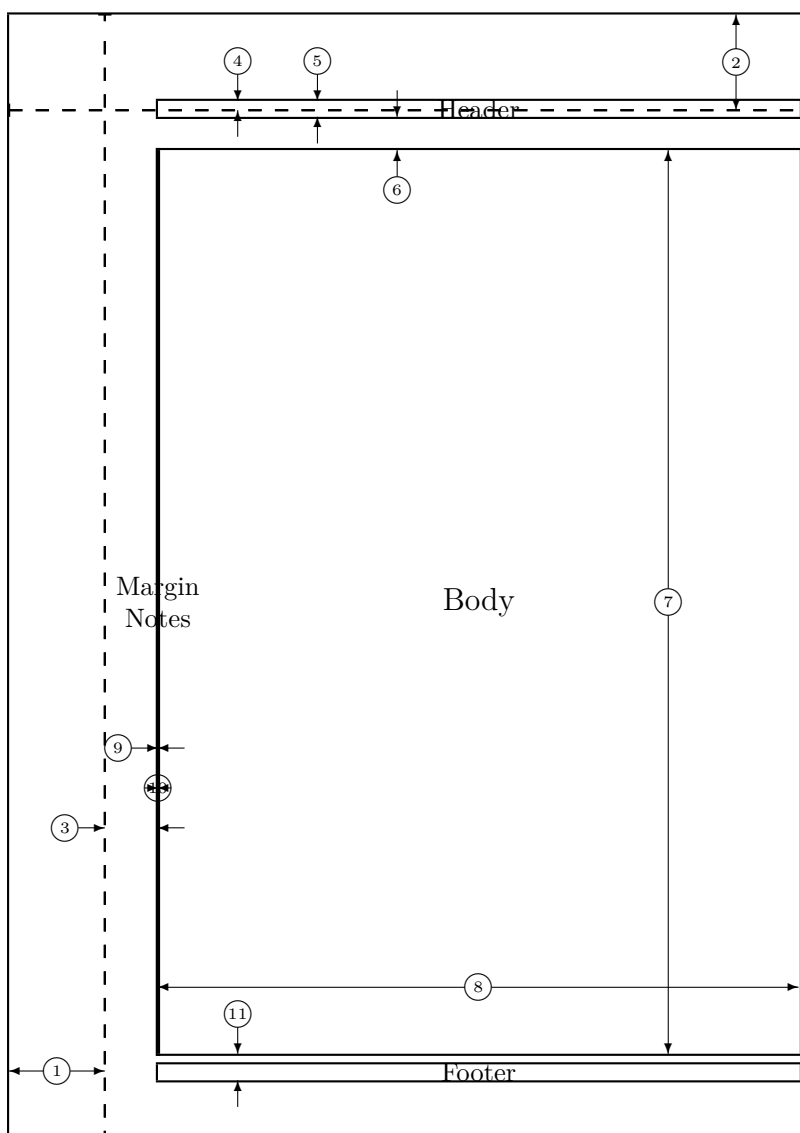


- | | | | |
|----|------------------------------------|----|---|
| 1 | one inch + <code>\hoffset</code> | 2 | one inch + <code>\voffset</code> |
| 3 | <code>\oddsidemargin = 70pt</code> | 4 | <code>\topmargin = -7pt</code> |
| 5 | <code>\headheight = 12pt</code> | 6 | <code>\headsep = 25pt</code> |
| 7 | <code>\textheight = 680pt</code> | 8 | <code>\textwidth = 483pt</code> |
| 9 | <code>\marginparsep = 0pt</code> | 10 | <code>\marginparwidth = 0pt</code> |
| 11 | <code>\footskip = 20pt</code> | | <code>\marginparpush = 7pt</code> (not shown) |
| | <code>\hoffset = 0pt</code> | | <code>\voffset = 0pt</code> |
| | <code>\paperwidth = 597pt</code> | | <code>\paperheight = 845pt</code> |



- | | | | |
|----|-------------------------------------|----|---|
| 1 | one inch + <code>\hoffset</code> | 2 | one inch + <code>\voffset</code> |
| 3 | <code>\evensidemargin = 40pt</code> | 4 | <code>\topmargin = -7pt</code> |
| 5 | <code>\headheight = 12pt</code> | 6 | <code>\headsep = 25pt</code> |
| 7 | <code>\textheight = 680pt</code> | 8 | <code>\textwidth = 483pt</code> |
| 9 | <code>\marginparsep = 0pt</code> | 10 | <code>\marginparwidth = 0pt</code> |
| 11 | <code>\footskip = 20pt</code> | | <code>\marginparpush = 7pt</code> (not shown) |
| | <code>\hoffset = 0pt</code> | | <code>\voffset = 0pt</code> |
| | <code>\paperwidth = 597pt</code> | | <code>\paperheight = 845pt</code> |



Turinys

Žymenys ir santrumpos	10
1. Įvadas. Užduoties analizė	11
2. Informacinių Parkinsono ligos diagnostikos sistemų apžvalga	14
3. Signalų analizės programos kūrimas	22
3.1. Bendros programos struktūrinės schemos sudarymas	22
3.2. Pirminio signalų apdorojimo programos kūrimas	23
3.3. Požymių išskyrimo programos kūrimas	29
3.3.1. Dimensijų mažinimas	29
3.3.2. Savybių tyrimas	31
3.4. Požymių klasifikavimo programos kūrimas	38
3.5. Duomenų analizės programos kūrimas	42
4. Signalų analizės programos įgyvendinimas	44
4.1. Bendro programos algoritmo schemos sudarymas	44
4.2. Požymių klasifikavimo programos algoritmo schemos sudarymas	45
4.3. Signalų analizės programos įgyvendinimas	48
5. Signalų analizės programos patikra	50
5.1. Eksperimentų plano rengimas	50
5.2. Duomenų eksperimentams rengimas	51
5.3. Programos patikros rezultatai	51
6. Rezultatų apibendrinimas	54



Žymenys ir santrumpos

- VŽRJ - Vertikali žemės reakcijos jėga (angl. Vertical Ground Reaction Force);
- PCA - principinė komponentų analizė (angl. Principal Component Analysis);
- LDA - linijinė diskriminanto analizė (angl. Linear Diskriminant Analysis);
- Co - kontrolinis subjektas, kurio ligos istorijoje nebuvo užregistruota jokių neurologinių susirgimų;
- Pt - Parkinsono liga sergantis subjektas;



1. Įvadas. Užduoties analizė

Darbo tema yra Parkinsono ligos eigos stebėjimo programa. Parkinsono liga yra dažniausiai pasitaikantis neurodegeneracinis judėjimo sutrikimas. Ankstyva ligos diagnozė ir efektyvus terapinis stebėjimas yra būtinas pacientų gydymui ir sveikatos priežiūros kainai sumažinti. Šiuo metu neegzistuoja gydytojų patvirtintos objektyvios ir vieningos vertinimo sistemos, kuri tiksliai atpažintų Parkinsono ligos simptomus. Vienas iš didžiausiai pasireiškiančių simptomų yra eisenos sutrikimas. Sutrikimo dažnumą ir svarbą patvirtina ir viešai prieinama duomenų bazė, kurioje yra pateikiami sveikų ir sergančių Parkinsono liga žmonių eisenos duomenys. Duomenų bazė vadinasi “PhysioBank” [?].

Darbo tikslas yra sukurti programą, gebančią atpažinti jėgos jutikliais gautus signalus, priklausančius Parkinsono liga sergantiems subjektams. Programa bus įgyvendinta Matlab platformoje. Tokia platforma buvo pasirinkta dėl plataus įrankių kiekio, kuris yra prieinamas Matlab aplinkoje. Taip pat nurodytoje platformoje yra labai patogų ir greitą realizuoti signalų apdorojimo sistemas dėl architektūros - visi kintamieji yra matricos.

Darbe bus panaudoti duomenys, kuriuos pateikia jėgos jutikliai. Jie matuoja vertikalią žemės reakcijos jėgą (VŽRJ) (angl. Vertical Ground Reaction Force (vGRF)). Fizikoje, ir būtent biomechanikoje, VŽRJ nurodo kokia jėga žemė atsako ją slegiančiam objektui. Kaip pavyzdžiui, stovinti žmogus slegia žemę jėga, kuri lygi jo masei ir tuo pat metu, žemė slegia žmogų priešinga, lygiai tokia pačia jėga. Tokie jutikliai buvo parinkti dėl to, kad šaltiniai [?] nurodo VŽRJ kaip žmogaus kūno stabilumo indikatorius.

Parkinsono liga buvo pasirinkta dėl to, kad šiuo metu ji yra aktyviai tiriama [?], kadangi visuomenė moka labai didelius pinigus Parkinsono liga sergantiems asmenims. Per metus ši suma gali siekti 6 milijardus dolerių. Ligos rizika su amžiumi tik didėja, todėl analitikai prognozuoja, kad ateityje visuomenė mokės žymiai daugiau dėl žmonių populiacijos senėjimo.

Didžiausia ligos atpažinimo problema slypi savybių, kurios geriausiai atskirs sergantį Parkinsono liga nuo sveiko, nesergančio asmens. Darbo metu bus apžvelgtos kelios galimo savybės, kurios gali atskirti tokius subjektus. Blogiausias galimas variantas būtų nelineinė funkcija atskirti didelių dimensijų duomenys. Geriausias galimas variantas būtų mažos dimensijos duomenys (iki 3 dimensijų) ir linijškai atskiriami duomenys (kadangi linijinę funkciją realizuoti yra lengviausia). Turint didelių dimensijų duomenis, planuojamos sistemos aparatiniai reikalavimai automatiškai padidėja, kadangi būtina apdoroti labai daug duomenų. Tokią problemą galima išspręsti panaudojus dimensijų mažinimo algoritmus, kurių dažniausiai taikomi [?]: Principinių komponentų analizė (angl. Principal Component Analysis (PCA)), linijinė diskriminanto analizė (angl. Linear Discriminant Analysis (LDA)). Metodai bus aptarti vėlesnėse skyriuose. Jie taip pat vadinami dimensijų priskyrimo algoritmais. Jie yra taikomi, kuomet duomenis yra atskirti netiesiškai. Duomenims, atskirtu tiesiniu dėsniu, galima taikyti paprastą klasifikavimo algoritmą. Jeigu duomenis taip ir nepavyksta atskirti tiesiškai, tenka taikyti kompleksinį



in PD; Eur J Neuro, 2005), “Ju” - “Hausdoff et al” (RAS in PD; Eur J Neuro, 2007), “Si” - “Silvi Frenkel-Toledo et al” (Treadmill walking in PD; Mov Disorders, 2005). Toliau, “Co” nurodo kontrolinį subjektą arba nesergantį Parinsono liga subjektą, “Pt” nurodo Parkinsonu sergantį subjektą. Pirmas numeris nurodo subjekto identifikacinį numerį, po brūkšnio einantis antras numeris nurodo subjekto duomenų rinkimo seką. Aprašyme nurodyta, kad visuose duomenų rinkimų sesijose iš viso dalyvavo 93 sergantis subjektas ir 73 Parkinsono liga sergantis subjektas.

Programos veikimas bus vertinamas taiklumu ir jautrumu. Parametrai yra apskaičiuojami iš pasiklovimo matricos.

Žingsnio fazė	Statistinis parametras	Kontrolinis subjektas	Parkinsono subjektas
		Vidurkis \pm nukrypimas	Vidurkis \pm nukrypimas
Bendras žingsnis	$\mu(s)$	1.09 ± 0.09	1.12 ± 0.12
	$\sigma(s)$	0.03 ± 0.01	0.06 ± 0.04
	STC	12.44 ± 10.46	55.87 ± 29.89
Kojos pakilimas	$\mu(s)$	0.39 ± 0.04	0.37 ± 0.05
	$\sigma(s)$	0.02 ± 0.01	0.04 ± 0.02
	STC	8.5 ± 9.39	54.13 ± 26.96
Kojos prisilietimas	$\mu(s)$	0.70 ± 0.06	0.75 ± 0.09
	$\sigma(s)$	0.02 ± 0.01	0.05 ± 0.04
	STC	15.81 ± 12.07	61.27 ± 25.62

2.2 lentelė Bendro žingsnio, kojos pakilimo, kojos prisilietimo prie žemės vidurkio, vidutinio nuokrypio duomenys [?].

$$\mu = \sum_{b=1}^B x_b \hat{p}(x_b), \quad (2.1)$$

$$\sigma^2 = \sum_{b=1}^B (x_b - \mu)^2 \hat{p}(x_b). \quad (2.2)$$

Tyrimo metu buvo nustatyta, kad Parkinsono subjektų žingsnio ilgio variacija yra žymiai padidėjusi, lyginant su kontroliniais subjektais (2.2 lentelė), tačiau žingsnio vidurkis lieka beveik nepakitęs. Lentelėje taip pat pateikiami STC (angl. Signal Turn Counter) savybės vertės. Signalas $x(i)$ STC skaičiuojamas, remiantis tokiu loginiu ryšiu:

$$if \begin{cases} [x(i) - x(i-1)][x(i+1) - x(i)] < 0 \\ |x(i+1) - x(i)| \leq Th, 2 \leq i \leq I-1 \end{cases} \quad (2.3)$$

kur Th nusako ribinę vertę, I nusako signalo ilgį.

Tokia savybė taip pat turi žymių skirtumų tarp kontrolinio subjekto ir Parkinsono subjekto, tačiau ši savybė turi neigiamą aspektą - jos rezultatas priklauso ne nuo signalo kitimo pobūdžio, o nuo pasirinktos ribinės vertės (toks matematinis procesas dar vadinamas “derinimu”), o tai neatspindi Parkinsono subjektų eisenos ypatybės. Tokio matematinio “derinimo” rezultate gali įvykti, kad algoritmas bus priderintas būtent prie nagrinėjime naudojamų duomenų, tačiau jis gali būti visiškai bevertis į jį pateikus visiškai nematytus algoritmui duomenis.

Iš gauto tyrimo rezultato, galima spręsti, kad bendro žingsnio, kojos pakilimo, kojos prisilietimo prie žemės signalo ilgio dispersija (σ) gali būti panaudoti kaip savybės, norint identifikuoti Parkinsono subjektą.

Sekančiam darbe, kuris bus pateiktas apžvalgai [?], nėra naudojama vertikali žemės reakcijos jėga, skaičiuojant savybes. Sprendimas orientuojamas naudoti pigius, keturis linijinio pagreičio jutiklius. Toks sprendimas yra paprastesnis, už prieš tai apžvelgtą, kadangi jame buvo panaudoti šešiolika vertikalios žemės reakcijos jutiklių. Nagrinėjamos Parkinsono ligos požymiai buvo eisenos stingimas ir nesimetriškumas. Iš pirminės sprendimo apžvalgos iškarto galimas toks sprendimo trūkumas - tiriant galimas savybes, buvo panaudotas tik vienas Parkinsono liga sergantis subjektas ir vienas kontrolinis subjektas. Taip atliekant tyrimą, egzistuoja didelė tikimybė, kad visos gaunamos signalo eisenos savybės gali būti būdingos tik nagrinėjamam subjektui, tačiau gali visiškai negalėti kitiems subjektams ar subjektų grupei. Egzistuoja ir tokia tikimybė, kad ištirtos savybės, kurios galioja Parkinsono subjektui, gali galėti ir kontroliniam subjektui, kadangi buvo panaudotas tik vienas kontrolinis subjektas tyrimo metu.

Nepriklausomai nuo atlikti tyrimo siaurumo, darbe yra pateikta svarbios eisenos analizės išvados. Kontrolinio subjekto eisenoje nebuvo pastebėta jokių eisenos sustingimo požymių, kairės ir dešinės kojos motorinės operacijos buvo atliekamos simetriškai, išskyrus kairės kojos kelio didžiausios amplitudės pagreičio kojos susilietimo ir pakilimo su žeme metu. Abiejų kojų susilietimo su žeme laiko ilgis buvo simetriškas, kas ir patvirtina pusiausvyrą eiseną. Parkinsono subjektas parodė dešinės kojos eisenos sąstingio nesimetriškumą. Pacientas naudojo kompensacijos mechanizmą, kadangi jo eiseną automatiškai turėjo stoka. Jis peržengdavo dešinės pusės judesius, kuomet pradėdavo ėjimą. Kairės pusės judesiai buvo atliekami tik palaikant balansą. Pati eiseną yra labai lėta, o sustingimas įvyksta tik dešinėje pusėje. Iš gautų rezultatų seka, kad analizuojamas pacientas visuomet bando kompensuoti eisenos pokytį, naudodamasis viena puse. Kadangi tyrimas buvo atliktas tik su vienu pacientu, nėra galima tiksliai nusakyti, kad kompensacija vyksta tik iš dešinės pusės, tačiau ką galima tvirtai sakyti, kad naudojantis kompensacijos mechanizmu - egzistuoja padidėjusi variacijos tikimybė. Tokią hipotezę sėkmingai patvirtina ir prieš tai atlikta darbo analizė [?]. Kompensacijos nebuvimas kontrolinio subjekto atveju garantuoja, kad žingsnio ilgio variacija nebus didelė, o atvirkščiai - artės į nulį.

Tolimesnė sistema apžvalgai siūlo jau konkretų algoritmą ir techninę įrangą jo įgyvendinimui - “Characterization of gait abnormalities in Parkinson’s disease using a wireless inertial sensor system” [?]. Darbe panaudoti žingsnio, siūbavimo ir nuokrypio sukimo fizinės eisenos savybės ėjimo metu, panaudota principinė komponentų analizė (angl. Principal Component Analysis (PCA)) savybių erdvės sudarymui ir palaikymo vektorių mašina (angl. support vector machine (SVM)) klasifikavimo mechanizmui įgyvendinti. Naudojama sensorinė Micro-AHRS sensorinė platforma į kurią įeina 50 – g trijų ašių linijinio pagreičio jutiklis ir 1200 °/s trijų ašių kampinio pagreičio jutiklis. Gautas modelis sėkmingai veikia 93 % tikslumu ir taiklumu. Naudojamas svoriui jautrus (angl. cost-sensitive) apmokymas tam, kad nustatyti kokios savybės turi didesnę reikšmę formuojant savybių erdvę, bei atliekant klasifikavimą. Sprendimas buvo atliktas, panaudojus pakankamą skaičių subjektų - 23 subjektai su diagnozuota Parkinsono liga ir 16 kontrolinių subjektų, kurie neturi jokių neurologinių susirgimų ligos istorijoje. Tyrimo

	Pastovus kriterijus	Kintantis kriterijus
Jautrumas	93.3 %	88.9 %
Taiklumas	95.8 %	100.0 %
Klaidingai klasifikuotas rodiklis	4.2 %	0.0 %
Tikslumas	97.7 %	100.0 %

2.3 lentelė Klasifikavimo rezultatas, naudojant pastovų ir kintantį klaidingo klasifikavimo metodo kriterijų [?].

	Klasės atpažinimas	Klasės tikslumas
Sunkus PD eisenos sutrikimas	52.4 %	84.6 %
Lengvas PD eisenos sutrikimas	66.7 %	64.0 %
Kontrolinis subjektas	91.7 %	71.0 %

2.4 lentelė Klasifikavimo rezultatas identifikuojant lengvą ir sunkią Parkinsono ligos eigos atvejį. [?].

metu taip pat buvo nustatytas tikslas identifikuoti mažą ir didelį eisenos sutrikimo atvejį. Iš dalyvaujančių Parkinsono subjektų, 11 turi didelį eisenos sutrikimą, 12 turi mažesnį eisenos sutrikimą.

Tyrime buvo panaudoti iš viso 67 normalizuotos laiko srities savybės, kurios yra svarbios motorinei veiklai. Duomenys buvo normalizuoti dėl skirtingų savybių amplitudės srities. Normalizacija yra labai svarbi procedūra, prieš pateikiant duomenis dimensijų analizavimo metodu. Taip yra užtikrinama, kad naudojamos savybės turi vienodą įtakos faktorių, projektuojant naują dimensijų erdvę [?]. Po dimensijų mažinimo metodo pritaikymo, liko tik 11 savybių, kurių bendra variacija yra 80.20 %. Parkinsono ligos eigos sunkumui nustatyti, prie bendros savybių erdvės buvo pridėti dar papildomos keturios savybės. Sprendimas, pateikti klasifikatoriui dar papildomas keturias dimensijas nebuvo tiksliai argumentuotas. Nėra nurodyta ar savybės buvo įterptos po dimensijų mažinimo ar prieš tai. Jeigu tai buvo atlikta po dimensijų mažinimo, tai kelia abejonių naujai sukonstruota erdvė - sistema nėra gerai apibendrinta, jiniai yra "priderinta" prie esamų duomenų, todėl kelia abejonių ir pačio produkto lankstumas naujų, nematytų duomenų atžvilgiu.

Nepaisant neaiškumų dėl savybių erdvės - vektoriaus palaikymo mašinos klasifikatorius su radialiniu gauso branduoliu sugeba subendrinti savybių erdvę. Parkinsono subjekto atpažinimas yra pateiktas 2.3 lentelėje. Geriausias klasifikavimo rezultatas buvo pasiektas, naudojant kintamą klaidos vertinimo kriterijų. Nurodytas klasifikavimo metodas atpažįsta Parkinsono ligą labai gerai (100 % tikslumas). Toks rezultatas yra labai didelis iššūkis darbe naudojamai sistemai. Projektuojama sistema turi mažiausiai atsilikti nuo esamo rezultato 5 % tikslumo atžvilgiu, o idealiausiai atveju - jam prilygti.

Tolimesnis darbas apžvalgai yra “A ground reaction force artificial neural network classifier for the diagnosis of Parkinson’s disease” [?]. Duomenis analizei buvo panaudoti tokie patys, kokie bus naudojami ir šiame darbe, iš tos pačios duomenų bazės, pagrindinis nagrinėjamo darbo skirtumas yra naudojamų subjektų skaičius - analizei buvo pasirinkta 40 subjektu, turinčių Parkinsono ligą ir 40 subjektų, neturinčių neurologinių sutrikimų. Nebuvo argumentuota kodėl būtent toks duomenų kiekis buvo panaudotas analizei, kadangi duomenų bazėje yra 93 subjektai su Parkinsono ligos sutrikimais ir 73 sveiki subjektai. Iš atliktos pirminės analizės buvo pastebėta, kad Parkinsono subjektai vŽRJ signale turi mažiau galios aukštuose dažniuose, žemesnę pirmo ir antro maksimumo amplitudę ir pavėluotą pirmą maksimumą. Atlikta DFA (angl. Detrended fluctuation analysis) parodė, kad Parkinsono subjektai turi ilgesnį vidurinį ilgo laikotarpio žingsnio pakilimo nuo žemės fazės koreliacija.

Darbe buvo panaudoti 8 signalo požymiai: vidutinė galia tarp 0.5 Hz ir 1.5 Hz (dB), vidutinė galia tarp 1.5 Hz ir 20 Hz (dB), kurios pakilimo nuo žemės santykis viso žingsnio (procentais), pirmo maksimumo reikšmė (N/BW); antro maksimumo reikšmė (N/BW), kairės kojos atliktos DFA analizės skalės α eksponentė, dešinės kojos atliktos DFA analizės skalės α eksponentė [?]. Joks dimensių mažinimo algoritmas nebuvo pritaikytas. Klasifikavimui buvo parinktas dirbtinių neuronų tinklas su 8 paslėptais neuronais, apmokymas vyko naudojantis porinio mastelio gradiento atvirkštinis dauginimas (angl. scaled conjugate gradient backpropagation algorithm). Apmokymui buvo parinkta po 20 subjektų iš kiekvienos grupės, tikrinimas buvo įvykdytas panaudojus po 10 subjektų iš kiekvienos grupės, testavimas buvo atliekamas irgi panaudojus po 10 subjektų duomenis iš kiekvienos grupės. Klasifikavimo rezultatai yra pateikti 2.5 lentelėje. Bendras klasifikatoriaus tikslumas buvo pasiektas 95 % sėkmingo klasifikavimo eilės, kontroliniai subjektai buvo atpažinti su 90 % jautrumu (vienas iš dešimt subjektų buvo atpažinti klaidingai), Parkinsono liga sergantys subjektai buvo atpažinti su 100 % jautrumu, visi subjektai buvo atpažinti teisingai.

18

	C1	C2
Tikslumas	0.950	0.950
Taiklumas	0.900	1.000

2.5 lentelė Klasifikavimo rezultatai, panaudojus dirbtinių neuronų klasifikatorių [?].

Žingsnio ypatybė	Kontrolinis subjektas	Parkinsono subjektas
	Vidurkis \pm nukrypimas	Vidurkis \pm nukrypimas
Žingsnio ilgis (m)	0.550 ± 0.080	0.420 ± 0.200
Eisenos greitis (m/s)	0.920 ± 0.160	0.650 ± 0.330

2.6 lentelė Pagrindiniai eisenos parametrai remiantis atlikta analize [?]

mo dirbtinių neuronų klasifikatoriaus. Metodas neleidžia įžvelgti kaip būtent jis apibendrina naudojamas savybės savybių erdvėje. Taip pat nebuvo pavaizduota kaip esami duomenys pasiiskirstę savybių erdvėje. Kadangi buvo naudojamos 8 savybės (kas apibrėžia ir savybių erdvės dimensijų skaičių), galima buvo panaudoti dimensijų mažinimo algoritmą ir pateikti esamus rezultatus grafiniu pavidalu. Taip pat nurodytos savybės yra skirtingos skalės. Kiekviena iš išvardintų savybių gali turėti savo amplitudės diapazoną. Labai svarbu yra normalizuoti naudojamas signalo savybės taip užtikrinant kiekvienos savybės lygų įvertį klasifikatoriui apibendrinant duomenis. Šiuo metu nėra įmanoma nustatyti kiek viena savybė turi įtakos bendram algoritmo atpažinimo veikimo efektyvumui.

Ketvirtas ir paskutinis darbas apžvalgai yra gilesnis statistinis eisenos analizės darbas, “Statistical Analysis of Parkinson Disease Gait Classification using Artificial Neural Network” [?]. Darbe buvo išnagrinėtos trys grupės eisenos savybių: pagrindinės, kinetinės ir kinematinės. Pagrindinės savybės įvardijamos: žingsnio laikas, žingsnių skaičius per minutę, žingsnio ilgis, eisenos greitis. Kinetinės savybės: maksimalus pėdos kontaktas su žeme pagal vertikalią; maksimalus pėdos kontaktas su žeme pėdos pakilimo metu pagal vertikalią; minimali jėga tarp dviejų kojos prisilietimo prie žemės maksimumų pagal vertikalią; maksimali pėdos jėga pagal horizontalę, kuomet pėda prisiliečia prie žemės; maksimali pėdos jėga pagal horizontalę, kuomet koja pakyla nuo žemės. Kinematinės savybės: kulkšnies kampas, kuomet pėda prisiliečia prie žemės; kelio kampas, kuomet pėda prisiliečia prie žemės; klubo kampas kuomet koja prisiliečia prie žemės; maksimali kulkšnies, kelio ir klubo kampas. Duomenys išskirti tokiomis savybėmis buvo surinkti pasitelkus slėgiui jautrų paviršių, kurio diskretizavimo dažnis buvo 200 Hz ir 37 markerius, kurie buvo pritvirtinti prie subjektų kojos ir stebėti infraraudonųjų spindulių kamerų. Visi duomenys buvo apdorojami *Vicon*[®] programinio paketo. Norint išvengti slėgio jautriu paviršiumi surenkamų duomenų priklausomybė nuo subjektų svorio, duomenys buvo normalizuojami pagal subjektų svorį:

Savybių grupė	Klasifikavimo taiklumas
Pagrindiniai	81.25 %
Kinetiniai	81.25 %
Kinematiniai	84.38 %
Pagrindiniai ir kinetiniai	87.50 %
Pagrindiniai ir kinematiniai	87.50 %
Kinetiniai ir kinematiniai	84.38 %
Pagrindiniai, kinetiniai ir kinematiniai	87.25 %
Keturių didžiausio skirtumo savybės	95.63 %

2.7 lentelė Dirbtinių neuronų tinklo klasifikavimo rezultatai, panaudojus skirtingas savybių grupes [?]

$$\text{Normalizuotas GRF}(\%) = \frac{GRN(N)}{Svoris(N)} * 100\% \quad (2.4)$$

Iš viso duomenų surinkimo metu dalyvavo 12 Parkinsono liga sergančių subjektų ir 20 kontrolinių subjektų, kurie neturėjo savo ligos istorijoje jokių neurologinių susirgimų.

Klasifikavimo uždaviniui spręsti buvo panaudotas dirbtinis neuronų tinklas su daugiasluoksniu perceptronu. Duomenis algoritmui buvo pateikiami padalinus visus turimus duomenis į keturis rinkinius.

Darbo tikslas buvo pirmiausiai analitiškai išanalizuoti visus gaunamus duomenis ir padaryti hipotezę, kokios eisenos savybės geriausiai identifikuotų subjektų grupes. Vėliau, pagal klasifikatoriaus rezultatus parinkti geriausius požymius, pagal kuriuos vyks subjektų klasifikavimas. Pagrindiniai eisenos parametrai, kurių parametrai labiausiai skyrėsi tarp subjektų, yra pateikti 2.6 lentelėje. Buvo pastebėta, kad kontrolinių subjektų žingsnio ilgis yra ilgesnis už Parkinsono subjektų. Taip pat iš pagrindinių savybių buvo išskirtas ėjimo greitis. Parkinsono subjektų eisenos greitis buvo pažymėtas kaip lėtesnis, negu kontrolinių subjektų. Jėgos jutiklių parametų palyginimas parodė, kad daugiausiai informacijos parodo minimumas, kuris yra tarp pėdos pirmo prisilietimo prie žemės maksimumo ir pėdos pakilimo nuo žemės. Iš kinematinų savybių, didžiausias skirtumas tarp subjektų buvo nurodytas kaip klubo maksimalus kampas: 19.904 ± 8.623 laipsnių kontrolinių subjektų atveju ir 11.271 ± 6.157 laipsnių Parkinsono subjektų atveju.

Atlikus analitinį palyginimą, toliau buvo atliktas savybių tikrinimas klasifikatoriaus darbo rezultatais. Tikrinimo duomenys yra parodyti 2.7 lentelėje. Pagrindinių ir kinetinių savybių grupės surinko po vienodą klasifikavimo rezultatą. Tai leidžia daryti išvadą, kad abi šios savybių grupės rodo panašius eisenos skirtumus. Kinematinės savybės šiuo atveju pralenkia kitas dvi savybių grupes. Sujungus pagrindines ir kinetines savybių grupes buvo pasiektas kiek di-



desnis klasifikavimo rezultatas, tačiau lygiai tokį patį pateikė ir pagrindinių su kinematinėmis savybėmis sujungimas. Kiti galimi savybių grupių sujungimai nepateikė geresnių klasifikavimo rezultatų. Geriausią rezultatą pateikė keturių savybių grupė, kuri paaiškėjo po atliktos statistinės analizės: žingsnio ilgis, eisenos greitis, klubo maksimumo kampas ir minimumas tarp dviejų pėdos prisilietimo prie žemės maksimumų.

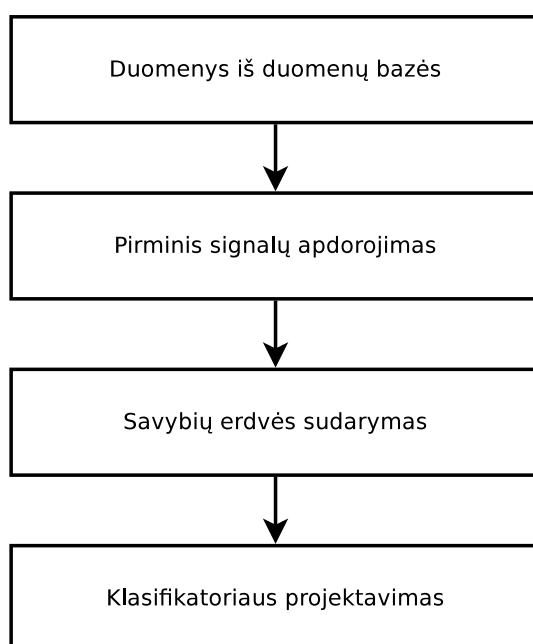
Iš aptarto darbo, šiame darbe bus panaudotos savybės savybių erdvei sudaryti bei taip pat bus pritaikytas dirbtinių neuronų tinklas kaip klasifikatoriaus mechanizmas. Kitų duomenų savybių erdvei sudaryti panaudoti nėra galima, kadangi turimas priėjimas tik prie jėgos jutiklių duomenų.

3. Signalų analizės programos kūrimas

Šiame skyriuje bus apžvelgtos kylančios problemos kuriant signalų analizės programą, pateikti galimi jų sprendimo būdai. Skyrius prasidės bendros programos struktūrinės schemos sudarymo poskyriu. Jame bus pateiktas bendra programos veikimo struktūra, kiek funkcinių blokų sudarys programą, kokią funkciją atliks nurodyti blokai. Sekančiam poskyryje bus aptarti galimi pirminio signalo apdorojimo etapai. Tai yra vienas iš svarbiausių programos žingsnių, kadangi nuo pirminio signalo apdorojimo kokybės priklauso ir visos programos rezultatas. Požymio išskyrimo poskyryje seks signalo savybių nagrinėjimas, bus atlikta signalo savybių analizė. Nagrinėjama savybių erdvė bus panaudota klasifikavimo metu, todėl būtina rasti tokias savybes, kurios skiriasi tarp kontrolinio subjekto ir Parkinsono subjekto. Požymio klasifikavimo programos kūrimo metu bus aptarti galimi naudoti klasifikatoriai, kiekvienam iš jų bus pateikti vienodi duomenis klasifikavimo užduočiai atlikti ir nuspręsta koks klasifikatorius geriausiai tinka nagrinėjamų savybių erdvei. Galiausiai, duomenų analizės programos metu bus apžvelgti išnagrinėti metodai, pateiktas geriausiai užduotį atliekantis pirminio signalo apdorojimo metodas, dimensijų mažinimo algoritmas, bei klasifikavimo mechanizmas.

3.1. Bendros programos struktūrinės schemos sudarymas

Priklausomai nuo taikomos metodikos, programos struktūrinės schemos skiriasi, kadangi vienas sprendimas reikalauja vieno tipo duomenų struktūros, kitas - kitos struktūros. Programos projektavimo metu buvo išbandyti keli programos variantai ir suprastinus veikimo schemas, kiekvienas iš programos variantų veikė pagal bendrą schemą, kurią galima apibrėžti 3.1 pav. Reikalaujama, kad kiekvienas iš algoritmų blokų būtų nepriklausomas nuo žemiau ar aukš-



3.1 pav. Pirminė programos schema.



čiau einančio bloko, kas reiškia, kad “Pirminio signalų apdorojimo” bloko algoritmo pakeitimas neturi turėti jokios įtakos po jo einančiam “Savybių erdvės sudarymas” blokui.

Kiekvieno bloko užduotis:

1. “Duomenys iš duomenų bazės” bloko užduotis yra nuskaityti iš naudojamos “PhysioBank” duomenų bazės duomenis ir juos įkrauti į Matlab aplinką. Kadangi duomenys pateikiami tekstiniu pavidalu, Matlab aplinkoje yra labai patogi funkcija tokiems duomenims nuskaityti - *dlmread*.
2. “Pirminis signalų apdorojimas” bloko užduotis yra pradinis signalų filtravimas, nuolatinės komponentės pašalinimas. Taip pat, į šį bloką įeina ir atskirų signalų išskyrimas iš bendrai gaunamo signalo: signalo išskyrimas kuomet koja turi susilietimą su žeme, signalo išskyrimas, kuomet koja praranda susilietimą su žeme. Į bloką taip pat įeina ir slankiojančio lango algoritmo taikymas.
3. “Savybių erdvės sudarymas” bloko paskirtis yra pateikti pasirinktam klasifikatoriui išskirtas signalų savybes po pirminio signalo apdorojimo. Tokiomis savybės gali būti signalų dažninės komponentės (Furjė koeficientai), koreliacijos koeficientai tarp dešinės ir kairės kojos, savi-koreliacijos koeficientai dešinės ar kairės kojos. Detaliau nagrinėjamas savybes bus aptartos tolimesniuose poskyriuose. Į bloką taip pat įeina ir darbas su dimensijomis - dimensijų skaičiaus mažinimas ar naujos erdvės paieška, kurioje nagrinėjami duomenys yra geriau tiesiškai atskiriami.
4. “Klasifikatoriaus projektavimas” bloko paskirtis yra klasifikatoriaus apmokymas ir jo testavimas. Klasifikatoriaus pasirinkimas yra labai svarbus klausimas nagrinėjamam darbe. Jis turi būti parinktas argumentuotai, pateikiant detalius klasifikavimo rezultatus, remiantis rezultato grafiku. Nepasitikėjimo matrica, bei iš jos apskaičiuotais įverčiais: taiklumu bei tikslumu. Dar svarbus klasifikatoriaus aspektas yra savybių erdvės bendro dėsnio radimas arba jo aproksimacija. Šį faktą galima apibrėžti iš klasifikavimo grafiko. Jeigu klasifikavimo rezultatas yra nestabilus, ilgą laiką neturintis pastovaus rezultato, vadinasi, galima teigti, kad klasifikatorius blogai atliko erdvės dėsnio aproksimavimą, jis nėra tinkamas nagrinėjamai savybių erdvei.

Tolimesniuose poskyriuose bus aptarti įgyvendinti blokinės struktūros elementai.

3.2. Pirminio signalų apdorojimo programos kūrimas

Šiame skyriuje bus aptartas pirminio signalų apdorojimo programos kūrimas. Aptartas duomenų nuskaitymas iš duomenų bazės bylos tekstiniame pavidale, galimas signalo išskaidymas slankiojančio lango metodu arba signalo formos nuskaitymas, priklausomai nuo žinomo fizinio poveikio, kuriuo metu signalas buvo gautas.

Programos kūrimo patogumui, buvo parinkta direktorių architektūra:



3.1 programinis kodas. Duomenų nuskaitymo funkcija iš tekstinės duomenų bylos.

```
1 function data = read_data( file_name )
2 %READ_DATA Function reads the data from file
3 %
4 %   Input:
5 %       - file_name the name of the file
6 %
7 %   Output:
8 %       - data the Left foot data
9
10 database_directory = '../database/';
11
12 % Check if file exists
13 if ~exist( [ database_directory file_name ], 'file' )
14     % Problem arise
15     error( [ 'File_' file_name '_does_not_exist' ] );
16 end
17
18 data_raw = dlmread( [ database_directory file_name ] );
19
20 % We'r interested only in left foot data
21 data = data_raw(:,19);
22
23
24 end
```

- Tėvinė direktorija
 - <programos versija, nurodyta datos formatu>
 - database
 - cache

Kadangi programos kūrimo metu yra svarbu saugoti ankstesnes programos versijas, buvo parinktas kasdienis programos versijos saugojimas: kiekvienos pradžioje darbas buvo pradėdamas su ankstesnės dienos kopija. Taip buvo išsaugotas kiekvienos darbo dienos programos versija ir taip galima peržiūrėti kokių analitinių kelių buvo eita prie dabartinės programos versijos.

Pirmoji funkcija, priklausanti pirminio signalo apdorojimo programai yra *read_data*. Funkcijos kodas pateikiamas 3.1 pav.

Funkcijai užtenka nurodyti tik norimos nuskaityti bylos pavadinimą, kaip pavyzdžiui “SiPt30_01.txt” ir funkcija gražins kairės kojos duomenis. Duomenų analizėje naudojami tik vienos kojos duomenis, kadangi dešinės ir kairės kojos duomenys yra stipriai koreliuoti [?], todėl nėra prasmės naudoti dviejų kojų duomenų. Pasirinkus tokį sprendimą taip pat yra sumažinamas skaičiavimų skaičius programoje. Duomenų bazės direktorija nurodyti kintamojo vietoje yra taip pat svarbus aspektas, kadangi pakitus duomenų bazės vietai, užteks tik pakeisti vieną kintamąjį,



3.2 programinis kodas. Slankiojančio lango algoritmo pritaikymas.

```
1 function output = split_data( input, window_length, window_overlay )
2 %SPLIT_DATA Function to split data into windows
3
4 % Buffer to store the signal information
5 buffer = [];
6 output = [];
7
8 for i=1:length(input)
9     % if buffer reached the length of the window
10    if ( length(buffer) >= window_length )
11        % store the buffer to output signal
12        output = [ output; buffer ]; %%ok<AGROW>
13        % Do the overlay step
14        if ( window_overlay ~= 0 )
15            buffer = buffer(window_overlay:end);
16        end
17    else
18        % otherwise, populate the buffer with new entry
19        buffer = [ buffer input(i) ]; %%ok<AGROW>
20    end
21 end
22
23 end
```

o ne visą kodą, nors kodas ir nėra ilgas. Matlab funkcija *dlmread* gražina duomenis masyvo pavidalu, kur stulpelis nurodo įvade aptartus duomenis, o eilutė nurodo duomenų vertes tam tikru laiko momentu.

Sekantis žingsnis, po duomenų nuskaitymo, yra jų pirminis apdorojimas. Nagrinėjimas bus pradėtas nuo slankiojančio lango metodo. Programos kodas pateikiamas 3.2 pav. Funkcijos įvestyje pateikiamas nagrinėjamas signalas, slankiojamo lango ilgis ir slankiojamo lango perdanga. Slankiojamo lango perdanga nurodo reikšmių arba laiko verčių kiekį, kurį algoritmas pašalina iš spartinančiosios atminties, laukdamas naujų verčių langui užpildyti. Pavyzdžiui, jeigu lango ilgis yra 4 signalo verčių, o perdanga - 2 signalo vertės, vadinasi, kai algoritmas užpildys langą 4 signalo vertėmis, esamą langą jis perduos į išėjimo spartinančiąją atmintį, pasakutines dvi vertes ištrins iš spartinančiosios atminties ir iš naujo lauks naujų dviejų reikšmių langui pilnai užpildyti.

Funkcija yra tiek lanksti, kad nėra svarbu kokio tipo duomenys yra pateikiami - ar tai konkretaus signalo vertės ar iš ankščiau pritaikyto slankiojančio lango algoritmo išskirtos signalo savybės, kuriuos buvo panaudotos formuojant naują signalą. Tokia funkcijos savybė bus labai naudinga tolimesniame darbe.

Sekanti programos dalis atlieka paprastą signalo filtravimą su dvejais *Butterworth* skaitmeniniais filtrais. Pirmasis, aukštų dažnių filtras, skirtas pašalinti signalo nuolatinei komponentei. Antras, žemų dažnių filtras, skirtas pašalinti aukšto dažnio triukšmą, kuris neneša visiškai jokios



3.3 programinis kodas. Signalų filtravimas dviem Butterworth filtrais.

```
1 [B,A] = butter(9, 1/50, 'high');
2 [BB,AA] = butter(9, 40/50, 'low');
3 output = filter(BB, AA, filter(B, A, input));
```

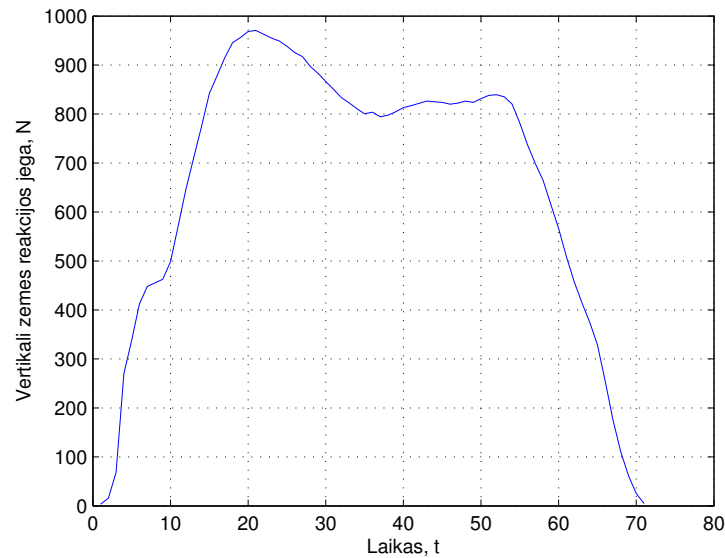
3.4 programinis kodas. Kontakto su žeme signalo išskyrimo programos kodo fragmentas.

```
1 for i=1:size( Pt_t, 2 )
2     fprintf(' ');
3     signal = Pt_t{i};
4     buffer = [];
5     k = 0;
6     for j=1:size(signal, 1)
7         if ( signal(j) ~= 0 )
8             buffer = [ buffer signal(j) ]; %%ok<AGROW>
9         else
10            if ~isempty(buffer) && length(buffer) > 10 && length(buffer) < 200
11                % Store the buffer
12                data.Pt{ length(data.Pt)+1 } = buffer;
13                % Clear the buffer
14                if ( j - k > 50 && j - k < 200 )
15                    data.Pt2( length(data.Pt2)+1 ) = j - k;
16                end
17                k = j; % Start of stance
18                buffer = [];
19            else
20                buffer = [];
21            end
22        end
23    end
24 end
```

naudingos informacijos. Filtras įgyvendinamas labai paprastai. Kodo pavyzdys pateikiamas 3.3 pav. Abiejų filtrų eilė yra 9-ta, žemų dažnių filtro ribinis dažnis parinktas 40 Hz. Duomenys diskretizuojami 100 Hz dažniu, vadinasi didžiausias galimas signalo dažnis yra 50 Hz. Didžiausias žmogaus generuojamas dažnis ėjimo metu, remiantis šaltiniu, yra 20 Hz. Užtikrintumui buvo parinktas 40 Hz dažnis. Nuolatinė dedamoji pašalinama su aukšto dažnio filtru, kurio ribinis dažnis yra 1 Hz. Nuolatinė dedamoji neneša jokios informacijos apie eisena, kadangi ji tiktais nurodo naudojamų jutiklių jautrumą.

Toliau seka, priklausomai nuo pasirinkto pirminio signalų apdorojimo būdo, signalo išskyrimas pagal fizinę veiklą. Dominančios signalo būsenos yra, kai subjekto koja turi kontaktą su žeme ir nurodytas subjektas neturi kontakto su žeme. Kontakto su žeme signalo išskyrimui programos kodas yra pateikiamas 3.4 pav.

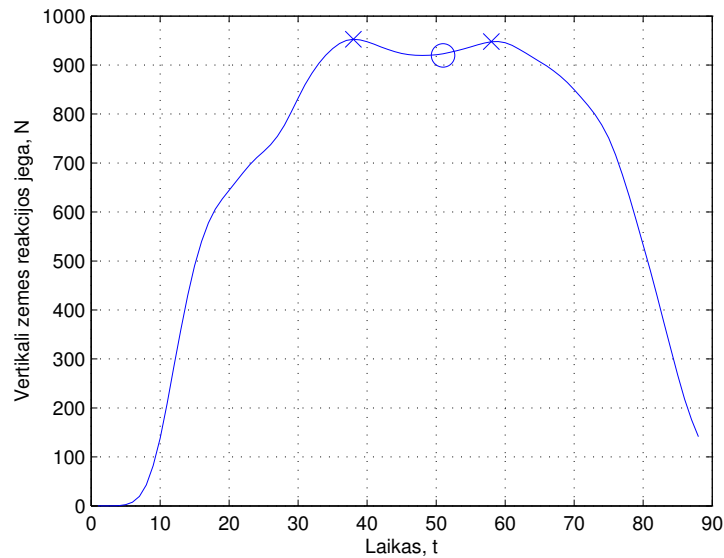
Pirmiausiai, “Pt_t” kintamojo struktūroje yra saugoma kairės kojos Parkinsono liga sergančių subjektų duomenys. Kiekvienas signalas yra priskiriamas prie *signal* kintamojo, su kuriuo toliau yra tęsiamas apdorojimo procesas. Jeigu signalas nėra lygus nuliui, tuomet jis



3.2 pav. Susilietimo su žeme signalas.

įdedamas į laikinąją atmintį. Taip signalas yra tikrinamas iki tol, kol signalas tampa lygus nuliui ir tęsiamas tolimesnis apdorojimas. Apdorojimas susideda iš signalo ilgio patikros. Jeigu signalas yra trumpesnis už 10 verčių, arba turint omenyje, kad signalas buvo diskretizuojamas 100 Hz, tai 0.1 s, vadinasi, signalas yra tiesiog aukšto dažnio triukšmas arba blogas pavyzdys ir signalas yra atmetamas. Jeigu signalas yra ilgesnis už 200 verčių (2 s), vadinasi, duomenų rinkimo metu įvyko klaida ir koja per tokį laiką pakelta nebuvo. Tokia klaida gali būti sukelta, kuomet subjektas ne eina, o stovi ant dviejų kojų arba tik ant kairės kojos. Jeigu visi kriterijai patenkinami, signalas yra priimamas ir kraunamas į laikinąją atmintį, pavadinimu „*data.Pt*“. Signalas, kuriuo metu koja neličia žemės, yra randamas kartu su ankščiau išnagrinėtu metu. Algoritmas patikrina kiek praėjo laiko (arba kiek verčių yra priskaičiuota) nuo paskutinio užskaityto kojos ant žemės signalo ir įrašo tą signalą į laikiną atmintį, jeigu po praeito signalo nepraėjo mažiau negu 50 verčių arba 0.5 s ir nedaugiau nei 200 verčių arba 2 s. Kojos signalas yra pateiktas 3.2 pav.

Dar vienas pirminis signalo apdorojimas, kurį teko panaudoti tiriant galima bendrą erdvę - kiekvieno kojos susilietimo su žeme signalo dviejų globalių maksimumų ir vieno lokalaus minimumo paieška [?]. Algoritmo rezultatas yra pateiktas 3.3 pav. Kryžiais pažymėtos globalaus maksimumo vietos, apskritimu pažymėta lokalaus minimumo signalo vieta. Išanalizavus daugumos subjektų žingsnio signalus, buvo padaryta išvada, kad kiekviename žmogaus žingsnyje egzistuoja du maksimumai. Vienas maksimumas randamas, kai subjektas yra visiškai atsirėmęs galine pėdos dalimi į žemę, antras maksimumas randamas, kai subjektas visiškai atsiremia priekinę pėdos dalimi. Tarp šių dviejų maksimumų yra pereinamasis laikotarpis, kuris yra signalo lokalus minimumas. Priežastis, dėl kurios algoritmas buvo įgyvendintas, yra vienas darbas, kuriame buvo pasiūlyta naudotis būtent tokiais savybėmis atpažinti Parkinsono liga sergančius



3.3 pav. Dviejų globalių maksimumų ir vieno lokalaus minimumo radimas signalė.

3.5 programinis kodas. Dviejų globalių maksimumų ir vieno lokalaus minimumo radimo algoritmo fragmentas.

```

1  for i=1:size(signal_window, 2)
2      buffer = [ buffer signal_window(i) ]; %%ok<AGROW>
3      if length(buffer) > 2
4          delta = buffer(i) - buffer(i-1);
5          if delta < d_pr
6              % Falls
7              buffer(i) = buffer(i-1);
8          end
9
10         if buffer(i) == buffer(i-1) && right_index == 0 && right_max == 0 && buffer(i) > min_valu
11             right_index = i-1;
12             % Return the maximum in the buffer
13             right_max = max(buffer);
14         end
15     end
16 end

```

subjektus nuo nesergančių subjektų. Pačio algoritmo kodo dalis yra pateikta 3.5 pav.

Programos kodas krauna gaunamą signalą į laikinąją atmintį ir tikrina ar signalas pakito per užduotą dydį δ . Delta nurodo kiek signalas turi pakisti, kad algoritmas nuspręstų, kad signalas pradėjo mažėti. Tokio tikrinimo priežastis yra ta, kad prieš globalų maksimumą taip pat yra sritis, kurioje signalas kilimas sulėtėja, o kai kuriuose pavyzdžiuose buvo pastebėta, kad signalas net pradeda mažėti. Dėl šios priežasties buvo įvesta pokyčio tikrinimo sąlyga.

Pirminio signalų apdorojimo programa baigiasi nurodytais metodais. Sekantis žingsnis yra apdorotų duomenų perdavimas tolimesnei programos pakopai - požymių išskyrimui.

3.3. Požymių išskyrimo programos kūrimas

Požymių išskyrimas priklauso nuo pirminio signalo apdorojimo mechanizmo. Jeigu ankstesniame bloke pakeisime ribinį filtro dažnį ir kaip nagrinėjamą požymį bus parinktas koreliacijos koeficientai ar dažninės komponentės - požymio vektorius pakis, todėl prieš kiekvieną požymių analizavimo ciklą bus pateikta ir pirminiam signalų apdorojimo bloke naudojami metodai. Taip pat, kadangi požymių išskyrimo programos kūrimas taip pat sprendžia ir dimensių klausimą - skyriaus pradžioje bus trumpai apžvelgti populiariausi šiame darbe panaudoti dimensių mažinimo metodai.

3.3.1. Dimensių mažinimas

Dimensių mažinimas suteikia dideles galimybes supaprastinti klasifikavimo uždavinį. Kuomet duomenys savybių erdvėje nėra liniškai atskirti, yra sunku nustatyti ar klasifikatorius ras apibendrinantį duomenų priskyrimo funkciją. Norint užtikrinti, kad klasifikatorius veiks teisingai - yra atliekamas dimensių priskyrimas. Metodus taip pat taikomas, kuomet norima sumažinti dimensių skaičių. Problema iškyla, kuomet norima įgyvendinti sukurto sprendimą įterptinėje sistemoje, kurioje negalima pasiekti didelių aparatinių resursų. Turint didelių dimensių duomenis taip pat kyla ir pačios sistemos kaina, dėl tų pačių aparatinių resursų reikavimų. Norint išvengti tokių problemų - taikomi dimensių mažinimo algoritmai. Populiariausi dimensių mažinimo algoritmai yra:

- Linijinė diskriminanto analizė (angl. Linear Discriminant Analysis (LDA));
- Principinė komponentių analizė (angl. Principal Component Analysis (PCA)).

Taip pat, kiekvienas iš paminėtų būdų taip pat gali būti papildytas branduolio (angl. Kernel) funkcija. Metodų apibūdinimas bus pradėtas nuo paprasčiausios iš dviejų - PCA.

Principinė komponentių analizė [?] yra statmena ir paprasta transformacija, kuri yra daug kur naudojama dėl paprastos matematinės pusės ir lengvo įgyvendinimo. Egzistuoja skirtingi PCA metodo įgyvendinimo algoritmai. Vienas iš jų yra vienetinės reikšmės skaidymas (angl. Singular Value Decomposition (SVD)). Literatūroje PCA ir SVD dažniausiai minimi kaip sinonimai. SVD yra greitas, tačiau didelių atminties resursų reikalaujantis metodas. Kuomet dideli atminties resursai nėra prieinami (įterptinėje sistemoje), reikia naudoti kitą įgyvendinimo variantą - naudoti tikrinių vektorių skaidymą. Toks sprendimas užima didesnius laiko resursus, lyginant su SVD, tačiau jam reikia mažesnių atminties resursų ir jis tinka nagrinėti didelių dimensių duomenis. Toks sprendimas bus taikomas ir šiame darbe. PCA įgyvendinimas naudojant SVD ir tikrinių vektorių skaidymą yra pateiktas priede. Toliau bus išnagrinėtas PCA iš matematinės pusės.

Matematiškai, PCA apibūdinamas kaip statmena linijinė transformacija, kuri transformuoja duomenis į naują koordinačių sistemą, kurioje didžiausią variaciją projektuojama į pirmą ašį,

antroji didžiausia variacija (statmena pirmajai) projektuojama į antrą ašį. Procesas tęsiasi tol, kol yra pasiekiamas norimas dimensijų skaičius.

Apibrėžta duomenų matrica \mathbf{X}^T , su empiriniu vidurkiu, kuris lygus nuliui (empirinis vidurkis reiškia vidurkio apskaičiavimą ir jo atėmimą iš duomenų), kur kiekviena eilutė n atspindi duomenų rinkinį, o stulpelis m - turimas duomenų dimensijas. Matricos \mathbf{X} vienetinės reikšmės skaidymas išreiškiamas:

$$\mathbf{X} = \mathbf{W}\Sigma\mathbf{V}^T, \quad (3.1)$$

kur $m * n$ matrica \mathbf{W} yra matricos $\mathbf{X}\mathbf{X}^T$ tikrinių vektorių matrica, Σ matrica yra $m * n$ stačiakampio įstrižainės matrica su realiais skaičiais įstrižainėje ir $n * n$ tikrinių vektorių $\mathbf{X}^T\mathbf{X}$ matrica \mathbf{V} . PCA transformacija, kuri sukonstruoja naujas dimensijas, yra apibrėžiama:

$$\mathbf{Y}^T = \mathbf{X}^T\mathbf{W} = \mathbf{V}\Sigma^T\mathbf{W}^T\mathbf{W} = \mathbf{V}\Sigma^T. \quad (3.2)$$

Kadangi \mathbf{W} yra statmena matrica, kiekviena \mathbf{Y}^T eilutė yra matricos \mathbf{X}^T eilutės sukimas. Pirmasis \mathbf{Y}^T stulpelis yra pirmosios komponentės rezultatas, antras stulpelis yra antrosios komponentės rezultatas. Kiek stulpelių \mathbf{Y}^T turi, tiek ir rezultatų (dimensijų) yra po transformacijos.

Linijinė diskriminanto analizė [?] yra dimensijų mažinimo metodas (kuris kartu yra naudojamas kaip ir klasifikatorius), yra vienas iš metodų, kuris neša kartu ir duomenų žymėjimo informaciją. Tai reiškia, kad mažinant dimensijų skaičių, metodas turi žinoti kokie duomenys priklauso kokiai duomenų klasei. Pagrindinis LDA tikslas yra didinti sekantį kriterijų:

$$\mathfrak{J}(\mathbf{w}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}, \quad (3.3)$$

kur S_B yra “išorinės klasės scatter matrica”, S_W yra “vidinė klasės scatter matrica”. Scatter matricių apibrėžimas yra:

$$S_B = \sum_c (\mu_c - \bar{x})(\mu_c - \bar{x})^T \quad (3.4)$$

$$S_W = \sum_c \sum_{i \in c} (x_i - \mu_c)(x_i - \mu_c)^T \quad (3.5)$$

Branduolio metodas yra labai lengvai paaiškinamas pavyzdžiu [?]. Tarkim, egzistuoja tam tikri empiriniai duomenys:

$$(x_1, y_1), \dots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}. \quad (3.6)$$

Čia, \mathcal{X} yra netuščia sritis, iš kurios imami x_i duomenys. Tikslu arba žymekliu vadinami yra $y_i \in \mathcal{Y}$ (sistemos atsakas), $i, j \in [n]$ žymimas eilės, identifikavimo numeris, kur $n := \{1, \dots, n\}$.

Reikia pastebėti, kad sričiai \mathcal{X} nebuvo suteikta jokių apribojimų. Norint išspręsti apmokymo problemą reikia papildomos struktūros. Mašiniame apmokyme siekiama apibendrinti neturimus duomenis. Binarinio klasifikavimo atveju, turint naujus duomenis $x \in \mathcal{X}$, norima nuspėti jų žymeklį $y \in \{\pm 1\}$. Paprastai kalbant, norima pasirinkti tokį y , kuris labiausiai atspindėtų apmokymo metu naudotas (x, y) poras. Tam įgyvendinti reikalingi kuo panašesnis duomenys į \mathcal{X} ir \mathcal{Y} . Pastarąjį palengvina tai, kad spėjimas gali būti arba identiškas arba atvirkščias. Formaliai yra reikalinga tokia funkcija:

$$k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}, \quad (x, x') \rightarrow k(x, x'), \quad (3.7)$$

kuri tenkinta tokias sąlygą visiems $x, x' \in \mathcal{X}$:

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle, \quad (3.8)$$

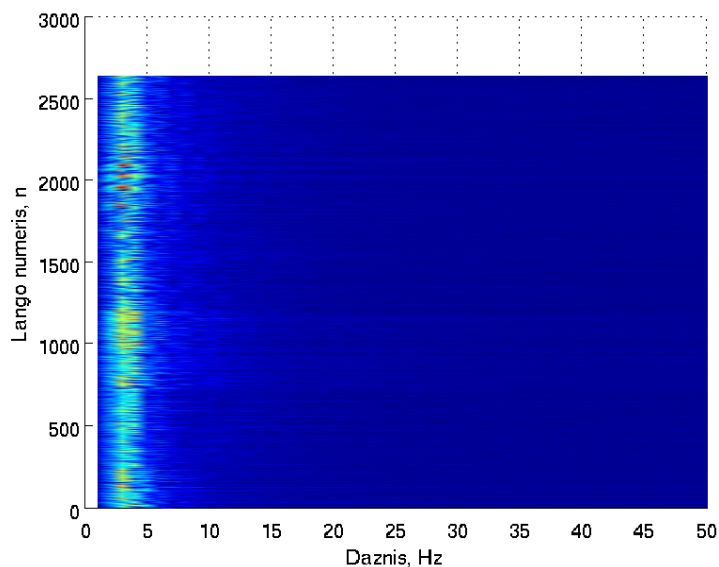
kur Φ projektuota taškus į Hilberto plokštumą \mathcal{H} , kai kada vadinama savybių erdvė. Vieno dumo matmuo k dažniausiai vadinamas branduoliu, o Φ vadinama savybių projekcija.

3.3.2. Savybių tyrimas

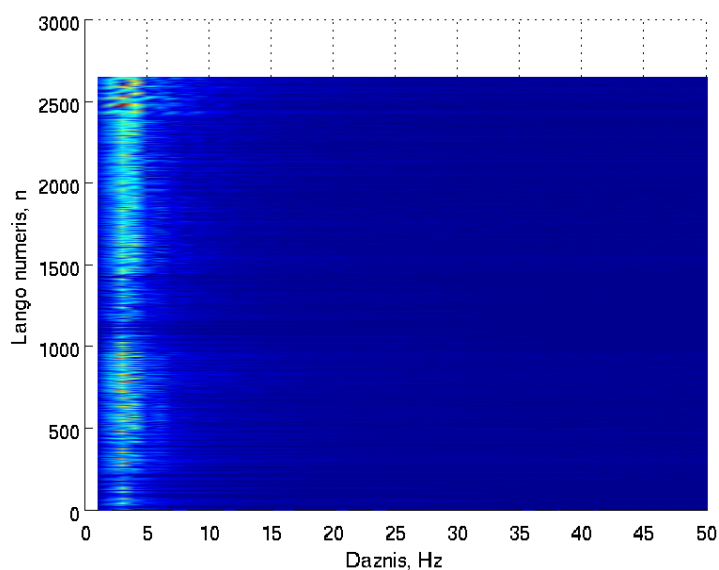
Šiame skyriuje bus išnagrinėtos galimos signalų savybės, pagal kurias galima atskirti Parkinsono liga sergantį subjektą nuo sveiko subjekto. Kaip buvo minėta anksčiau, ne visos savybės gali būti išskirtos, naudojantis vienu pirminio signalo apdorojimo mechanizmu, todėl kiekvienos savybės nagrinėjimo pradžioje bus paminėta ir pirminio signalo apdorojimo bloko sudėtis.

Galimų savybių analizavimas gali būti pradėtas nuo signalo dažninių komponentių (Furjė transformacijos). Kaip nurodo šaltiniai, Parkinsono liga sergančių subjektų žingsniai turi statėsnius šlaitus, kas iš signalų apdorojimo srities reiškia, kad signalas turi turėti aukštas dažnines komponentes. Šiai savybei išskirti pirminiam signalo apdorojimo bloke bus panaudotas slankiojančio lango metodas. Taip kiekvienas signalas turės fiksuotą ilgį ir taip bus galima lyginti signalus tarpusavyje.

Toliau bus išanalizuoti Parkinsono liga sergančių subjektų ir kontrolinių subjektų žingsnių signalų dažninės komponentės. Kontrolinių subjektų kairės kojos žingsnių dažninės komponentės yra pateiktos 3.4 pav. Parkinsono liga sergančių subjektų kairės kojos žingsnių dažninės komponentės yra pateiktos 3.5 pav. Kaip matoma iš duotų komponentių grafikų - tiek kontrolinių subjektų, tiek Parkinsono liga sergančių subjektų pagrindinės dažninės komponentės išsidėsto iki 5 Hz ruože. Ties 0 Hz dažninių komponentių nėra, kadangi jos buvo pašalintos filtro pagalba. Už 10 Hz ribos, komponentės neneša visiškai jokios informacijos. Iš to galima padaryti išvadą, kad tiek Parkinsono liga sergančių subjektų, tiek kontrolinių subjektų eisenos yra visiškai vienodos dažnių srityje ir vien remiantis šita informacija nėra galima nustatyti ar



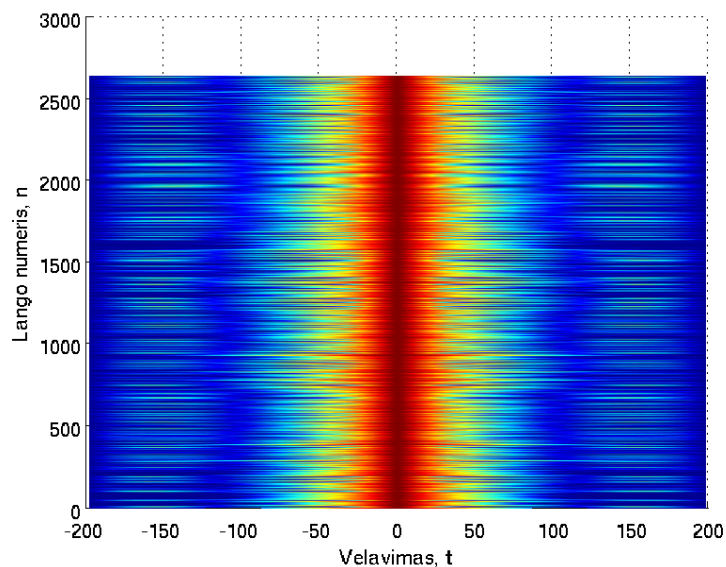
3.4 pav. Kontrolinių subjektų kairės kojos žingsnių signalų dažninės komponentės



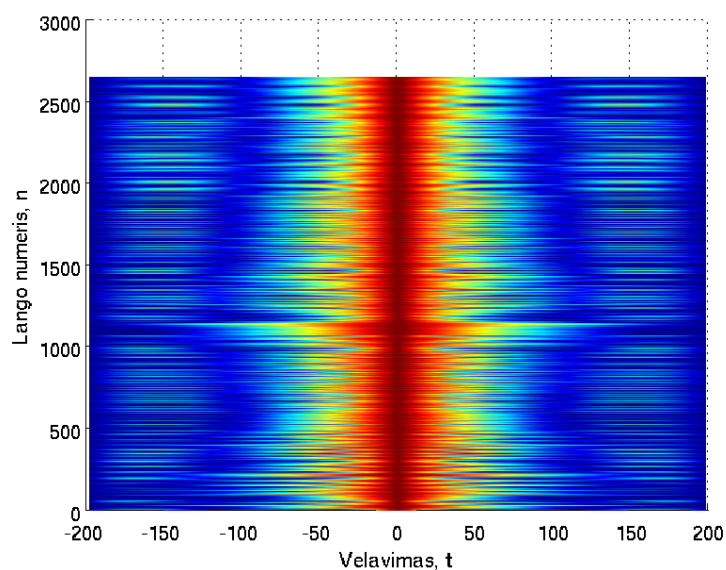
3.5 pav. Parkinsono liga sergančių subjektų kairės kojos žingsnių signalų dažninės komponentės

subjektas serga Parkinsono liga ar ne.

Tolimesnė analizė gali būti atlikta, remiantis koreliacijos koeficientais. Analizuojant šias savybes, galima remtis tokiu pačiu pirminiu signalo apdorojimo bloku, kaip ir analizuojant dažnines komponentes. Dėl to, kad esama tikrais kairės kojos signalais, galima taikyti tik savi-koreliacijos koeficientus. Narinėjama savybė parodė labai gerus rezultatus ankstesniame tyrime, kuriame, remiantis akselerometro ir giroskopo jutiklių parodymais, reikėjo suprojektuoti algoritmą, gebantį atskirti tokias žmogaus veiklas: stovėjimas,ėjimas,ėjimas aukštyn laiptais,



3.6 pav. Kontrolinių subjektų kairės kojos žingsnių signalų savi-koreliacijos koeficientai.

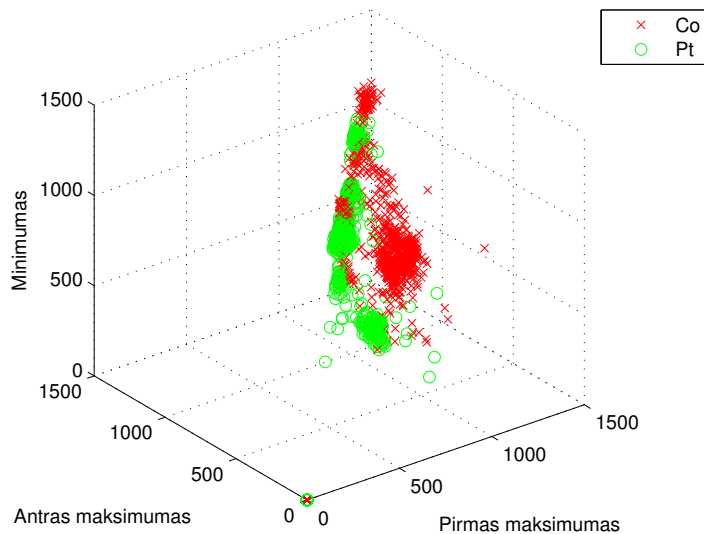


3.7 pav. Parkinsono liga sergančių subjektų kairės kojos žingsnių signalų savi-koreliacijos koeficientai.

ėjimas žemyn laiptais.

Kontrolinių subjektų kairės kojos savi-koreliacijos koeficientai parodyti 3.6 pav. Parkinsono liga sergančių subjektų kairės kojos savi-koreliacijos koeficientai parodyti 3.7 pav. Kaip matosi iš korelogramos, tiek sergantys, tiek kontroliniai subjektai turi panašias, o kai kuriais atvejais ir tokias pačias, koreliacijos reikšmes. Remiantis vien tik turima informacija, nustatyti ar subjektas serga Parkinsono liga ar ne, nėra įmanoma.

Sudėtingesnė analizė seka iš dviejų globalių maksimumų ir vieno lokalaus minimumo savybių



3.8 pav. Dviejų globalių maksimumų ir vieno lokalaus minimumo savybių erdvė.

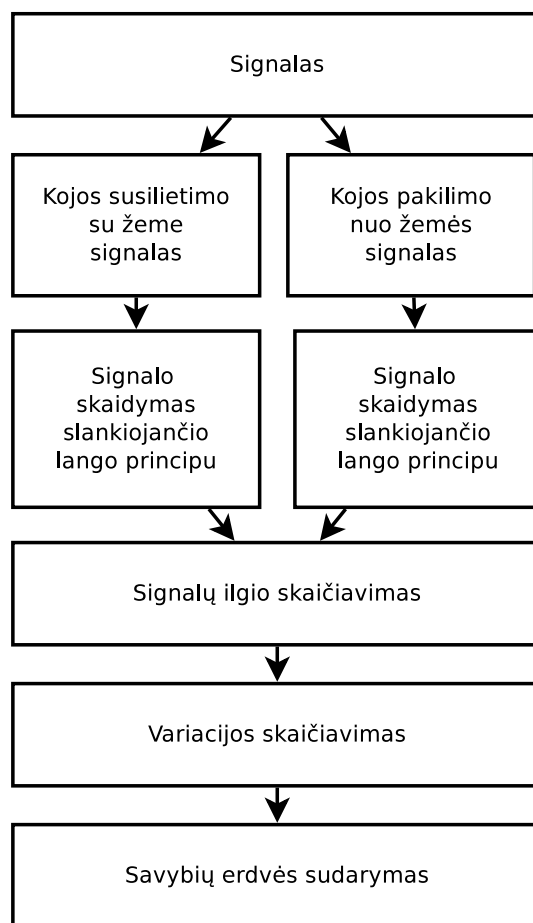
erdvės. Šios tris savybės sudaro trijų dimensių plokštumą, kurią galima lengvai pavaizduoti. Pilna savybių erdvė pavaizduota 3.8 pav. Kaip matosi iš duotos erdvės, duomenys neturi jokio koncentracijos centro. Erdvėje jie pasiskirstę pagal nežinomą dėsnį.

Tokių duomenų pateikti klasifikavimui nėra galima. Grafike “Co” taškai, pažymėti kryžiumi, parodo kontrolinį subjektą, “Pt” taškai, pažymėti apskritimu, parodo Parkinsono liga sergančius subjektus.

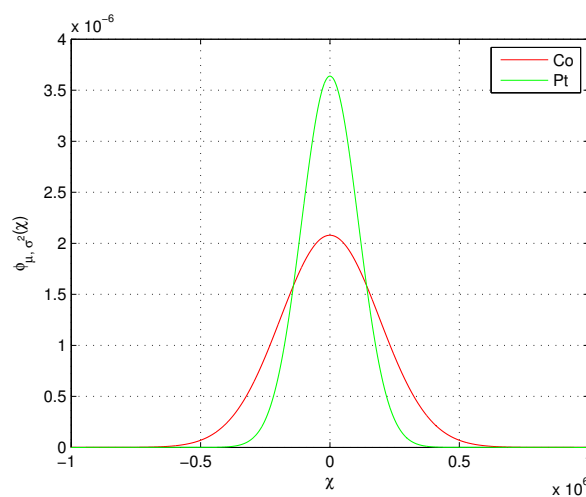
Sekančios savybės nagrinėjimui yra siūlomos atliktų tyrimų [?, ?, ?]. Nagrinėjama savybė yra žingsnio fazės laiko variacija. Kojos susilietimo su žeme laiko variacija ir kojos pakilimo nuo žemės laiko variacija.

Išskiriant variacijos savybes iš signalo, reikalinga nuo pagrindų pakeisti pirminio signalo apdorojimo algoritmą. Kadangi kojos pakilimas nuo žemės ir kojos susilietimas su žeme gali būti vienas nuo kito nepriklausomi (subjektas sustojo ir stovi dviem kojomis ant žemės arba subjektas stovi tik ant kairės kojos), pirminis signalo apdorojimas turi būti atliekamas lygiagrečiai, t.y. tuo pat metu išskiriamas tiek kojos pakilimo nuo žemės signalas, tiek ir kojos susilietimas su žeme signalas. Tokios sistemos struktūrinis grafikas yra parodytas 3.9 pav. Sekantis žingsnis yra signalo išskaidymas slankiojančio lango principu. Abiejų signalų slankiojančio lango ilgis buvo parinktas 4 verčių ilgio, su 2 verčių perdanga. Kuomet tiek kojos susilietimo su žeme, tiek kojos pakilimo nuo žemės keturi signalai patenkina keliamus reikalavimus - iš jų yra apskaičiuojami signalų ilgiai. Vėliau, abiejų etapų signalams yra paskaičiuojama jų variacija ir taip yra sudaroma savybių erdvė. Toliau bus nagrinėjamos abiejų signalų variacijos vienos dimensijos plokštumoje.

Abiejų signalų variacijų pasiskirstymai yra parodyti 3.10 pav. ir 3.11 pav. Abu pasiskirstymai turi vieną trūkumą - jų vidurkiai sutampa, tačiau gera žinia yra tai, kad jų variacijos

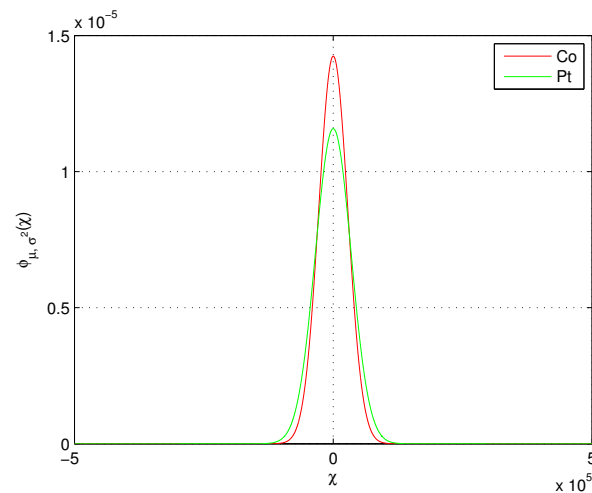


3.9 pav. Pirminis signalo apdorojimas, išskiriant savybę kaip kojos susilietimo su žeme laiko variaciją ir kojos pakilimo nuo žemės laiko variaciją.

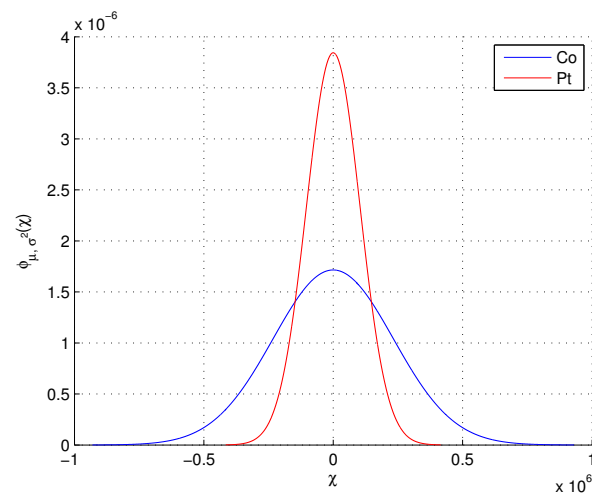


3.10 pav. Kojos prisilietimo prie žemės laiko variacija.

skirtingos. Norint išspręsti iškilusią problemą, reikia taikyti dimensijų praskyrimo metodus. Darbo metu buvo išbandyti tokie dimensijų praskyrimo metodai:



3.11 pav. Kojos pakilimo nuo žemės laiko variacija.

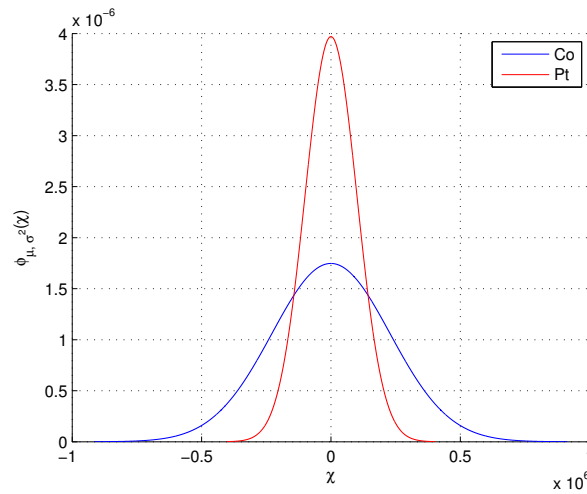


3.12 pav. Kojos prisilietimo prie žemės ir pakilimo nuo žemės laiko variacijos pasiskirstymas po PCA transformacijos.

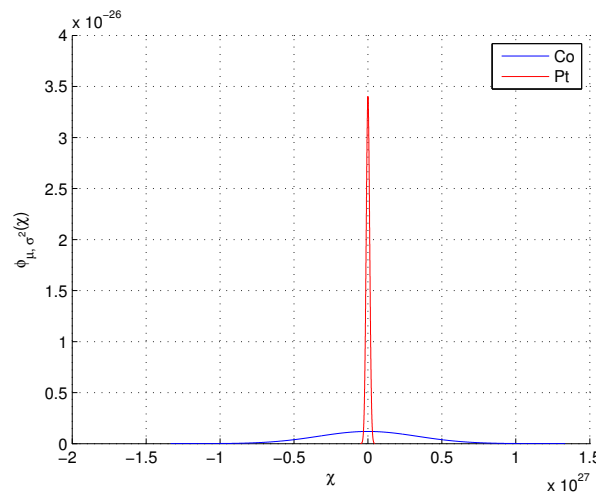
- PCA;
- LDA;
- Kernel PCA (su polinominiu, Gauso branduoliais);
- Kernel LDA (su polinominiu, Gauso branduoliais);

Vienmatė linijinė PCA transformacija pavaizduota 3.12 pav. Kaip matosi iš grafiko - vidurkis nepasikeitė, tačiau žymiai pasikeitė variacija. Vidurkio pokyčio nebuvimas pablogina situacija, kadangi to pačio vidurkio duomenis atskirti nėra įmanoma, todėl ir teikti duomenis klasifikavimui nėra prasmės.

Sekantis metodas yra LDA. Transformacijos rezultatas yra pavaizduotas 3.13 pav. Kaip matosi - tiek po LDA, tiek po PCA duomenų pasiskirstymas nėra kritiškai skirtingas. Galima



3.13 pav. Kojos prisilietimo prie žemės ir pakilimo nuo žemės laiko variacijos pasiskirstymas po LDA transformacijos.

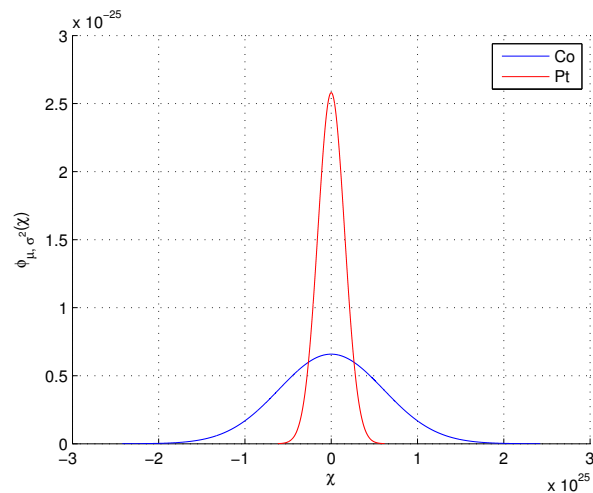


3.14 pav. Kojos prisilietimo prie žemės ir pakilimo nuo žemės laiko variacijos pasiskirstymas po PCA transformacijos, naudojant polinominį branduolį.

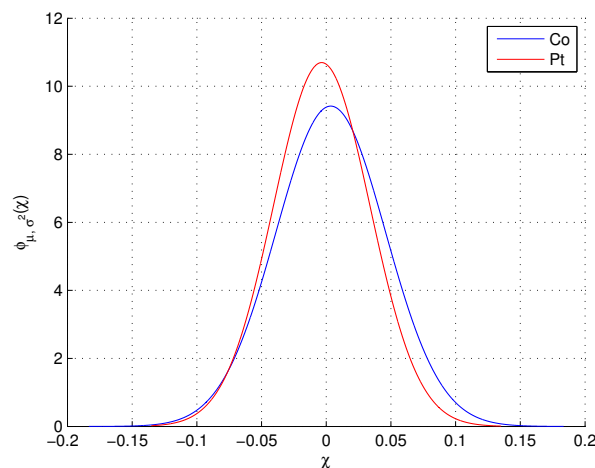
daryti hipotezę, kad linijinis duomenų praskirstymas šiuo atveju neveikia. Reikia taikyti branduolio metodą ir papildyti jau panaudotus transformacijos metodus polinominiu arba Gausso branduoliu.

Branduolio metodas pritaikytas PCA transformacijai yra pavaizduotas 3.14 pav. ir 3.16 pav. Branduolio metodas pritaikytas LDA transformacijai yra pavaizduotas 3.15 pav. ir 3.16 pav. Polinominis branduolys abiem atvejais buvo panaudotas su 4 branduolio argumentu.

Kaip matosi, panaudojus polinominį branduolį, tiek PCA (3.14 pav.), tiek LDA (3.15 pav.) atveju - vidurkis iš savo vietos nepajudėjo. Abidvi transformacijos pakeitė variaciją. Tai nėra norimas tikslas, todėl toks branduolys nėra tinkamas. Panaudojus Gauso branduolį rezultatas labai pagerėja LDA transformacijos atveju (3.17 pav.), PCA su tokiu branduoliu (3.16 pav.) duomenis tik dar labiau suvienodina.



3.15 pav. Kojos prisilietimo prie žemės ir pakilimo nuo žemės laiko variacijos pasiskirstymas po LDA transformacijos, naudojant polinominį branduolį.



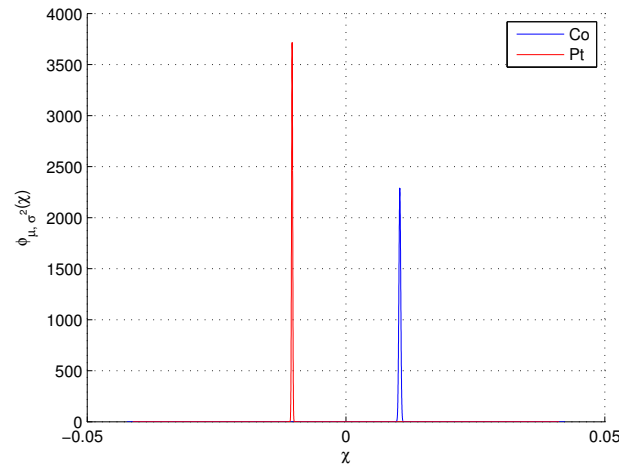
3.16 pav. Kojos prisilietimo prie žemės ir pakilimo nuo žemės laiko variacijos pasiskirstymas po PCA transformacijos, naudojant Gauso branduolį.

Remiantis pateikta analize galima teigti, kad geriausiai duomenis savybių erdvėje atskiria LDA su Gauso branduoliu. Tolimesniame darbe duomenis į klasifikatorių bus pateikiami būtent po tokios transformacijos.

3.4. Požymių klasifikavimo programos kūrimas

Šiame skyriuje bus išnagrinėti ir pritaikyti populiariausi šiuo metu naudojami klasifikatorių metodai. Tokie klasifikatoriai yra [?]:

- Vektoriaus palaikymo mašina (angl. Support Vector Machine (SVM));
- Paslėptas Markovo modelis (angl. Hidden Markov Model (HMM));



3.17 pav. Kojos prisilietimo prie žemės ir pakilimo nuo žemės laiko variacijos pasiskirstymas po LDA transformacijos, naudojant Gauso branduolį.

- Naivusis Bayes klasifikatorius (angl. NaiveBayes);
- Tiesioginio sklidimo neuronų tinklas (angl. Feed-Forward Neural Network (FFNN));

Labai svarbu yra atskirti duomenis, kuriais klasifikatorius yra apmokamas ir kuriais jis yra testuojamas. Jeigu klasifikatoriaus apmokymo duomenys yra pakankamai apibendrinti, tuomet naujus duomenis klasifikatorius turėtų gerai atpažinti. Pateikiant klasifikatoriaus testavime tokius pačius duomenis, kaip ir apmokyme - atliekamas tikrinimas ar klasifikatorius teisingai “suprato” nagrinėjamus duomenis, tačiau tai neapibrėžia kiek gerai jis apdoroja naujus duomenis.

Nurodytų klasifikatorių veikimas bus vertinamas taiklumo ir tikslumo parametrais. Taiklumas apskaičiuojamas:

$$\text{Taiklumas} = \frac{T_P + T_N}{T_P + F_P + T_N + F_N}, \quad (3.9)$$

kur T_P - teisingai identifikuotų klasių skaičius, T_N - teisingai atmestų klasių skaičius, F_P - klaidingai identifikuotų klasių skaičius ir F_N - klaidingai atmestų klasių skaičius.

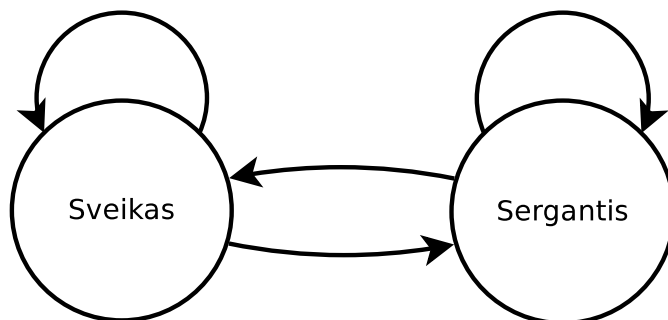
Tikslumas apskaičiuojamas:

$$\text{Tikslumas} = \frac{T_P}{T_P + F_P} \quad (3.10)$$

Visos tolesnės klasifikatorių patikros bus atliekamos remiantis tokia schema: iš turimų duomenų buvo išskirta kojos prisilietimo ir pakilimo nuo žemės laiko ilgio variacijos savybė. Tuomet visi duomenys buvo padalinti į lygias tris dalis: dimensijų operacijai atlikti buvo duota 500 reikšmių, klasifikatoriaus apmokymui 500 verčių, klasifikatoriaus testavimui 500 verčių. Klasifikavimas buvo atliekamas realiu laiku, t.y. klasifikatoriui pateikiami duomenis apie žingsnio

	Co	Pt
Tikslumas	0.489	0.489
Taiklumas	0.234	0.744

3.1 lentelė Vektoriaus palaikymo mašinos tikslumas ir taiklumas.



3.18 pav. Ergodinis paslėptas Markovo modelis

stadijos variaciją po transformacijos ir klasifikatorius pateikia savo spėjimą.

Vektoriaus palaikymo mašina [?] šiuo metu yra populiariausias klasifikatorius nagrinėjant netiesiškai atskiriamus duomenis. Įgyvendinimas Matlab aplinkoje buvo panaudotas iš [?]. Klasifikatoriaus testavimo metu buvo naudojami tokie parametrai:

- SVM tipas - Sequential Minimal Optimization;
- Branduolio tipas - linijinis;
- Suregulavimo konstanta - 2;
- Branduolio argumentas - 2;

Lentelėje 3.1 pateikti SVM tikslumo ir taiklumo duomenis. Kaip matosi, klasifikatorius veikia nepakankamai gerai - tikslumo koeficientas nėra didesnis negu pusę, abiem atvejais tik 0.489. Pirmos klasės atpažinimo taiklumas tėra 0.234, antros klasės atpažinimo taiklumas yra 0.744, tačiau to nepakanka. Iš padarytos klasifikatoriaus veikimo analizės galima teigti, kad klasifikatorius veikia blogai ir jo naudoti sprendime nėra galima.

Paslėptas Markovo modelis [?] yra vienintelis iš šiame darbe nagrinėjamų klasifikatorių, kuris turi laikinę informaciją. Tokia savybė suteikia “inkaro” galimybę - klasifikatorius gali užsilaikyti prie vienos klasės net ir tuomet, kai pagal savybių erdvę turi būti kitas klasifikavimo rezultatas. Tokia klasifikatoriaus savybė buvo pritaikyta ankstesniame darbe, sudarant žmogaus eisenos atpažinimo sprendimą [?]. Pasirinktas modelis, susidedantis iš dviejų būsenų - “Sveikas”, “Sergantis”. Būsenos tarpusavyje yra sujungtos (3.18 pav.). Perėjimo tikimybės tarp modelio elementų buvo parinktos žymiai mažesnės už tikimybę likti toje pačioje būsenoje. Įgyvendinimas Matlab aplinkoje buvo panaudotas iš [?].



	Co	Pt
Tikslumas	0.500	0.500
Taiklumas	0.000	1.000

3.2 lentelė Paslėpto Markovo modelio tikslumas ir taiklumas.

	Co	Pt
Tikslumas	0.508	0.508
Taiklumas	0.764	0.252

3.3 lentelė Naivaus Bayes klasifikatoriaus tikslumas ir taiklumas.

	Co	Pt
Tikslumas	0.282	0.287
Taiklumas	0.000	0.392

3.4 lentelė Tiesioginio sklidimo dirbtinių neuronų tinklų tikslumas ir taiklumas.

Lentelėje 3.2 pateikti tikslumo ir taikumo duomenys. Kaip matome iš tikslumo rezultato - klasifikatorius teisingai priskiria tik pusei reikiamų duomenų. Iš to galime teigti, kad klasifikatorius yra labai blogai apmokytas ir visiškai nesugeba apibendrinti turimų duomenų. Klasifikatorius visus pateikiamus testavimo duomenis priskiria vienai klasei.

Naivus Bayes [?] yra vienas iš pirmųjų statistinių metodų paremtu klasifikavimo mechanizmas. Jis veikia labai paprastai - ieškoma linijinės funkcijos, kuri geriausiai atskiria nagrinėjamus duomenis ir vieni žymenis priskiriami, jeigu duomenys yra vienoje linijos pusėje, atvirkšti žymenis priskiriami, jeigu duomenys yra kitoje linijos pusėje.

Lentelėje 3.3 pateikti tikslumo ir taiklumo duomenys. Tikslumo koeficientas viršija pusę, 0.508, tačiau tai yra mažai. Taiklumo koeficientas pirmuoju atveju yra neblogas, 0.764, tačiau antruoju atveju koeficientas yra visiškai nepatenkinamas, viso 0.252. Iš turimų rezultatų galima teigti, kad klasifikatorius veikia blogai ir duomenis vienmatėje erdvėje jis klasifikuoti teisingai negali.

Paskutinis klasifikatorius, kuris bus pritaikytas esamiems duomenims - tiesioginio sklidimo neuronų tinklas. Tai yra dirbtinių neuronų tinklų klasifikatorius, kuris veikia panašiai kaip ir Naivus Bayes - jis ieško funkcijos (linijinės arba polinominės), kuri geriausiai atskiria turimus duomenis. Bandymo metu buvo pasirinktas vienas įėjimas, du išėjimai ir vienas paslėptas sluoksnis.

Lentelėje 3.4 pateikiami tiesioginio sklidimo dirbtinių neuronų tinklų tikslumas ir taiklumas. Kaip matosi iš rezultatų - klasifikatoriaus tikslumas pirmos klasės atžvilgiu yra 0.282, taiklumas 0.000, antros klasės atžvilgiu klasifikavimo tikslumas yra artimas pirmai 0.287, tačiau turi geresnį taiklumą 0.392. Iš turimos patikros rezultatų galima spręsti, kad tiesioginio sklidimo neuronų tinklas užduotį atlieka blogai.

Iš pateiktos analizės galima spręsti, kad realiu laiku pateikti klasifikavimo rezultatai nėra pakankamai geri, norint atlikti kokybišką spėjimą ligos atžvilgiu. Vektoriaus palaikymo mašinos tikslumas neviršija pusės, paslėptas Markovo modelis nesugeba atlikti tikslingo apmokymo, naivus Bayes klasifikatorius užduotį atlieka geriausiai iš visų nagrinėjamų klasifikatorių, tiesioginio sklaidimo neuronų tinklas užduotį atlieka blogiausiai, iš visų klasifikatorių, kurie sugebėjo bent kažkiek apibendrinti duodamus duomenis apmokymo metu. Atlikta analizė reikalauja kito būdo klasifikavimui atlikti. Naudojamas būdas bus aprašytas sekančiame poskyryje.

3.5. Duomenų analizės programos kūrimas

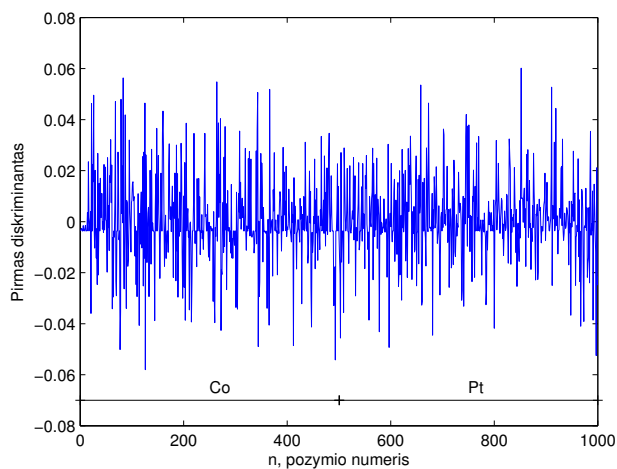
Šiame poskyryje bus aptartas duomenų analizės programos kūrimas. Pasinaudojus prieš tai esančių skyrių informacija bus pateiktas sprendimas, kuris leidžia efektyviai kiek įmanoma efektyviau atpažinti subjektų tipus.

Poskyryje “Požymių išskyrimo programos kūrimas” buvo atlikta savybių ir dimensijų mažinimo metodų analizė. Geriausia savybė, kuri atskiria kontrolinį subjektą nuo Parkinsono subjekto yra kojos pakilimo ir prisilietimo prie žemės laiko variacija. Geriausiai dimensijų mažinimo klausimą išsprendė linijinė diskriminanto analizė, naudojant Gauso branduolį, tačiau iškilo klasifikavimo problema - klasifikuojant duomenis realiu laiku, klasifikavimo rezultatas buvo nepatenkinamas.

Pagrindinė problema, kodėl joks nagrinėjamas klasifikatorius neatliko korektiško klasifikavimo, tai dėl jam pateikiamų duomenų. Kaip pavyzdys yra pateikiamas klasifikatoriaus testavimo metu naudoti duomenis, pavaizduoti 3.19 pav. Duomenis yra pateikiami po atliktos transformacijos, todėl jie yra vienos dimensijos. Horizontalėje yra pateiktas požymio eilės numeris, vertikalėje - pirmas LDA diskriminantas. Pirmi 500 požymiai priklauso kontroliniam subjektui, nuo 501 iki 1000 požymiai priklauso Parkinsono subjektui. Lyginant matomus duomenis su jų pasiskirstymu 3.17 pav., jie atrodo labiau chaotiški. Iš vaizdinės analizės nėra įmanoma spėti kuri signalo dalis kuriai subjektų grupei priklauso.

Tokios išvados verčia projektuoti kitą klasifikavimo mechanizmą, kuris duomenis klasifikuotų ne kiekvieną pagal kiekvieną nagrinėjamą savybę, o pagal savybės grupes - į laikiną atmintį yra rašoma požymio reikšmė, laukiama, kol laikina atmintis užsipildys iki tam tikros n eilės ir iš gautos sekos yra skaičiuojamas vidurkis (vidurkis skaičiuojamas todėl, nes pagal duomenų pasiskirstymą, kuris parodytas 3.17 pav., pasiskirstymų vidurkiai skiriasi, tačiau variacijos lieka tokios pačios) ir apskaičiuota reikmė naudojama kaip nauja savybė klasifikatoriaus apmokymui, bei testavimui.

Dėl gaunamų duomenų dimensijos ir jų mažo kiekio (kuris priklauso nuo laikinos atminties dydžio, į kurią rašomos naudos erdvės vertės), bus panaudotas paprasčiausias naivaus Bayes klasifikatoriaus mechanizmas. Dėl duomenų stokos, kai kurie duomenys buvo panaudoti visuose trijuose algoritmo veikimo žingsniuose. Viso erdvės sudarymui buvo panaudota 900, apmokymui 500, testavimui 700 reikšmių. Laikinos atminties ilgis buvo pasirinktas 150 verčių, iš kurių



3.19 pav. Kojos pakilimo ir kojos prisilietimo prie žemės ilgio variacijos kitimas slenkant langui, po transformacijos.

	Co	Pt
Tikslumas	0.800	0.800
Taiklumas	0.714	1.000

3.5 lentelė Gautas klasifikavimo rezultatas, naudojant naivų Bayes klasifikatorių.

buvo skaičiuojamas vidurkis. Apmokymo metu iš viso buvo 3 vidurkiai kiekvienai subjektų grupei, testavimo metu iš viso buvo 4 vidurkiai kiekvienai subjektų grupei. Gauti rezultatai pateikti 3.5 lentelėje.

Gauti klasifikavimo tikslumas yra 0.800, kas viršija prieš tai naudotų metodų tikslumą. Pirmos grupės klasifikavimo tikslumas yra 0.714, antros grupės klasifikavimo tikslumas 1.000. Detalesnė klasifikavimo mechanizmo veikimo apžvalga yra pateikta Požymių klasifikavimo programos kūrimo poskyryje.

4. Signalų analizės programos įgyvendinimas

Šiame skyriuje bus apžvelgta įgyvendintas programinis kodas, jo veikimo architektūra. Ankstesniuose poskyriuose buvo argumentuotai pažvelgti galimi analizės metodai, požymiai ir klasifikatoriai. Visi rezultatai bus panaudoti projektuojant galutinį sprendimą.

Poskyryje 4.1. bus pateikta bendra programos algoritmo veikimo schema, poskyryje 4.2. bus pateikta algoritmo klasifikavimo veikimo schema, poskyryje 4.3. bus apžvelgta galutinė programa.

4.1. Bendro programos algoritmo schemos sudarymas

Ankstesniame skyriuje buvo apžvelgta bendra programos veikimo schema. Bendros schemos pavyzdys yra pateiktas 3.1 pav. Šiame poskyryje bus pateikta detali algoritmo schema ir aptarta kiekviena jo bloko paskirtis, bei jame naudojamas metodas.

Viso programa yra sudaryta iš trijų modulių:

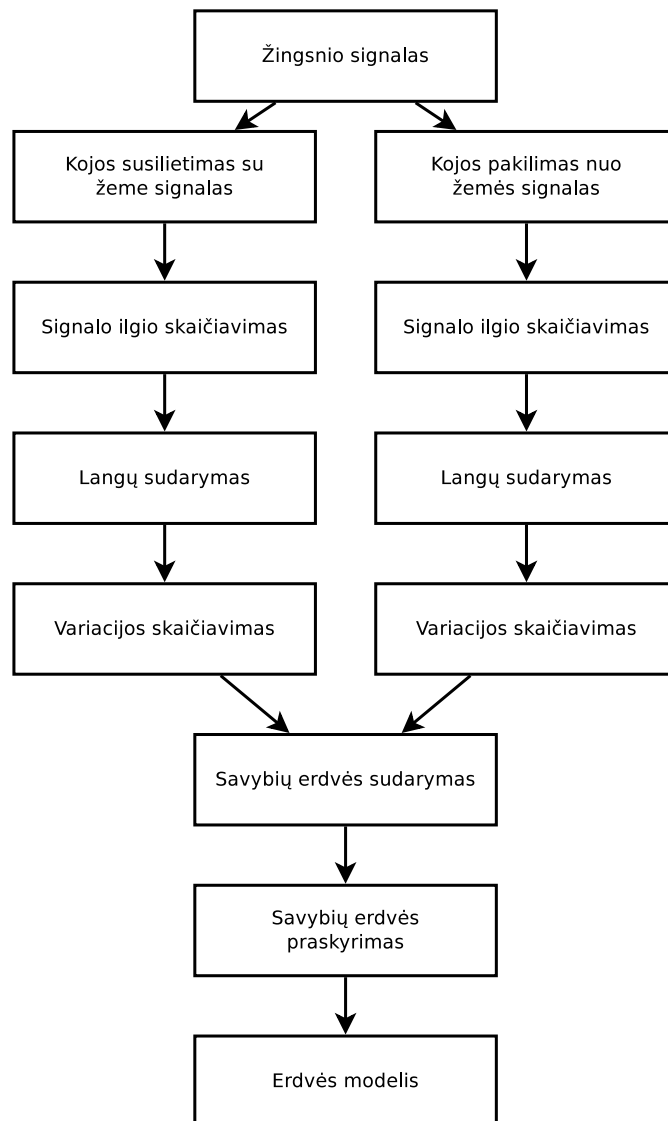
- Savybių erdvės sudarymas;
- Klasifikatoriaus apmokymas;
- Klasifikatoriaus tikrinimas.

Programos pirmo modulio struktūrinė schema pavaizduota 4.1 pav. Programos įėjime yra pateikiami pirminiai žingsnio duomenys, programa lygiagrečiai iš duomenų išskiria kojos susilietimo su žeme ir kojos pakilimo nuo žemės signalus, paskaičiuoja signalų ilgius, iš signalų ilgių sudaro duomenų langus, paskaičiuoja lange esančių signalų variacijos koeficientą. Toliau programa iš esamų dviejų dimensijų sudaro savybių erdvę, pritaiko LDA transformacija su Gauso branduoliu ir gražina erdvės modelį, kuris susideda iš tikrinio vektoriaus ir duomenų vidurkio.

Programos antro modulio struktūrinė schema pavaizduota 4.2 pav. Struktūroje naudojamas erdvės modelis iš pirmojo programos modulio. Jis reikalingas tam, kad kiekvieną kartą neprojektuoti naujos erdvės iš naujo, bet projektuoti naujus duomenis į jau esamą erdvę. Tai taip pat užtikrina, kad bus naudojama ta pati erdvė, kas leidžia užtikrinti duomenų apibendrinimą. Sistemos įėjimas yra toks pats, kaip ir pirmojo modulio. Skirtumas taip pirmojo ir antrojo modulio yra tas, kad antro modulio išėjime yra klasifikatoriaus modelis, kuris yra apmokyto klasifikatoriaus parametrai, kurie bus panaudoti testuojant klasifikatorių.

Trečiojo, paskutinio modulio, struktūrinė schema pavaizduota 4.3 pav. Struktūroje naudojamas erdvės modelis iš pirmo modulio ir klasifikatoriaus modelis iš antro modulio. Modulio įėjimas yra toks pats, kaip ir antrojo modulio, išskyrus tai, kad šio modulio išėjime yra klasifikavimo rezultatas, kurį galima pateikti tiek grafiškai, tiek skaitine išraiška, išreikšta tikslumo ir taiklumo verte.

Apibendrinus bendrą programos veikimo schemą, toliau paanalizuosim patį klasifikavimo mechanizmą - naivųjį Bayes klasifikatorių.



4.1 pav. Programos pirmo modulio grafikas.

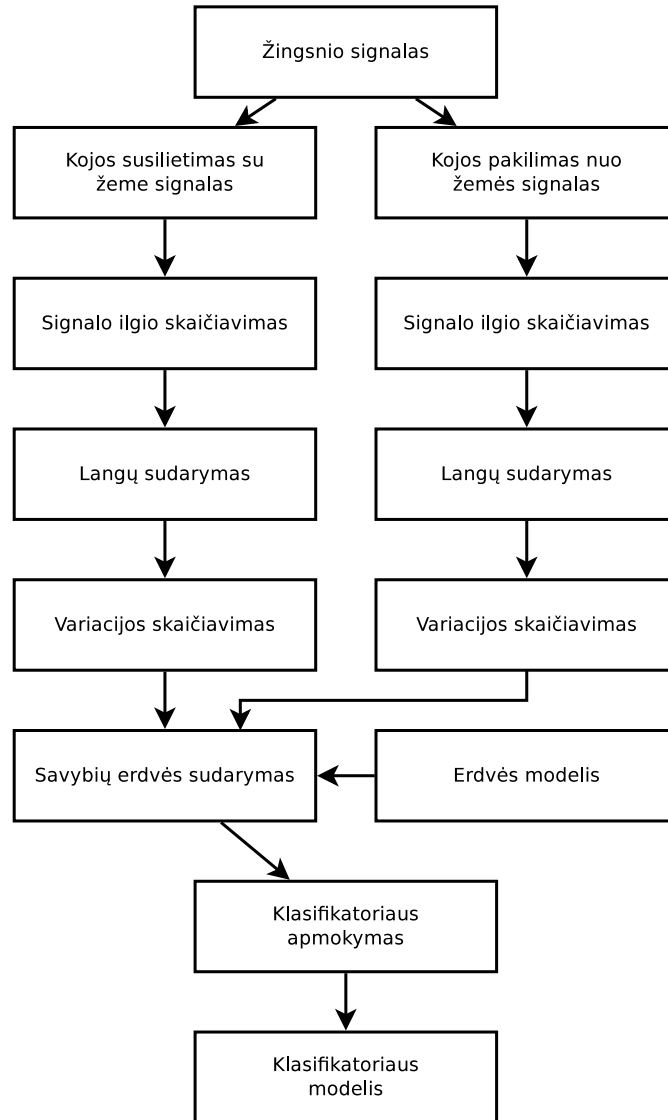
4.2. Požymių klasifikavimo programos algoritmo schemas sudarymas

Šiame poskyryje bus apžvelgtas naudojamas klasifikatorius, trumpai apžvelgtas jo veikimo principas, bei įgyvendinimas.

Naivus Bayes klasifikatorius [?] yra paprastas tikimybinis klasifikatorius, kuris paremtas Bayes teorema su stipria duomenų nepriklausomybės prielaida, kuri išreiškiama:

$$P(\mathbf{X}|C) = \prod_{i=1}^n P(X_i|C), \tag{4.1}$$

kur $\mathbf{X} = (X_1, \dots, X_n)$ yra savybių vektorius, o C yra klasės identifikatorius. Nepriklausomai nuo to, kad tokia duomenų atskyrimo prielaida yra labai naivi - praktikoje metodas veikia



4.2 pav. Antro modulio grafikas.

pakankamai gerai, ir yra naudojamas daugelį sudėtingesnių metodikų.

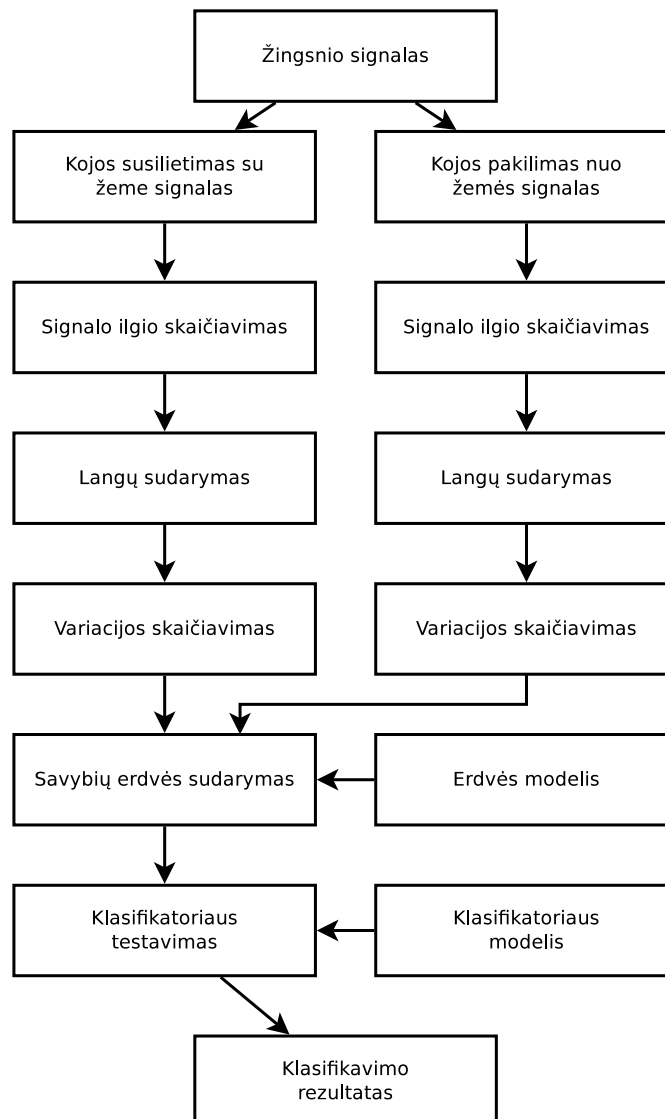
Bayes klasifikatorius $h^*(x)$, naudoja diskriminanto funkcijas atnaujinti klasės vėlesnes tikimybes, nurodant savybių vektorius:

$$f_i^*(x) = P(C = i | \mathbf{X} = x). \quad (4.2)$$

Pritaikius Bayes taisyklę nurodytai lygčiai, gaunamas rezultatas:

$$P(C = i | \mathbf{X} = x) = \frac{P(\mathbf{X} = x | C = i)P(C = i)}{P(\mathbf{X} = x)}, \quad (4.3)$$

kur $P(\mathbf{X} = x)$ yra identiška visoms klasėms, todėl yra ignoruojama. Iš to seka Bayes diskriminanto funkcija:



4.3 pav. Trečio modulio grafikas.

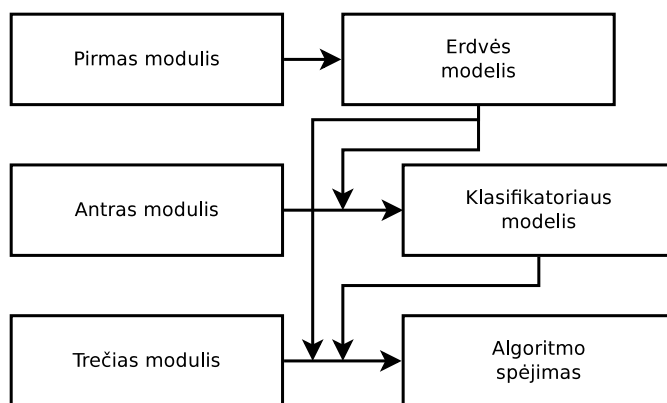
$$f_i^*(x) = P(\mathbf{X} = x | C = i)P(C = i), \quad (4.4)$$

kur $P(\mathbf{X} = x | C = i)$ vadinama klasės priklausomybės tikimybės pasiskirstymas. Taigi, Bayes klasifikatorius:

$$h^*(x) = \operatorname{argmax}_i P(\mathbf{X} = x | C = i)P(C = i) \quad (4.5)$$

randa didžiausią vėlesnės tikimybės hipotezę, nurodžius x .

Duomenys Hilberto erdvėje yra labai gerai praskiriami linijine funkcija, todėl šiame sprendime yra naudojamas tokios paprastos struktūros klasifikavimo mechanizmas. Tuo labiau, kadangi duomenis yra vienmatėje erdvėje - mažinti dimensių skaičių nelieka prasmės, ieškoti klasifikavimo algoritmo, kuris geriausiai apibendrintų turimus duomenis erdvėje nėra prasmės,



4.4 pav. Galutinės programos struktūrinė schema.

kadangi duomenų apibendrinimą puikiausiai atliko LDA su Gauso branduoliu.

Darbe bus panaudota Matlab aplinkoje įgyvendintas naivaus Bayes klasifikatoriaus versija, kuri yra “Statistical Toolbox” paketo dalis.

4.3. Signalų analizės programos įgyvendinimas

Šiame skyriuje apžvelgsime jau galutinę programą, aptarsime jos pritaikymo problemas, bei galimas pritaikymo platformas.

Galutinė programos schema yra pavaizduota 4.4 pav. Kaip buvo aptarta ankstesniuose poskyriuose - programa susideda iš trijų duomenų apdorojimo modulių. Kiekvieno modulio įėjimas yra toks pats, tačiau kiekvieno modulio išėjimas yra skirtingas. Kiekvienas žemiau esantis modulis naudoja virš jos esančio modulio darbo rezultatą. Kitiškai svarbu yra kiekvienam moduliui pateikti skirtingus duomenis.

Darbe pateiktas metodas leidžia efektyviai atpažinti ar subjektas serga Parkinsono liga. Didžiausias metodo trūkumas yra tas, kad ji neatsižvelgia į kitus Parkinsono ligos simptomus - drebulys, eisenos sustingimas. Nurodytas trūkumas yra planuojamas pašalinti vėlesnėse darbo versijose, kurios bus kuriamos po pateikto darbo įgyvendinimo.

Drebulys gali pasireikšti ne tik plaštakos raumenyse, tačiau ir kaklo srityje. Tai neleidžia tiksliai apibrėžti kurią kūno vietą reikia stebėti ir rinkti duomenis tyrimui. Eisenos sustingimas taip pat yra sunkiai apibrėžtas faktorius, kadangi jo aptikimas yra labai didelis iššūkis signalų apdorojimo srityje. Geriausią ką šiuo metu gali pasiūlyti mokslas, stebint tokius ligos simptomus - paciento stebėjimas kameros pagalba, jo veiklos automatinis nustatymas. Drebulys subjektui dažniausiai pasireiškia, kai jo kūnas yra visiškai atsipalaidavęs, t.y. kai subjektas stovi, sėdi, guli, kai jis išlaiko statišką poziciją, tuomet pasireiškia drebulys. Kameros pagalba galima nustatyti kokioje pozicijoje yra subjektas, tačiau identifikuoti drebulį yra labai sudėtinga, jeigu naudojama kamera yra mažos rezoliucijos. Drebulį veiklos nustatymo algoritmas gali palaikyti tiesiog pašaliniu triukšmu, kaip šešėlio sudarymą ant stebimo paviršiaus.

Darbas ties Parkinsono ligos identifikavimu ir jo diagnozavimu reikia įdėti dar daug darbo,



tačiau besivystant kompiuteriniai technikai, bei atsirandant vis naujiems algoritmams nestandartinėms problemoms spręsti - šansas, kad ateityje šios ligos diagnozavimas pagerės, išlieka labai didelis.



		Taškų rinkinio skaičius					
		500	400	300	200	100	21
Co	Tikslumas	0.700	0.800	0.450	0.450	0.500	0.450
	Taiklumas	0.750	0.875	0.467	0.429	0.500	0.400
Pt	Tikslumas	0.700	0.800	0.450	0.450	0.500	0.450
	Taiklumas	0.667	0.875	0.400	0.462	0.500	0.467

5.1 lentelė Klasifikavimo rezultatai, mažinant linijiškai pateikiamų duomenų dimensijų mažinimo algoritmui.

		Taškų rinkinio skaičius					
		500	400	300	200	100	21
Co	Tikslumas	0.700	0.650	0.700	0.600	0.600	—
	Taiklumas	0.750	0.714	0.750	0.750	0.750	—
Pt	Tikslumas	0.700	0.650	0.700	0.600	0.600	—
	Taiklumas	0.667	0.615	0.667	0.562	0.750	—

5.2 lentelė Klasifikavimo rezultatai, mažinant linijiškai pateikiamų duomenų klasifikatoriaus apmokymo algoritmui.

5.2. Duomenų eksperimentams rengimas

Šiame poskyryje bus aptartas duomenų eksperimentams rengimas.

Pradžioje, bus pratestuotas dimensijų mažinimo algoritmas, mažinant pateikiamų duomenų skaičių. Kadangi struktūriškai buvo nuspręsta, kad kiekvienas lygmuo turi lygiai po 500 taškų, testavimas bus pradėtas nuo 500 taškų rinkinių ir mažinamas kas 100 rinkinių, vadinasi, iš viso dimensijų mažinimo algoritmui bus pateikta 500, 400, 300, 200, 100 ir minimalus 21 rinkinių skaičius. Į apmokymo lygmenį bus pateikta po tiek rinkinių, kiek ir buvo nuspręsta struktūriškai analizuojant gautą algoritmą. Laikinos atminties dydis bus palaikomas pastovus - 50 verčių. Po dimensijų mažinimo žingsnio testavimo, rinkinių skaičius tokia pat metodika bus mažinamas klasifikatoriaus apmokymui. Klasifikatoriaus testavimo atveju duomenų dydis nebus mažinamas, kadangi gauta informacija nepateiks visiškai jokio naudingo rezultato.

Laikinos atminties ilgio tikrinimo metu bus panaudoti tokie laikinos atminties ilgiai: 10, 30, 50, 70, 100, 130 ir 150. Kiekvienas toks tikinimas bus atliktas su duomenų kiekio mažinimo tikrinimu, kuris buvo aprašytas ankstesniame paragrafe. Panaudoti rezultatai atitinkamai garantuos optimalų duomenų kiekio pasiskirstymą, geriausiam klasifikavimo rezultatui pasiekti.

5.3. Programos patikros rezultatai

Šiame poskyryje bus pateikti ir aptarti patikros rezultatai patikros rezultatai, išanalizuoti galimi sistemos geriausiai parametrai, kurie gali būti derinami: duomenų skaičius, kuris yra paduodamas kiekvienam programos moduliui, bei atminties dydis, kuris naudojamas pirmajai diskriminanto vertei saugoti.

		Taškų rinkinio skaičius					
		500	400	300	200	100	21
Co	Tikslumas	0.550	0.450	0.600	0.600	0.450	0.650
	Taiklumas	0.533	0.467	0.583	0.583	0.400	0.667
Pt	Tikslumas	0.550	0.450	0.600	0.600	0.450	0.650
	Taiklumas	0.600	0.400	0.625	0.625	0.467	0.636

5.3 lentelė Klasifikavimo rezultatai, mažinant nelineiškai pateikiamų duomenų dimensijų mažinimo algoritmui.

		Taškų rinkinio skaičius					
		500	400	300	200	100	21
Co	Tikslumas	0.550	0.450	0.500	0.400	0.550	—
	Taiklumas	0.553	0.455	0.500	0.400	1.000	—
Pt	Tikslumas	0.550	0.450	0.500	0.400	0.550	—
	Taiklumas	0.600	0.444	0.500	0.400	0.526	—

5.4 lentelė Klasifikavimo rezultatai, mažinant nelineiškai pateikiamų duomenų klasifikatoriaus apmokymo algoritmui.

		Laikinosios atminties dydis						
		150	130	100	70	50	30	10
Co	Tikslumas	0.667	0.667	0.800	0.429	0.550	0.625	0.410
	Taiklumas	0.600	0.600	0.714	0.444	0.556	0.700	0.390
Pt	Tikslumas	0.667	0.667	0.800	0.429	0.550	0.625	0.410
	Taiklumas	1.000	1.000	1.000	0.400	0.545	0.591	0.424

5.5 lentelė Klasifikavimo rezultatai, mažinant laikinosios atminties dydį.

Pirma patikra buvo atlikta linijiškai mažinant duomenų kiekį, kuris yra paduodamas dimensijų mažinimo algoritmui, taip siekiant nustatyti koks duomenų kiekis geriausiai tinka naujai savybių erdvei projektuoti. Eksperimentiniai duomenys yra pateikti 5.1 lentelėje. Vertinant dimensijų sudarymą pagal klasifikatoriaus duodamą rezultatą, geriausias duomenų kiekis, skirtas naujai dimensijų erdvei sudaryti yra 400 erdvės taškų. Tokiu atveju pasiekiamas 0.800 tikslumas ir 0.875 tikslumas.

Antra patikra buvo atlikta linijiškai mažinant duomenų kiekį, kuris yra paduodamas klasifikatoriaus apmokymui, taip siekiant nustatyti, kokio duomenų kiekio reikia klasifikatoriui, kad jis sugebėtų apibendrinti duomenis, naudojamus testavimo stadijoje. Eksperimentiniai duomenys yra pateikti 5.2 lentelėje. Šio tikrinimo metu paaiškėjo, kad geriausiai klasifikatoriaus apmokymui tinka du duomenų rinkiniai - po 500 ir po 300. Tokiu atveju pasirenkamas didesnis duomenų kiekis, taip užtikrinant bendrų duomenų apibrėžtumą.

Trečia patikra buvo atlikta nelineiškai mažinant duomenų kiekį, kuris yra paduodamas dimensijų mažinimo algoritmui. Klasifikavimo rezultatai yra pateikiami 5.2 lentelėje. Bendri klasifikavimo rezultatai yra prastesni, negu mažinant duomenų pateikimą linijiniu būdu, todėl iš eksperimento galima teigti, kad duomenis į dimensijų mažinimo algoritmą geriausiai yra pateikti linijiniu būdu - nuo 1 eilės iki 400 eilės numerio. Taip duomenis yra geriau "apibendrinami" dimensijų mažinimo algoritmo, ką parodo klasifikavimo rezultatas.

Ketvirta patikra buvo atlikta nelineiškai mažinant duomenų kiekį, kuris yra paduodamas klasifikatoriaus apmokymo metodui. Klasifikavimo rezultatai yra pateikiami 5.4 lentelėje. Kaip ir trečios patikros atveju - klasifikavimo duomenis nėra patenkinami, todėl ir klasifikavimo apmokymo atveju, duomenis geriausiai pateikti linijiniu būdu - nuo 0 eilės iki 500 eilės numerio.

Paskutinė patikra buvo atlikta keičiant laikinosios atminties dydžio matmenis. Duomenų erdvei sudaryti buvo panaudoti duomenys, remiantis pirma patikra, klasifikatoriaus apmokymui buvo panaudoti duomenys, remiantis antra patikra. Klasifikavimo rezultatai yra pateikti 5.5 lentelėje. Geriausias laikinosios atminties lango dydis, prie kurio pasiekiamas geriausias klasifikavimo rezultatas yra 100 pirmojo diskriminanto verčių.

Iš pateiktos analizės, buvo priimti tokie sistemos veikimo parametrai:

- Naujai duomenų erdvei konstruoti panaudoti duomenis nuo 1 iki 400 eilės numerio;
- Klasifikatoriaus apmokymui panaudoti duomenis nuo 500 iki 1000 eilės numerio;
- Laikinosios atminties dydis, kuriame saugomos pirmojo diskriminanto vertės, 100 eilės.

Patikros metu buvo nustatyta, kad programa geriausiai veikia prie 80 % tikslumu, kas įveda nepasitikėjimo faktorių, kuris lygus $1/5$ visų rezultatų tikslumu. Toks rezultatas yra pakankamai geras tik tuo atveju, jeigu aprašytas įrankis bus panaudotas mažiausiai penkis kartus, norint galutinai nurodyti, kad pacientas turi arba neturi Parkinsono ligos, pagal eisenos sutrikimo simptomą.

