



TURINYS

Žymenys ir santrumpos	8
1. Įvadas. Užduoties analizė	9
2. Informacinių Parkinsono ligos diagnostikos sistemų apžvalga	12
3. Signalų analizės programos kūrimas	20
3.1. Bendros programos struktūrinės schemos sudarymas	20
3.2. Pirminio signalų apdorojimo programos kūrimas	21
3.3. Požymių išskyrimo programos kūrimas	27
3.3.1. Dimensijų mažinimas	27
3.3.2. Savybių tyrimas	29
3.4. Požymių klasifikavimo programos kūrimas	36
3.5. Duomenų analizės programos kūrimas	40
4. Signalų analizės programos įgyvendinimas	42
4.1. Bendro programos algoritmo schemos sudarymas	42
4.2. Požymių klasifikavimo programos algoritmo schemos sudarymas	43
4.3. Signalų analizės programos įgyvendinimas	46
5. Signalų analizės programos patikra	48
5.1. Eksperimentų plano rengimas	48
5.2. Duomenų eksperimentams rengimas	49
5.3. Programos patikros rezultatai	49
6. Rezultatų apibendrinimas	53
Literatūros ir informacinių šaltinių sąrašas	54
Santrauka	57
PRIEDAI	58



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;
- Hz – ciklų skaičius per vieną sekundę (angl. Hertz);
- SVM – vektoriaus palaikymo mašina (angl. Support Vector Machine);
- STC – signalo nuokrypio skaitiklis (angl. Signal Turn Counter);



1. Įvadas. Užduoties analizė

Bakalauro baigiamojo darbo tema – Parkinsono ligos eigos stebėjimo programa. Parkinsono liga yra dažniausiai pasitaikantis neurodegeneracinis judėjimo sutrikimas. Ankstyva ligos diagnozė ir efektyvus terapijos stebėjimas yra būtinas pacientų gydymui ir ligos. 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 viešai prieinama duomenų bazė, kurioje yra pateikiami sveikų ir sergančių Parkinsono liga žmonių eisenos duomenys. Duomenų bazė vadinasi “PhysioBank” [14].

Darbo tikslas – sukurti programą, kuri atpažintų sveiką ir Parkinsono liga sergantį žmogų, remiantis vertikalios jėgos jutikliais gautais signalais. Programa įgyvendinta Matlab platformoje. Ji pasirinkta dėl plataus įrankių kiekio, kuris yra įgyvendintas Matlab aplinkoje. Nurodytoje platformoje taip pat yra labai patogų ir greitą realizuoti signalų apdorojimo sistemas dėl jos architektūros – visi kintamieji yra matricos.

Darbe naudojami 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 veikiančiam objektui. Kaip pavyzdžiui, stovinti žmogus slekia žemę jėga, kuri lygi jo masei ir tuo pat metu, žemė slekia žmogų priešinga, lygiai tokia pačia jėga. Jutikliai parinkti dėl šaltinyje [20] pateiktos informacijos, kad VŽRJ yra geras žmogaus kūno stabilumo indikatorius. Duomenų surinkimo platformos ir duomenų surinkimo sesijų atlikinėti nereikėjo, kadangi naudojami duomenys yra prieinami duomenų bazėje.

Šiuo metu Parkinsono liga yra aktyviai tiriama [1], kadangi ją susirgęs žmogus praranda galimybę laisvai gyventi. Motorinės funkcijos sutrikimai padaro gyvenimą problematišką: drebulys sukelia nepatogumų atliekant didesnio kruopštumo reikalaujančius darbus, galūnių sustingimas sukelia didelių nepatogumų atliekant paprasčiausius kasdienes procedūras. Toliau pasireiškia psichiniai sutrikimai, dėl neurologinio sutrikimo: depresija, apatija, miego sutrikimas. Tik ankstyva ligos diagnozė gali sulėtinti ligos plitimą ir pagelbėti ją susirgusiems žmonėms. Dėl šios priežasties darbo tema pasirinkta Parkinsono liga.

Didžiausia ligos atpažinimo problema slypi savybių, kurios geriausiai identifikuos sergantį Parkinsono liga nuo sveiko, nesergančio asmens. Darbe apžvelgtos kelios galimos savybės, kurios gali atskirti tokius subjektus, remiantis analoginių sistemų apžvalga. Iš galimų savybių sudaroma savybių erdvė, kurią naudojantis atliekamas subjektų atpažinimas. Blogiausias galimas variantas būtų nelinejinė funkcija atskirti didelių dimensijų savybių duomenys. Geriausias galimas variantas būtų mažos dimensijos duomenys (iki 3 dimensijų) ir linijiškai atskiriami duomenys (kadangi linijinę funkciją realizuoti yra lengviausia). Turint didelių dimensijų duomenis, planuojamos sistemos aparatiniai reikalavimai automatiškai didėja, kadangi būtina ap-

doroji didelį kiekį duomenų. Tokią problemą galima išspręsti panaudojus dimensijų mažinimo algoritmus, kurių dažniausiai taikomi [9]: Principinių komponentų analizė (angl. Principal Component Analysis (PCA)), linijinė diskriminanto analizė (anl. Linear Discriminant Analysis (LDA)). Dar žinomi kaip dimensijų praskyrimo algoritmai. Jie taikomi, kuomet duomenys yra atskirti netiesiškai. Duomenims, kurie atskirti tiesiniu dėsniu, galima taikyti paprastą klasifikavimo algoritmą. Jeigu duomenis taip ir nepavyksta atskirti tiesiškai, tenka taikyti kompleksinį klasifikatorių, ko pasekoje gali labai sumažėti klasifikavimo tikslumas ir taiklumas. Darbe panaudotas Matlab aplinkoje įgyvendintas įrankis, kuris skirtas suprojektuoti naują dimensijų plokštumą, kuri įgyvendino Vojtech Franc savo magistriniam darbe [13].

Sukurtas produktas gebės pateikti diagnozės rezultatą – ar subjektas turi Parkinsono liga sergančių subjektų eisenos požymių ar jų neturi. Produktas neatsižvelgs į kitus ligos simptomus: drebulys (rankų, kojų, žandikaulio, galvos), standumas (galūnių arba liemens sustingimas), bradikinezija (judesių lėtumas), pozicijos nestabilumas (arba sutrikęs balansas). Pati programa duomenis analizuos jau po duomenų surinkimo. Tai reiškia, kad pirmiausiai duomenys yra surenkami, o vėliau įkeliami į programą tolimesniam apdorojimui.

Darbo tema, Parkinsono ligos eigos stebėjimo programa, reiškia, darbo rezultate yra sukurtas algoritmas, įgyvendintas kompiuterine programa. Pačiam kompiuteryje turi būti veikiantis Matlab programinis paketas. Programa parašyta naudojantis Matlab 7.12.0 (R2011) versija su “Neural Network Toolbox” ir “Statistical Toolbox” įrankiu. Eigos stebėjimas reiškia, kad visuomet egzistuoja neapibrėžtas, galimas programos netikslumas. Visiškai programa remtis, diagnozuojant Parkinsono ligą nėra galima, kadangi, kaip jau minėta anksčiau – eigos sutrikimas nėra vienintelis ligos požymis. Turi būti atlikti ir kiti tyrimai, norint tiksliai diagnozuoti ligą.

Darbo objekto sudėtis yra asmeniniam kompiuteriui skirta programa, jėgos jutiklių signalų generavimo programa. Kokiam kompiuteriui kuriama programa, paminėta darbo užduoties aprašyme. Jėgos jutiklių generavimo programa (modulis) atsakinga už signalų nuskaitymą iš duomenų bazės ir jų pateikimą sistemos algoritmui. Taip pat signalų generavimo programa (modulis) naudojamas programos demonstracinei versijai įgyvendinti.

Duomenys sistemai pateikiami iš “PhysioBank” duomenų bazės [14]. Duomenys duomenų bazėje surinkti diskretizuojant signalus 100 Hz diskretizavimo dažniu. Kiekvienu laiko momentu yra įrašoma nauja eilutė į duomenų tekstinę bylą. Eilutę sudaro 19 skilčių:

- Skiltis 1 nurodo laiką (sekundėmis);
- Skiltys 2–9 nurodo kairės kojos 8 jutiklių VŽRJ, Niutonais;
- Skiltys 10–17 nurodo dešinės kojos 8 jutiklių VŽRJ, Niutonais;
- Skiltis 18 nurodo kairės kojos suminę VŽRJ, Niutonais;



- Skiltis 19 nurodo dešinės kojos suminę VŽRJ, Niutonais;

Duomenų bazės bylų pavadinimai, pavyzdžiui: “GaCo01_02.txt” ar “JuPt03_06.txt”, sudaryti nurodant duomenų rinkimų sesijų pavadinimus: “Ga” – “Galit Yogev et al” (dual tasking 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, atliekamas klasifikavimo mechanizmu, vertinamas taiklumu ir jautrumu. Parametrai yra apskaičiuojami iš pasiklovimo matricos.

2. Informacinių Parkinsono ligos diagnostikos sistemų apžvalga

Šiame skyriuje apžvelgtos analoginės informacinės sistemos arba bandymai sukurti sistemą, kuri, remiantis įvairių jutiklių pagalba, gebėtų atpažinti Parkinsono ligą. Apibendrintai aptarti nagrinėjamų sistemų teigiamos ir neigiamos savybės – ar pasirinkti požymiai yra argumentuoti ar naudojamas klasifikatorius yra parinktas atsižvelgus į naudojamą savybių erdvę, kokie gauti rezultatai.

Pirmas darbas apžvelgtai yra “Statistical Analysis of Gait Rhythm in Patients With Parkinson’s Disease” [23]. Šis darbas nagrinėja galimas žingsnio variacijos galimybes ir pateikia signalo nuokrypio skaičiavimą kaip pagrindinę savybę, atskiriančią kontrolinius subjektus nuo sergančių. Nagrinėjime panaudoti žingsnio pakilimo nuo žemės ilgis, žingsnio prisilietimo prie žemės ilgis ir bendras žingsnio ilgis. Duomenys gaunami iš vertikalios žemės reakcijos jėgos matavimų. Autorių teigimu, žingsnio pakilimo ir nusileidimo ilgiai turi aukštą koreliacijos koeficientą su bendro žingsnio ilgiu, todėl nagrinėjime panaudotas tik kairės kojos žingsnio ilgis (2.1 lentelė). Gautas savybių vektorius apskaičiuotas, naudojantis “vieno palikimo-išmetimo tarpusavio patikros” (angl. leave-one-out cross-validation) metodu. Jis parodė, kad mažiausių šaknų vektoriaus palaikymo mašina (angl. Support Vector Machine), su daugianariu branduoliu sugeba klasifikuoti duomenis su 90,32 % tikslumu.

Nagrinėjime naudoti duomenis, pateikti “Hausdorff et al” [6], kurie yra laisvai prieinami iš PhysioNet duomenų bazės [14]. Tokie duomenys yra panaudoti ir šiame darbe, todėl reikia įsigilinti ir į subjektus, kurie dalyvavo duomenų rinkimo metu. Tyrime dalyvavo 16 sveikų subjektų, 20 – 74 metų amžiaus. Kontroliniai subjektai yra visiškai sveiki neurologiškai, neturi jokių širdies ligų sutrikimų. Parkinsono subjektų skaičius yra 15, kurių amžius variavo nuo 44 iki 80 metų. Ūgis ir svoris žymiai nesiskyrė tarp kontrolinių ir Parkinsono subjektų, todėl į tokius duomenis dėmesys nebuvo kreipiamas. Parkinsono subjektai vertinami pagal “Hoehn and Yahr (HAY)” vertinimo sistemą, nustatant jų ligos stadiją. Du pacientai surinko 1,5 HAY balo (lengva stadija), keturi pacientai surinko tarp 2 ir 2,5 HAY balo, penki surinko 3 HAY balo (vidutinė stadija), likusieji surinko virš 3 HAY balų, jie turėjo žymesnių eisenos sutrikimų.

Remiantis gauta pasiskirstymo funkcija, kuri žymima $\hat{p}(x_b)$, kur $x_b, b = 1, 2, \dots, B$, kur B nusako segmentų skaičių, skaičiuojant x amplitudės ruožą, vidurkis μ ir variacija σ^2 apskai-

2.1 lentelė Koreliacijos koeficientai tarp σ bendro žingsnio (σ_r), kojos pakilimo (σ_w) ir kojos prisilietimo (σ_a) prie žemės ilgio parametrų [23]

σ	σ_r	σ_w	σ_a
σ_r	1,00	0,99	0,94
σ_w	0,99	1,00	0,94
σ_a	0,94	0,94	1,00

2.2 lentelė Bendro žingsnio, kojos pakilimo, kojos prisilietimo prie žemės vidurkio, vidutinio nuokrypio duomenys [23]

Žingsnio fazė	Statistinis parametras	Kontrolinis subjektas	Parkinsono subjektas
		Vidurkis \pm nukrypimas	Vidurkis \pm nukrypimas
Bendras žingsnis	$\mu(s)$	$1,09 \pm 0,09$	$1,12 \pm 0,12$
	$\sigma(s)$	$0,03 \pm 0,01$	$0,06 \pm 0,04$
	STC	$12,44 \pm 10,46$	$55,87 \pm 29,89$
Kojos pakilimas	$\mu(s)$	$0,39 \pm 0,04$	$0,37 \pm 0,05$
	$\sigma(s)$	$0,02 \pm 0,01$	$0,04 \pm 0,02$
	STC	$8,5 \pm 9,39$	$54,13 \pm 26,96$
Kojos prisilietimas	$\mu(s)$	$0,70 \pm 0,06$	$0,75 \pm 0,09$
	$\sigma(s)$	$0,02 \pm 0,01$	$0,05 \pm 0,04$
	STC	$15,81 \pm 12,07$	$61,27 \pm 25,62$

čiuojama:

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

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

Tyrimo metu 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 signalo nuokrypio skatiklio duomenys (angl. Signal Turn Counter STC) savybės vertės. Signalo $x(i)$ 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 ypatybių. Tokio matematinio “derinimo” rezultate gali įvykti, kad algoritmas 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 prisilie-

timo prie žemės signalo ilgio dispersija (σ) gali būti panaudoti kaip savybės, norint identifikuoti Parkinsono subjekta.

Tolimesnis darbas, pateikiamas apžvalgai “Gait feature extraction in Parkinson’s disease using low-cost accelerometers” [19], nėra naudojama vertikali žemės reakcijos jėga, skaičiuojant savybes. Sprendime panaudoti pigūs, keturi linijinio pagreičio jutikliai. Toks sprendimas yra lengvesnis, už prieš tai apžvelgtą, kadangi jame panaudoti šešiolika vertikalios žemės reakcijos jutiklių. Nagrinėjamos Parkinsono ligos eisenos stingimo ir nesimetriškumo požymiai. Iš pirminės sprendimo apžvalgos iškarto išryškėja sprendimo trūkumas – tiriant galimas savybes, panaudotas tik vienas Parkinsono liga sergantis 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 negalioti kitiems subjektams ar subjektų grupei. Egzistuoja ir tokia tikimybė, kad ištirtos savybės, kurios galioja Parkinsono subjektui, gali galioti ir tyrime nedalyvavusiam kontroliniam subjektui, kadangi panaudotas tik vienas kontrolinis subjektas tyrimo metu.

Nepriklausomai nuo atlikto tyrimo siaurumo, darbe pateiktos svarbios eisenos analizės išvados. Kontrolinio subjekto eisenoje nepastebėta jokių eisenos sustingimo požymių, kairės ir dešinės kojos motorinės operacijos atliekamos simetriškai, išskyrus kairės kojos kelio didžiausios amplitudės pagreičio kojos susilietimo ir pakilimo su žeme metu. Abiejų kojų susilietimo su žeme laiko ilgis simetriškas, kas ir patvirtina pusiausvyrą eiseną. Parkinsono subjektas parodė dešinės kojos eisenos sąstingio nesimetriškumą. Pacientas naudojo kompensacijos mechanizmą – jis peržengdavo dešinės pusės judesius, kuomet pradėdavo ėjimą. Kairės pusės judesiai atliekami tik palaikant balansą. Pati eiseną yra labai lėta, o sustingimas įvyksta tik dešinėje pusėje. Iš gautų rezultatų galima padaryti išvadą, kad analizuojamas pacientas visuomet bando kompensuoti eisenos pokytį, naudodamasis vieno kūno šonu. Tyrimas atliktas tik su vienu pacientu, todėl nėra galima tiksliai teigti, kad kompensacija vyksta tik iš dešinės pusės, tačiau ką galima tvirtai sakyti, kad naudojantis kompensacijos mechanizmu – egzistuoja žingsnio variacijos tikimybė. Tokią hipotezę sėkmingai patvirtina ir prieš tai atlikta darbo analizė [23]. Kompensacijos nebuvimas kontrolinio subjekto atveju garantuoja, kad žingsnio ilgio variacija nėra 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” [21]. 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

2.3 lentelė Klasifikavimo jautrumas, taiklumas, klaidingai klasifikuotas rodilis ir tikslumas, naudojant pastovų ir kintantį klaidingo klasifikavimo metodo kriterijų [21]

	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 %

2.4 lentelė Sunkios, lengvos ir kontrolinio subjekto eisenos sutrikimo klasifikavimo tikslumas ir taiklumas, identifikuojant lengvą ir sunkų Parkinsono ligos eigos atvejį [21]

	Klasės taiklumas	Klasės tikslumas
Sunkus PD eisenos sutrikimas	52,4 %	84,6 %
Lengvas PD eisenos sutrikimas	66,7 %	64,0 %
Kontrolinis subjektas	91,7 %	71,0 %

reikšmę formuojant savybių erdvę, bei atliekant klasifikavimą. Sprendimas atliktas, panaudojus pakankamą skaičių subjektų – 23 su Parkinsono ligos diagnoze ir 16 kontroliniai subjektai, kurie neturi jokių neurologinių susirgimų ligos istorijoje. Tyrimo metu taip pat nustatytas tikslas identifiikuoti mažą ir didelį eisenos sutrikimo atvejus. Iš dalyvaujančių Parkinsono subjektų, 11 turi didelį eisenos sutrikimą, 12 turi mažesnį eisenos sutrikimą.

Tyrime iš viso panaudotos 67 normalizuotos laiko srities savybės, kurios yra svarbios motorinei veiklai. Duomenys normalizuoti dėl skirtingų savybių amplitudės srities. Normalizacija yra labai svarbi procedūra, prieš pateikiant duomenis dimensių mažinimo metodui. Taip yra užtikrinama, kad naudojamos savybės turi vienodą įtakos faktorių, projektuojant naują dimensių erdvę [9]. Po dimensių mažinimo metodo pritaikymo, liko tik 11 savybių, kurių bendra variacija yra 80,20 %. Parkinsono ligos eigos sunkumui nustatyti, prie bendros savybių erdvės pridėtos dar papildomos keturios savybės. Sprendimas, pateikti klasifikatoriui dar papildomas keturias dimensijas nėra tiksliai argumentuotas. Nėra nurodyta ar savybės įterptos po dimensių mažinimo ar prieš tai. Jeigu tai atlikta po dimensių mažinimo, tai kelia abejonių naujai sukonstruota erdvė – sistema nėra gerai apibendrinta, jina 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 subendrina savybių erdvę. Klasifikatoriaus tikslumo ir taiklumo rezultatai yra pateikiami 2.3 lentelėje. Geriausias klasifikavimo tikslumas pasiektas, naudojant kintamą klaidos vertinimo kriterijų. Nurodytas klasifikavimo metodas atpažįsta Parkinsono ligą sėkmingai (100 % tikslumas).

Atliekant sudėtingesnę klasifikavimą tarp lengvos eisenos sutrikimo, sunkios eisenos sutrikimo Parkinsono ir kontrolinio subjekto, klasifikavimo tikslumas ir taiklumas gaunamas prastės. Sunkios eisenos sutrikimas atpažįstamas prastai, taiklumas rezultatas yra 52,4 %, lengvos eisenos sutrikimo klasifikavimo taiklumas siekia 66,7 %. Geriausiai atpažįstamas tik kontrolinės eilės subjektas, taiklumas yra 91,7 %.

Aptartas metodas siūlo konkrečią įrangą, algoritmą ir savybes, naudojamas savybių erdvei sudaryti, tačiau užsibrėžtas tikslas pasiektas tik iš dalies – suprojektuota sistema, naudojanti pigius vertikalios pagreičio jutiklius, kuri atpažįsta Parkinsono liga sergančių subjektų eigos sutrikimus, tačiau sistema labai blogai atpažįsta lengvą ir sunkų eisenos sutrikimą. Nurodyta kryptimi reikia atlikti didelį darbo kiekį, norint sėkmingai identifikuoti Parkinsono ligos stadiją ir atitinkamai pateikti HAY skalės vertinimą, analizuojat žmogaus eisenos požymius.

Tolimesnis darbas apžvalgai yra “A ground reaction force artificial neural network classifier for the diagnosis of Parkinson’s disease” [1]. Duomenys analizei panaudoti tokie patys, kokie naudojami ir šiame darbe, iš tos pačios duomenų bazės, tačiau pagrindinis nagrinėjamo darbo skirtumas yra naudojamų subjektų skaičius – analizei pasirinkta 40 subjektų, turinčių Parkinsono ligą ir 40 subjektų, neturinčių neurologinių sutrikimų. Nėra argumentuota kodėl toks subjektų skaičius panaudotas analizei, kadangi duomenų bazėje yra 93 subjektai su Parkinsono ligos sutrikimais ir 73 kontroliniai, sveiki subjektai. Iš atliktos pirminės analizės 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 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 atžvilgiu (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 [7]. Dimensijų mažinimo metodas nepritaikytas. Klasifikavimui pasirinktas dirbtinių neuronų tinklas su 8 paslėptais neuronais, apmokymas vyko naudojantis porinio mastelio gradiento atvirkštinį dauginimą (angl. scaled conjugate gradient backpropagation algorithm). Apmokymui pasirinkta po 20 subjektų iš kiekvienos grupės, dirbtinių neuronų tinklo tikrinimas vykdytas panaudojus po 10 subjektų iš kiekvienos grupės, testavimas atliekamas panaudojus po 10 subjektų duomenis iš kiekvienos grupės. Klasifikavimo taiklumas ir tikslumas yra pateikti 2.5 lentelėje. Klasifikatoriaus tikslumas yra 95 %, kontroliniai subjektai atpažinti su 90 % taiklumu (vienas iš dešimt subjektų atpažinti klaidingai), Parkinsono liga sergantys subjektai atpažinti su 100 % taiklumu, visi subjektai atpažinti teisingai.

Atliktas tyrimas pateikia gerus klasifikavimo rezultatus. Metodo trūkumas yra naudojamos dirbtinių neuronų klasifikatoriaus. Jis neleidžia išsiaiškinti kaip būtent yra apibendrintos naudojamos savybės savybių erdvėje. Taip pat nepavaizduota kaip esami duomenys pasiskirstę savybių

2.5 lentelė Klasifikavimo taiklumas ir tikslumas, panaudojus dirbtinių neuronų tinklo klasifikatorių [1]

	C1	C2
Tikslumas	0,950	0,950
Taiklumas	0,900	1,000

2.6 lentelė Pagrindiniai eisenos parametrai remiantis atlikta analize [12]

Žingsnio ypatybė	Kontrolinis subjektas	Parkinsono subjektas
	Vidurkis \pm nukrypimas	Vidurkis \pm nukrypimas
Žingsnio ilgis (m)	0,550 \pm 0,080	0,420 \pm 0,200
Eisenos greitis (m/s)	0,920 \pm 0,160	0,650 \pm 0,330

erdvėje. Kadangi naudojamos 8 savybės (kas apibrėžia ir savybių erdvės dimensijų skaičių), galima panaudoti dimensijų mažinimo algoritmą ir pateikti esamus rezultatus grafiniu pavidalu, vizualinei analizei. 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.

Penktas ir paskutinis darbas apžvalgai yra gilesnės statistinės eisenos analizės darbas, “Statistical Analysis of Parkinson Disease Gait Classification using Artificial Neural Network” [12]. Darbe išnagrinėtos trys grupės eisenos savybių: pagrindinės, kinetinės ir kinematinės. Pagrindinės savybės: žingsnio laikas, žingsnių skaičius per minutę, žingsnio ilgis, eisenos greitis. Kinetinės savybės: maksimalus pėdos kontaktas su žeme pagal vertikale; maksimalus pėdos kontaktas su žeme pėdos pakilimo metu pagal vertikale; minimali jėga tarp dviejų kojos prisilietimo prie žemės maksimumų pagal vertikale; maksimali pėdos jėga pagal horizontale, kuomet pėda prisiliečia prie žemės; maksimali pėdos jėga pagal horizontale, kuomet koja pakyla nuo žemės. Kinematinės savybės: kulکشnies kampas, kuomet pėda prisiliečia prie žemės; kelio kampas, kuomet pėda prisiliečia prie žemės; klubo kampas kuomet koja prisiliečia prie žemės; maksimalus kulکشnies, kelio ir klubo kampas ėjimo metu. Duomenys išskirti tokiomis savybėmis surinkti pasitelkus slėgiui jautrų paviršių, kurio diskretizavimo dažnis yra 200 Hz ir 37 markerių, kurie buvo pritvirtinti prie subjektų kojų ir stebėti infraraudonųjų spindulių kamerų. Visi duomenys apdorojami *Vicon*[®] programinio paketo. Norint išvengti slėgiui jautriam paviršiumi surenkamų duomenų priklausomybės nuo subjektų svorio, duomenys normalizuojami pagal svorį:

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

2.7 lentelė Dirbtinių neuronų tinklo klasifikavimo rezultatai, panaudojus skirtingas savybių grupes [12]

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 %

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

Darbo tikslas yra pirmiausiai analitiškai išanalizuoti visus gaunamus duomenis ir padaryti hipotezę, kokios eisenos savybės geriausiai identifikuoja subjektų grupes. Vėliau, pagal klasifikatoriaus tikslumo ir taiklumo rezultatus, parinkti geriausius požymius, pagal kuriuos vyks subjektų atpažinimas. Pagrindiniai eisenos parametrai, kurie labiausiai skyrėsi tarp subjektų, yra pateikti 2.6 lentelėje. Pastebėta, kad kontrolinių subjektų žingsnio ilgis yra ilgesnis už Parkinsono subjektų žingsnio ilgį. Iš pagrindinių savybių išskirtas ėjimo greitis. Parkinsono subjektų eisenos greitis pažymėtas kaip lėtesnis, lyginant su kontroliniais subjektais. Jėgos jutiklių parametrų palyginimas parodė, kad daugiausiai informacijos pateikia minimumas, kuris yra tarp pėdos pirmo prisilietimo prie žemės maksimumo ir pėdos pakilimo nuo žemės maksimumo. Iš kinematinės savybių, didžiausias skirtumas tarp subjektų yra klubo maksimalus kampas: $19,904 \pm 8,623$ laipsnių kontrolinių subjektų atveju ir $11,271 \pm 6,157$ laipsnių Parkinsono subjektų atveju.

Atlikus analitinį palyginimą, toliau atliktas savybių tikrinimas klasifikatoriaus taiklumo rezultatais. Rezultatai yra pateikiami 2.7 lentelėje. Pagrindinių ir kinetinių savybių grupės surinko po vienodą klasifikavimo taiklumo rezultatą. Tai leidžia daryti išvadą, kad abi šios savybių grupės rodo panašias eisenos savybes. Kinematinės savybių taiklumo rezultatas yra 84,38 % ir yra didesnis už dviejų savybių grupių taiklumą. Sujungus pagrindines ir kinetines savybių grupes pasiektas didesnis klasifikavimo taiklumo rezultatas 87,50 %, tokį patį rezultatą pateikė ir pagrindinių su kinematinėmis savybėmis sujungimas. Kiti galimi savybių grupių sujungimai nepateikė geresnių klasifikavimo taiklumo 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. Klasifikavimo taiklumas 95,63 %.

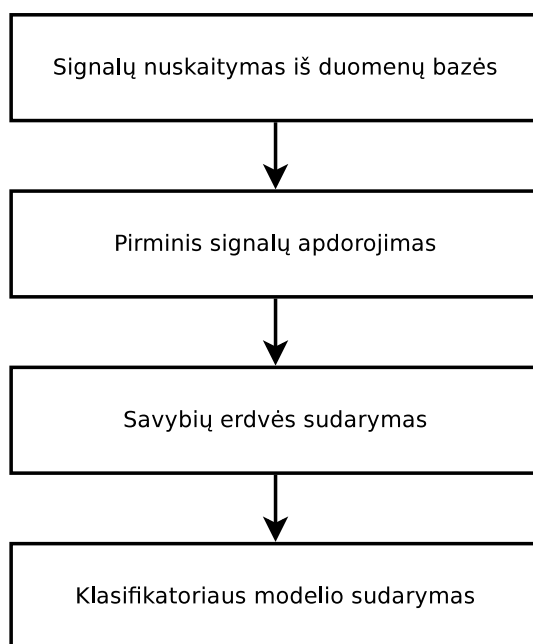
Iš aptarto darbo, šiame darbe yra panaudotos savybės savybių erdvei sudaryti, bei pritaikytas dirbtinių neuronų tinklas, klasifikavimui atlikti.

3. Signaļu analizēs programos kūrimas

Šiame skyriuje apžvelgtos kylančios problemos kuriant signalų analizės programą, pateikti galimi jų sprendimo būdai. Skyrius prasidės bendros programos struktūrinės schemos sudarymo poskyriu. Jame pateikta bendra programos veikimo struktūra, nurodytas funkcinių modulių skaičius, kiekvieno nurodyto modulio paskirtis. Toliau aptarti galimi pirminio signalo apdorojimo etapai. Tai yra vienas iš svarbiausių programos žingsnių, kadangi nuo pirminio signalo apdorojimo kokybės priklauso ir visos programos rezultatas. Požymių išskyrimo poskyryje atliktas signalo savybių nagrinėjimas, signalo savybių analizė. Nagrinėjama savybių erdvė yra naudojama klasifikavimo metu, todėl būtina rasti tokias savybes, kurios skiriasi tarp kontrolinio ir Parkinsono subjekto. Požymio klasifikavimo programos kūrimo metu aptarti galimi naudoti klasifikatoriai, kiekvienam iš jų pateikti vienodi duomenys klasifikavimo užduočiai atlikti ir nuspręsta koks klasifikatorius geriausiai tinka nagrinėjamų savybių erdvei. Galiausiai, duomenų analizės programos kūrimo poskyryje pateiktas geriausiai užduotį atliekantis pirminis signalų apdorojimo metodas, dimensių 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 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 aukščiau einančio bloko, kas reiškia, kad “Pirminio signalų apdorojimo” bloko algoritmo pakeitimas



3.1 pav. Pirminė programos schema



neturi turėti jokios įtakos po jo einančiam “Savybių erdvės sudarymas” blokui.

Kiekvieno bloko užduotis:

1. “Signalų nuskaitymas iš duomenų bazės” bloko užduotis yra nuskaityti duomenis iš naudojamos “PhysioBank” duomenų bazės ir juos įkelti 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: pėdos prisilietimas prie žemės signalas ir pėdos pakilimo nuo žemės signalas. 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 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 modelio sudarymas” bloko paskirtis yra klasifikatoriaus apmokymas ir jo testavimas. Klasifikatoriaus pasirinkimas yra labai svarbus klausimas darbe. Jis turi būti parinktas argumentuotai, išanalizavus kiekvieno naudojamo klasifikatoriaus taiklumo ir tikslumo rezultatus. Svarbus klasifikatoriaus aspektas yra savybių erdvės bendro dėsnio radimas arba erdvės aproksimacija. Šį faktą galima apibrėžti iš klasifikatoriaus rezultatų grafiko. Jeigu ilgą laiko tarpą į klasifikatorių yra siunčiami vienos klasės duomenys, o klasifikavimo rezultatai yra nestabilūs, ilgą laiką neturintis pastovaus rezultato, vadinasi, galima teigti, kad klasifikatorius blogai atliko erdvės duomenų pasiskirstymo dėsnio aproksimavimą, jis nėra tinkamas nagrinėjamai savybių erdvei.

Tolimesniuose poskyriuose aptarti įgyvendinti blokinės struktūros elementai.

3.2. Pirminio signalų apdorojimo programos kūrimas

Šiame skyriuje 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 gautas signalas.

Programos kūrimo patogumui, parinkta direktorių architektūra:

- Pagrindinė direktorija



3.1 programinis kodas. Duomenų nuskaitymo funkcija iš tekstinės duomenų bylos

```
function data = read_data( file_name )
%READ_DATA Function reads the data from file
%
%   Input:
%       - file_name the name of the file
%
%   Output:
%       - data the Left foot data

database_directory = '../database/';

% Check if file exists
if ~exist( [ database_directory file_name ], 'file' )
    % Problem arise
    error( [ 'File_' file_name '_does_not_exist' ] );
end

data_raw = dlmread( [ database_directory file_name ] );

% We'r interested only in left foot data
data = data_raw(:,19);

end
```

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

Kadangi programos kūrimo metu yra svarbu saugoti ankstesnes programos versijas, pasirinktas kasdienis programos versijos saugojimas: kiekvienos pradžioje darbas pradedamas su ankstesnės dienos kopija. Taip buvo išsaugotas kiekvienos darbo dienos programos versija ir taip galima peržiūrėti kokių analitinių kelių 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 [5], 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, 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.



3.2 programinis kodas. Slankiojančio lango algoritmo pritaikymas

```
function output = split_data( input, window_length, window_overlay )
%SPLIT_DATA Function to split data into windows

% Buffer to store the signal information
buffer = [];
output = [];

for i=1:length(input)
    % if buffer reached the length of the window
    if ( length(buffer) >= window_length )
        % store the buffer to output signal
        output = [ output; buffer ]; %%ok<AGROW>
        % Do the overlay step
        if ( window_overlay ~= 0 )
            buffer = buffer(window_overlay:end);
        end
    else
        % otherwise, populate the buffer with new entry
        buffer = [ buffer input(i) ]; %%ok<AGROW>
    end
end

end
```

3.3 programinis kodas. Signalų filtravimas dviem Butterworth filtrais

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

Sekantis žingsnis, po duomenų nuskaitymo, yra jų pirminis apdorojimas. Nagrinėjimas 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 panaudotos formuojant naują signalą. Tokia funkcijos savybė labai naudinga tolimesniame darbe.

Sekanti programos dalis atlieka paprastą signalo filtravimą su dvejais *Butterworth* skaitme-

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

```

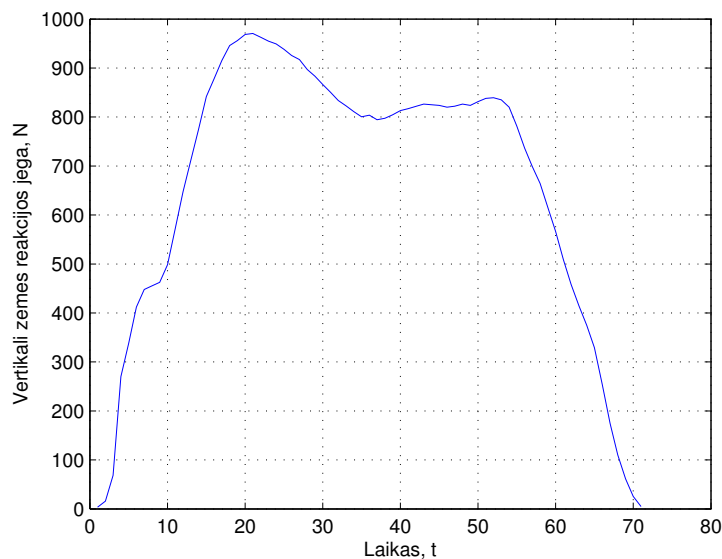
for i=1:size( Pt_t, 2 )
    fprintf( ' ' );
    signal = Pt_t{i};
    buffer = [];
    k = 0;
    for j=1:size(signal, 1)
        if ( signal(j) ~= 0 )
            buffer = [ buffer signal(j) ]; %%ok<AGROW>
        else
            if ~isempty(buffer) && length(buffer) > 10 && length(buffer) < 200
                % Store the buffer
                data.Pt{ length(data.Pt)+1 } = buffer;
                % Clear the buffer
                if ( j - k > 50 && j - k < 200 )
                    data.Pt2( length(data.Pt2)+1 ) = j - k;
                end
                k = j; % Start of stance
                buffer = [];
            else
                buffer = [];
            end
        end
    end
end
end
end

```

niniais filtrais. Pirmasis, aukštų dažnių filtras, skirtas pašalinti signalo nuolatinei komponentei. Antras, žemų dažnių filtras, skirtas pašalinti aukšto dažnio triukšmą, kuris neneša visiškai jokios 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 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 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. Jei-

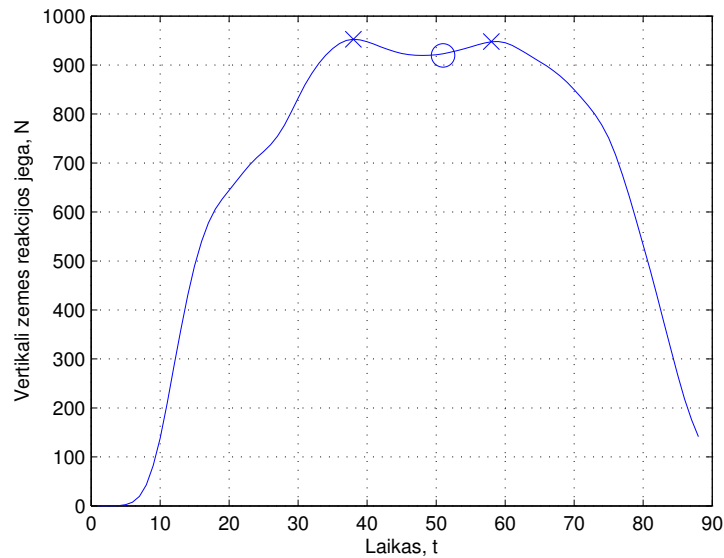


3.2 pav. Susilietimo su žeme signalas

gu signalas yra trumpesnis už 10 verčių, arba turint omenyje, kad signalas diskretizuojamas 100 Hz , tai $0,1\text{ s}$, 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 nebuvo pakelta. 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 $0,5\text{ s}$ 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 [12]. Algoritmo rezultatas yra pateiktas 3.3 pav. Kryžiais pažymėtos globalaus maksimumo vietos, apskritimu pažymėta lokalaus minimumo signalo vieta. Išanalizavus daugumos subjektų žingsnio signalus, padaryta išvada, kad kiekviename žmogaus žingsnyje egzistuoja du maksimumai. Vienas maksimumas randamas, kai subjektas yra visiškai atsiremęs galine pėdos dalimi į žemę, antras maksimumas randamas, kai subjektas visiškai atsiremia priekine pėdos dalimi. Tarp šių dviejų maksimumų yra pereinamasis laikotarpis, kuris yra signalo lokalus minimumas. Priežastis, dėl kurios algoritmas įgyvendintas, yra vienas darbas [12], kuriame pasiūlyta naudotis būtent tokiomis savybėmis atpažinti Parkinsono liga sergančius subjektus nuo nesergančių subjektų. Pačio algoritmo kodo dalis yra pateikta 3.5 pav.

Programos kodas kaupia gaunamą signalą į laikinąją atmintį ir tikrina ar signalas pakito



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

```
for i=1:size(signal_window, 2)
    buffer = [ buffer signal_window(i) ]; %%ok<AGROW>
    if length(buffer) > 2
        delta = buffer(i) - buffer(i-1);
        if delta < d_pr
            % Falls
            buffer(i) = buffer(i-1);
        end

        if buffer(i) == buffer(i-1) && right_index == 0
            && right_max == 0 && buffer(i) > min_value
                right_index = i-1;
                % Return the maximum in the buffer
                right_max = max(buffer);
            end
        end
    end
end
```

per užduotą dydį δ . Delta nurodo kiek signalas turi pakisti, kad algoritmas nustatytų signalo mažėjimo pradžią. Tokio tikrinimo priežastis yra signalo kilimo sumažėjimas prieš globalų maksimumą. Kai kuriuose pavyzdžiuose buvo net pastebėtas signalo mažėjimas. Dėl šios priežasties buvo įvesta pokyčio tikrinimo metodas.

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 yra parinkti koreliacijos koeficientai ar dažninės komponentės – požymio vektorius pakis, todėl prieš kiekvieną požymių analizavimo ciklą yra pateikta ir pirminiam signalų apdorojimo bloke naudojami metodai. Taip pat, kadangi požymių išskyrimo programos kūrimas taip pat sprendžia ir dimensių klausimą – skyriaus pradžioje 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. Metodas taip pat taikomas, kuomet norima sumažinti dimensių skaičių. Problema iškyla, kuomet norima įgyvendinti sukurta 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 reikalavimų. 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 pradėtas nuo paprasčiausios iš dviejų – PCA.

Principinė komponentių analizė [18] 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 taikomas ir šiame darbe. PCA įgyvendinimas naudojant SVD ir tikrinių vektorių skaidymą yra pateiktas priede. Toliau seka PCA matematinis nagrinėjimas.

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. Matlab aplinkoje

Linijinė diskriminanto analizė [22] 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 [8]. 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} nėra 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 $x, x' \in \mathcal{X}$:

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

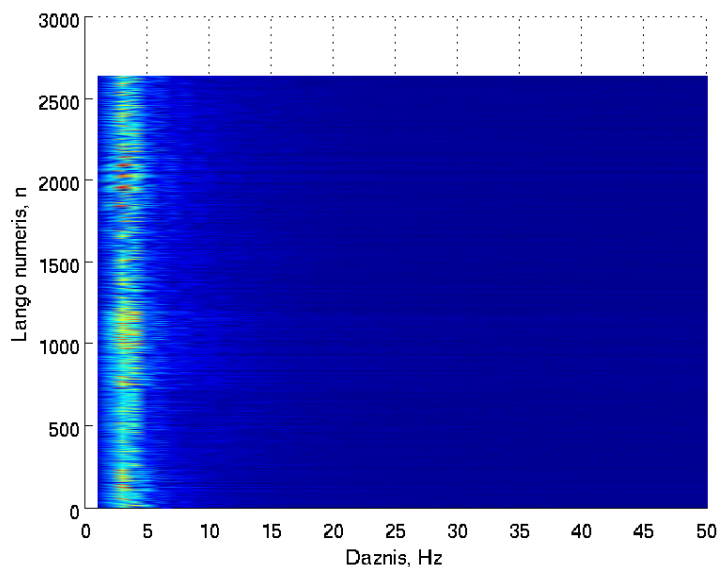
kur Φ projektuota taškus į Hilberto plokštumą \mathcal{H} , kai kada vadinama savybių erdvė. Vieno dumo matmuo k dažniausiai vadinamas branduoliu, o Φ vadinama savybių projekcija.

3.3.2. Savybių tyrimas

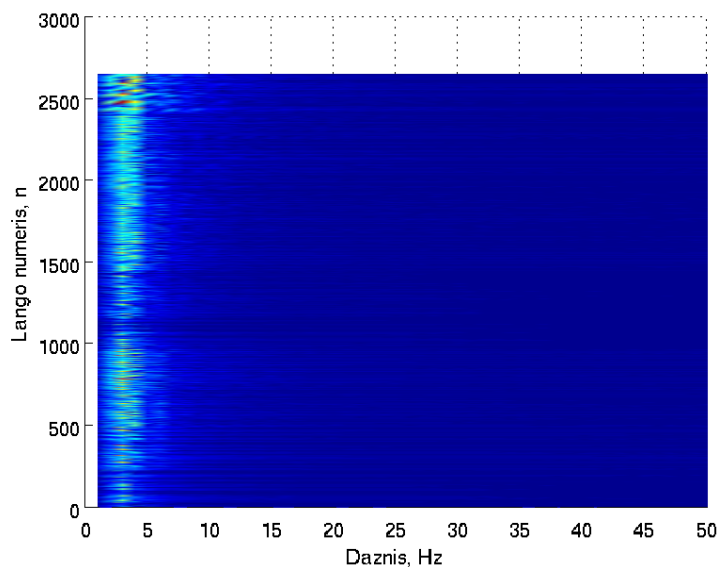
Šiame skyriuje išnagrinėtos galimos signalų savybės, pagal kurias galima atskirti Parkinsono liga sergantį subjektą nuo sveiko subjekto. Kaip minėta anksčiau, ne visos savybės yra galimos išskirti, naudojantis vienu pirminio signalo apdorojimo mechanizmu, todėl kiekvienos savybės nagrinėjimo pradžioje yra 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 statesnius šlaitus, kas iš signalų apdorojimo srities reiškia, kad signalas turi turėti aukštas dažnines komponentes. Šiai savybei išskirti pirminiam signalo apdorojimo bloke panaudotas slankiojančio lango metodas. Taip kiekvienas signalas turės fiksuotą ilgį ir taip galima lyginti signalus tarpusavyje.

Toliau seka išanalizuotos 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 ruožu. Ties 0 Hz dažninių komponentių nėra, kadangi jos 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



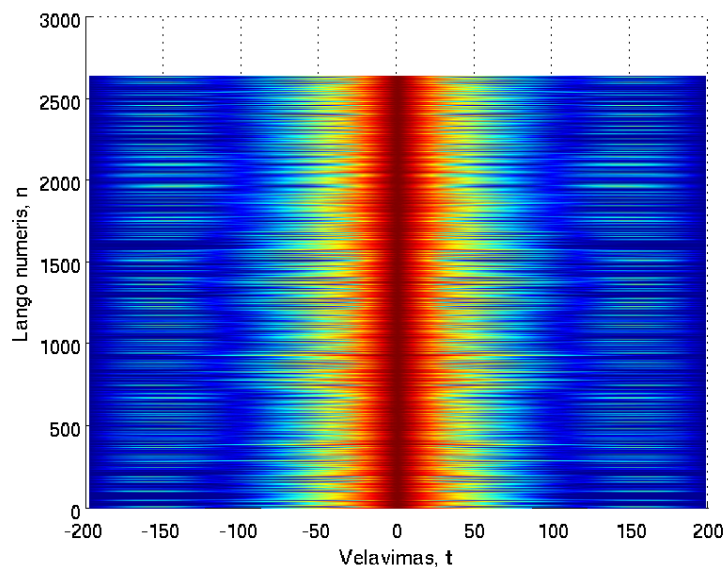
3.4 pav. Kontrolinių subjektų kairės kojos žingsnių signalų dažninės komponentės



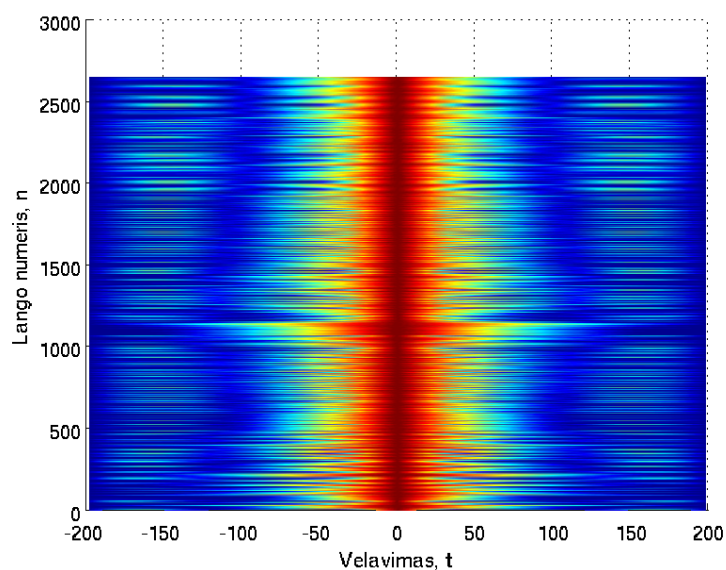
3.5 pav. Parkinsono liga sergančių subjektų kairės kojos žingsnių signalų dažninės komponentės

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 tikrasis kairės kojos signalai, galima taikyti tik savi-koreliacijos koeficientus. Narinėjama savybė parodė efektyviai identifikavo subjektus, remiantis eisenos duomenimis, ankstesniame tyrime, kuriame, remiantis tiesinio pagreičio ir kampinio pagreičio jutiklių parodymais, reikėjo suprojektuoti algoritmą, gebantį atskirti tokias žmogaus



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

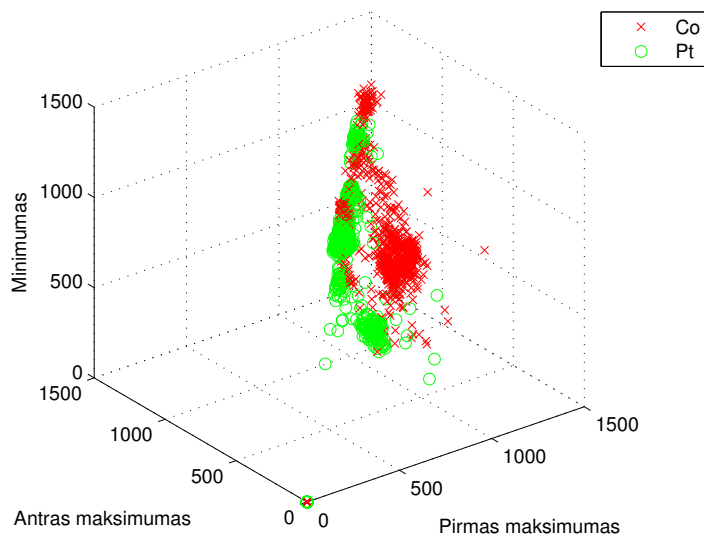


3.7 pav. Parkinsono liga sergančių subjektų kairės kojos žingsnių signalų savi-koreliacijos koeficientai

veiklas: stovėjimas, ėjimas, ėjimas aukštyn laiptais, 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

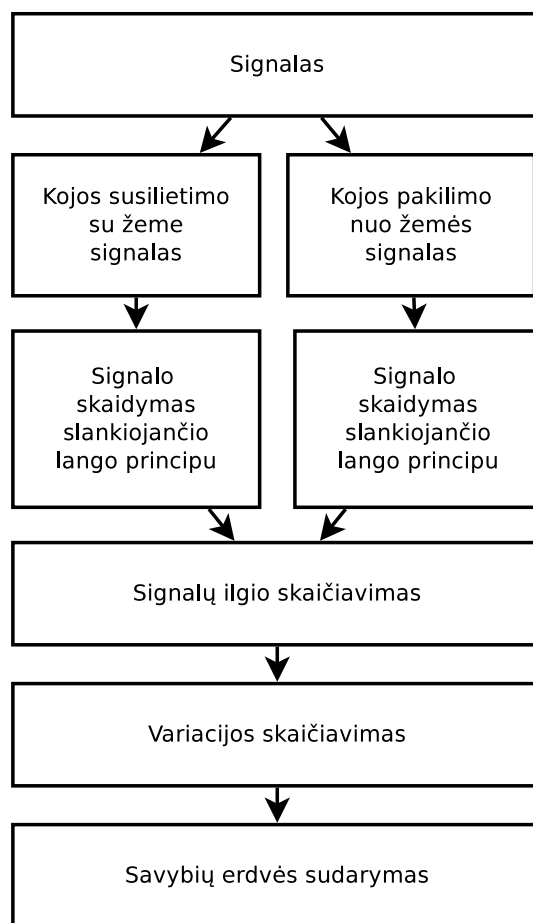
erdvės. Šios tris savybės sudaro trijų dimensių plokštumą, kurią galima lengvai pavaizduoti. Pilna savybių erdvė pavaizduota 3.8 pav. Kaip matosi iš duotos erdvės, duomenys neturi jokio koncentracijos centro. Erdvėje jie pasiskirstę pagal nežinomą dėsnį.

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

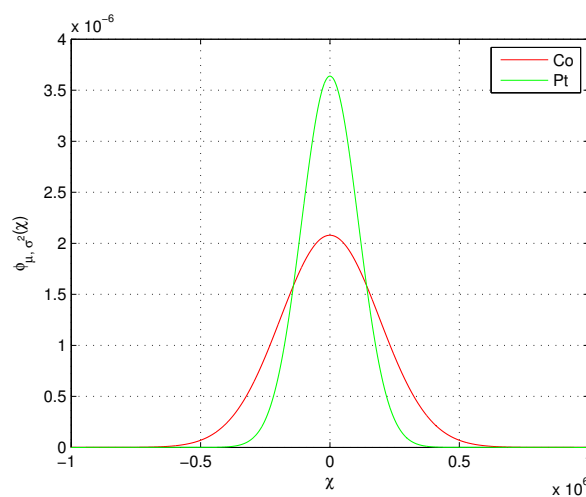
Sekančios savybės nagrinėjimui yra siūlomos atliktų tyrimų [5, 11, 4]. Nagrinėjama savybė yra žingsnio fazės laiko variacija. Kojos susilietimo su žeme laiko variacija ir kojos pakilimo nuo žemės laiko variacija.

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 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 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

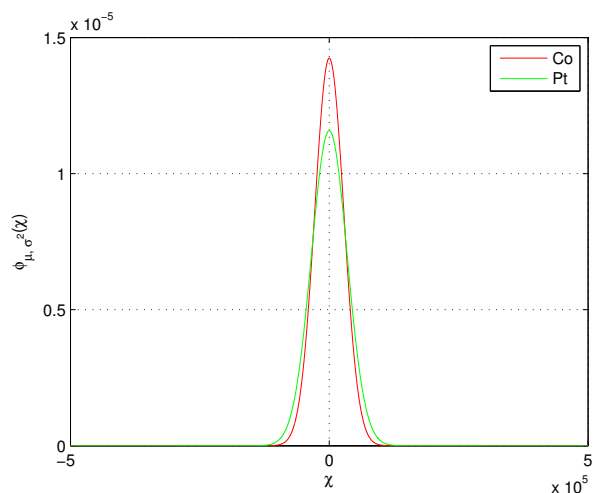


3.9 pav. Pirminis signalo apdorojimas, išskiriant kojos susilietimo su žeme laiko ir kojos pakilimo nuo žemės laiko variacijos savybes

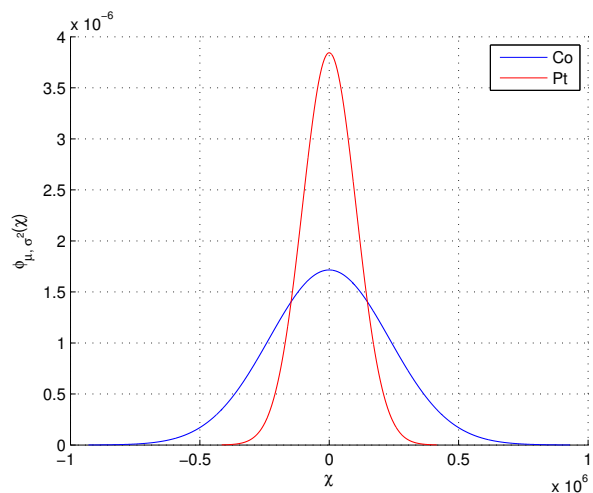


3.10 pav. Kojos prisilietimo prie žemės laiko variacija

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



3.11 pav. Kojos pakilimo nuo žemės laiko variacija

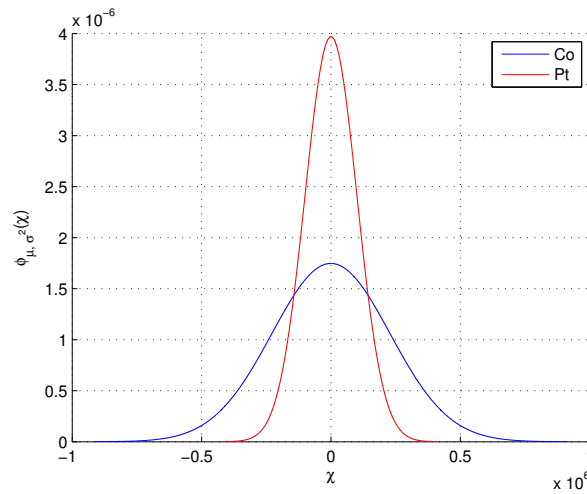


3.12 pav. Kojos prisilietimo prie žemės ir pakilimo nuo žemės laiko variacijos pasiskirstymas po PCA transformacijos

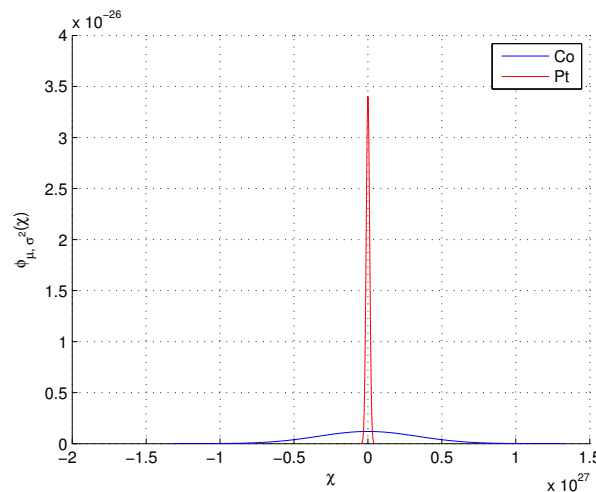
- PCA;
- LDA;
- Kernel PCA (su daugianariu, Gauso branduoliais);
- Kernel LDA (su daugianariu, Gauso branduoliais);

Vienmatė linijinė PCA transformacija pavaizduota 3.12 pav. Vizualiai įvertinus gaunamą grafiką – vidurkis nepasikeitė, tačiau pakito variacija. Vidurkio pokyčio nėra pastebima, todėl teikti duomenis klasifikavimui nėra prasmės, kadangi to pačio vidurkio duomenis atskirti nėra įmanoma.

Sekantis metodas yra LDA. Transformacijos rezultatas yra pavaizduotas 3.13 pav. Vizualiai įvertinus gaunamą grafiką – tiek po LDA, tiek po PCA duomenų pasiskirstymas nėra gerai



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

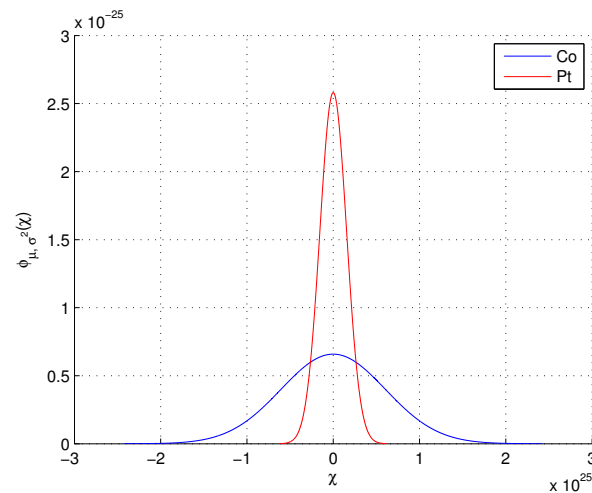


3.14 pav. Kojos prisilietimo prie žemės ir pakilimo nuo žemės laiko variacijos pasiskirstymas po PCA transformacijos, naudojant daugianarį branduolį

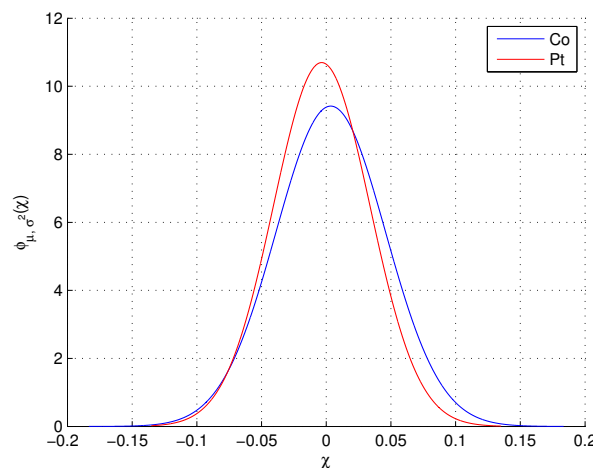
atskirtas. Galima daryti hipotezę, kad linijinis duomenų atskyrimas šiuo atveju naudoti nėra tinkamas. Reikia papildyti esamus algoritmus branduolio metodu ir papildyti transformacijas daugianariu arba Gauso branduoliu.

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. Daugianaris branduolys abiem atvejais panaudotas su 4 branduolio argumentu.

Vizualiai įvertinus gaunamą grafiką, panaudojus polinimonį 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 daugianarį branduolį



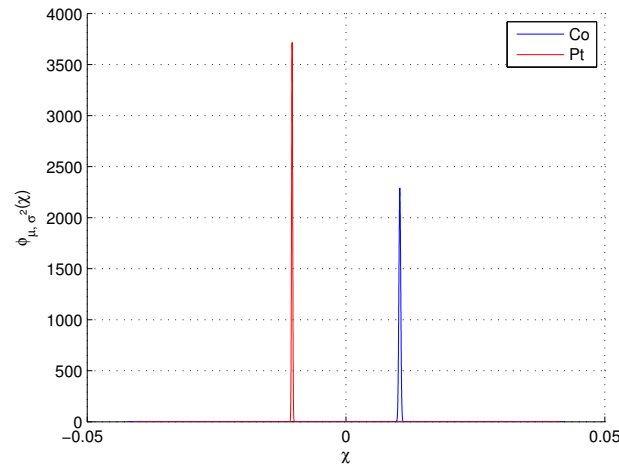
3.16 pav. Kojos prisilietimo prie žemės ir pakilimo nuo žemės laiko variacijos pasiskirstymas po PCA transformacijos, naudojant Gauso branduolį

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

3.4. Požymių klasifikavimo programos kūrimas

Šiame skyriuje išnagrinėti ir pritaikyti populiariausi šiuo metu naudojami klasifikatorių metodai. Tokie klasifikatoriai yra [9]:

- 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 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 atliekamos remiantis tokia schema: iš turimų duomenų išskirta kojos prisilietimo ir pakilimo nuo žemės laiko ilgio variacijos savybė. Visi duomenys padalinti į lygias tris dalis: dimensijų operacijai atlikti duota 500 reikšmių, klasifikatoriaus apmokymui tolimesnės eilės 500 reikšmių, klasifikatoriaus testavimui tolimesnės eilės 500 reikšmių.

3.1 lentelė Vektoriaus palaikymo mašinos tikslumas ir taiklumas

	Co	Pt
Tikslumas	0,489	0,489
Taiklumas	0,234	0,744

Klasifikavimas atliekamas realiu laiku, t.y. klasifikatoriui pateikiami duomenis apie žingsnio stadijos variaciją po transformacijos ir klasifikatorius pateikia savo spėjimą.

Vektoriaus palaikymo mašina [2] šiuo metu yra populiariausias klasifikatorius nagrinėjant netiesiškai atskiriamus duomenis. Igyvendinimas Matlab aplinkoje panaudotas iš [3]. Klasifikatoriaus testavimo metu naudojami tokie parametrai:

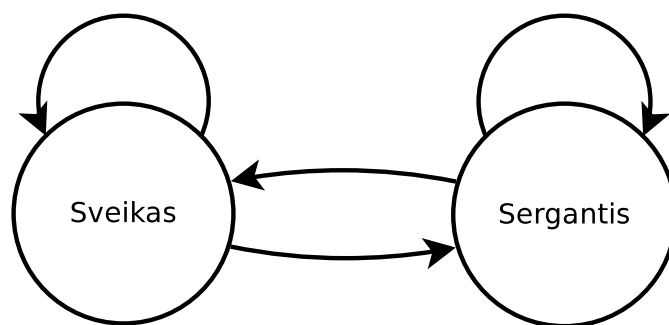
- SVM tipas – Sequential Minimal Optimization;
- Branduolio tipas – linijinis;
- Sureguliuojamoji konstanta – 2;
- Branduolio argumentas – 2;

Lentelėje 3.1 pateikti SVM tikslumo ir taiklumo duomenis. Kaip matosi, klasifikatorius veikia nepakankamai gerai – tikslumo koeficientas nėra didesnis negu pusę, abiem atvejams 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 [16] 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 kita duomenų grupė. Tokia klasifikatoriaus savybė pritaikyta ankstesniame darbe, sudarant žmogaus eisenos atpažinimo sprendimą [15]. Pasirinktas modelis, susidedantis iš dviejų būsenų – “Sveikas”, “Sergantis”. Būsenos tarpusavyje yra sujungtos (3.18 pav.). Perėjimo tikimybės tarp modelio elementų parinktos žymiai mažesnės už tikimybę likti toje pačioje būsenoje. Įgyvendinimas Matlab aplinkoje panaudotas iš [10].

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 [17] 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.



3.18 pav. Ergodinis paslėptas Markovo modelis

3.2 lentelė Paslėpto Markovo modelio tikslumas ir taiklumas

	Co	Pt
Tikslumas	0,500	0,500
Taiklumas	0,000	1,000

3.3 lentelė Naivaus Bayes klasifikatoriaus tikslumas ir taiklumas

	Co	Pt
Tikslumas	0,508	0,508
Taiklumas	0,764	0,252

3.4 lentelė Tiesioginio sklidimo dirbtinių neuronų tinklų tikslumas ir taiklumas

	Co	Pt
Tikslumas	0,282	0,287
Taiklumas	0,000	0,392

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 pritaikytas turimiems duomenims – tiesioginio sklidimo neuronų tinklas. Tai yra dirbtinių neuronų tinklų klasifikatorius, kuris veikia panašiai kaip ir Naivus Bayes – jis ieško funkcijos (linijinės arba daugianarės), kuri geriausiai atskiria turimus duomenis. Bandymo metu 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 tikslumas ir taiklumas nėra 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 aprašytas sekančiame poskyryje.

3.5. Duomenų analizės programos kūrimas

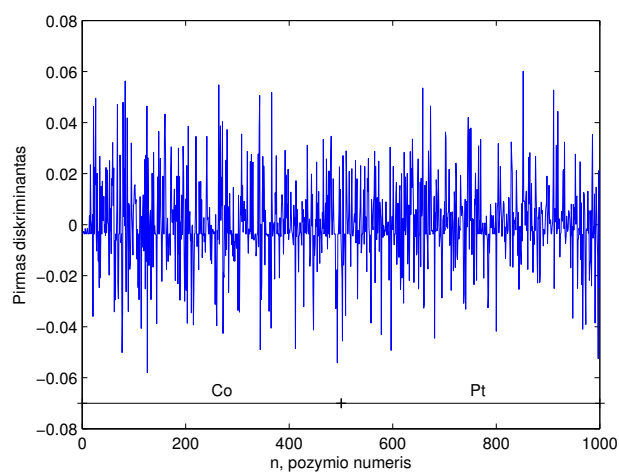
Šiame poskyryje aptartas duomenų analizės programos kūrimas. Pasinaudojus prieš tai esančių skyrių informacija yra pateiktas sprendimas, kuris leidžia efektyviai kiek įmanoma efektyviau atpažinti subjektų tipus.

Poskyryje “Požymių išskyrimo programos kūrimas” 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 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 diskriminant. 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 chaotiški. Iš vaizdinės analizės nėra įmanoma spėti kuri signalo dalis kuriai subjektų grupei priklauso.

Tokios išvados verčia projektuoti kitą klasifikavimo mechanizmą, kuris duomenis klasifikuo-tų ne kiekvieną pagal kiekvieną nagrinėjamą savybę, o pagal savybės grupę – į laikiną atmintį yra rašoma požymio reikšmės, 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), yra panaudotas paprasčiausias naivaus Bayes klasifikatoriaus mechanizmas. Duomenys nesikartojo jokiame algoritmo modulyje. Viso erdvės sudarymui panaudota 400 (nuo 1 iki 400), apmokymui 500 (nuo 501 iki 1000), testavimui 500 (1001 iki 1500) reikšmių. Laikinos atminties ilgis pasirinktas 50 verčių, iš kurių skaičiuojamas vidurkis. Apmokymo metu iš viso 10 vidurkiai kiekvienai subjektų grupei, testavimo metu iš



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

3.5 lentelė Klasifikavimo rezultatas, naudojant naivų Bayes klasifikatorių

	Co	Pt
Tikslumas	0,800	0,800
Taiklumas	0,714	1,000

viso 10 vidurkiai kiekvienai subjektų grupei. Gauti tikslumo ir taiklumo rezultatai pateikti 3.5 lentelėje.

Gauti klasifikavimo tikslumas yra 0,800, kas viršija prieš tai naudotų metodų tikslumą. Pirmos grupės klasifikavimo taiklumas yra 0,714, antros grupės klasifikavimo taiklumas 1,000. Detalesnė klasifikavimo mechanizmo veikimo apžvalga yra pateikta Požymių klasifikavimo programos kūrimo poskyryje.



4. Signalų analizės programos įgyvendinimas

Šiame skyriuje apžvelgta įgyvendintas programinis kodas, jo veikimo architektūra. Ankstesniuose poskyriuose argumentuotai pažvelgti galimi analizės metodai, požymiai ir klasifikatoriai. Visi rezultatai panaudoti projektuojant galutinį sprendimą. poskyryje 4.1. pateikta bendra programos algoritmo veikimo schema, poskyryje 4.2. pateikta algoritmo klasifikavimo veikimo schema, poskyryje 4.3. apžvelgta galutinė programa.

4.1. Bendro programos algoritmo schemos sudarymas

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

Viso programa sudaryta iš trijų modulių:

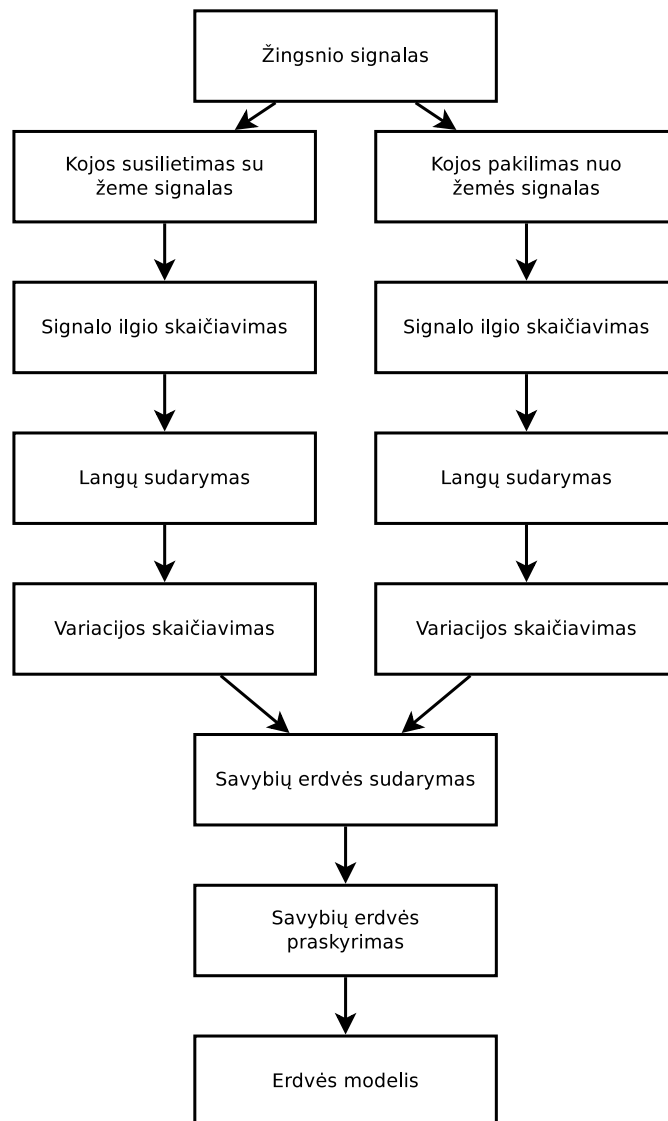
- Savybių erdvės sudarymas;
- Klasifikatoriaus apmokymas;
- Klasifikatoriaus tikrinimas.

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 transformaciją 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 naudojama ta pati dimensijų 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 yra 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, savybių erdvės sudarymo, veikimo schema

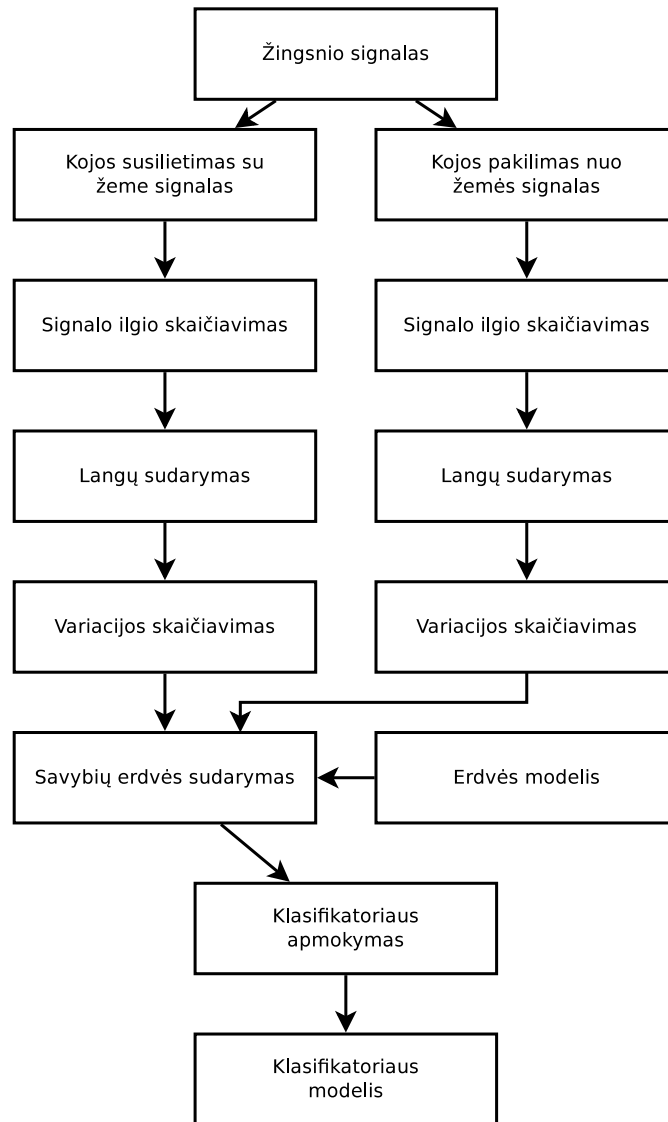
4.2. Požymių klasifikavimo programos algoritmo schemas sudarymas

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

Naivus Bayes klasifikatorius [17] 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, klasifikatoriaus apmokymo, veikimo schema

pakankamai gerai, ir yra naudojamas daugelį sudėtingesnių metodikų.

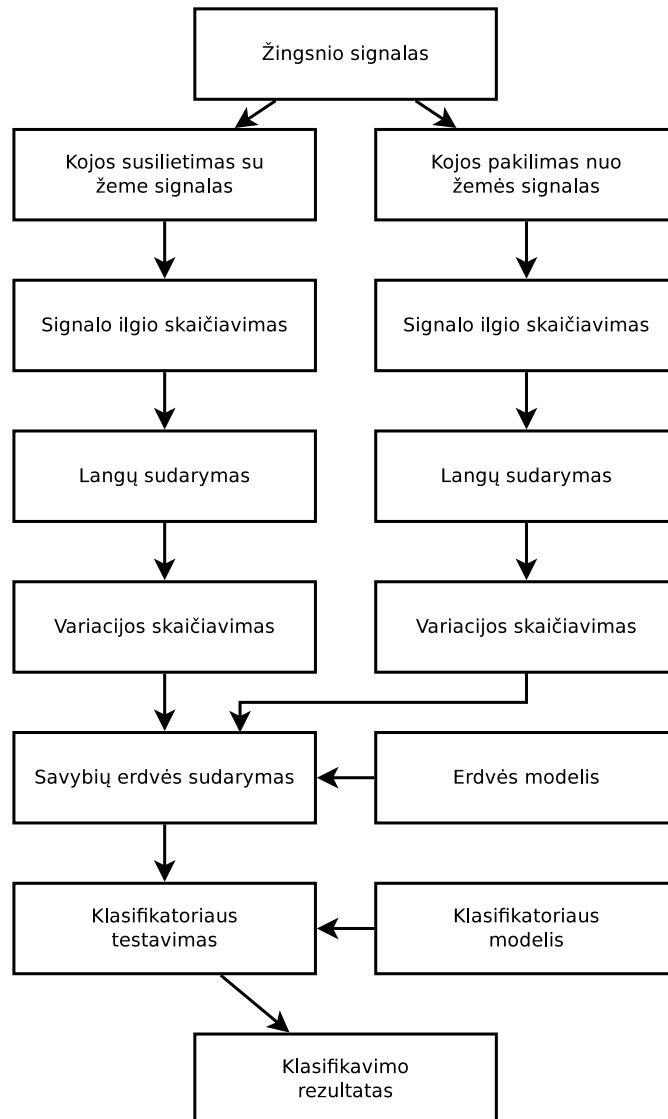
Bayes klasifikatorius $h^*(x)$, naudoja diskriminanto funkcijas atnaujinti klasės vėlesnes tikimybes, nurodant savybių vektorių:

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

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 diskrimi-



4.3 pav. Trečio modulio, klasifikatoriaus tikrinimo, veikimo schema

nato funkcija:

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

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 nelieta prasmės, ieškoti



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 diagnozavimui 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.

5. Signalų analizės programos patikra

Šiame skyriuje parengtas ir įgyvendintas algoritmo patikros planas. Programos patikra yra kritinis aspektas jos patikimumo tikrinimui. Algoritmą galima tikrinti mažinant apmokymo verčių skaičių ir didinant tikrinimo verčių kiekį – taip sužinota kiek mažiausiai verčių reikia metodui, norint pilnai apibendrinti turimus duomenis.

Tikrinant kiekvieno žmogaus žingsnio signalus, paaiškėjo, kad skirtingi subjektai sugeneruoja kitokį tikrinimą atliktų žingsnių skaičių, todėl tikrinimas atliekamas ne atskiriant konkrečius žmones, tačiau jau išskirtus kojos pakilimo ir kojos prisilietimo prie žemės signalus, konkrečiau – iš tų signalų suformuotus jų ilgių duomenų langus.

Po pirminio duomenų apdorojimo, kontrolinių subjektų grupėje liko 1790 duomenų langų, Parkinsono subjektų grupėje liko 1619 duomenų langų, todėl nuspręsta iš kiekvienos grupės pasiimti po 1500 langų duomenų, visi jie padalinti po lygias tris dalis: viena dalis erdvės sudarymui, antra dalis klasifikatoriaus apmokymui, trečia dalis klasifikatoriaus testavimui. Jokie duomenys jokioje dalyje pasikartotinai nesikartoja. Tai yra kritinis faktorius mašiniam apmokyme [9]. Taip pat reikia užtikrinti duomenų praskyrimo ir klasifikatoriaus apmokymui dešimtį kartų didesnių duomenų skaičių, negu yra tų pačių duomenų dimensijų. Kadangi prieš duomenų praskyrimą turima tik $d = 2$ dimensijas, tai teoriškai užtektų ir $n = 21$ ($n/d > 10$), tačiau dimensijų praskyrimas vyksta su 500 taškų, kas tik užtikrina apibendrinimą.

5.1. Eksperimentų plano rengimas

Šiame poskyryje aptarti galimi eksperimentiniai algoritmo patikros planai, nurodytas galimos duomenų pateikimo metodikos.

Pirminį patikros planą sudaro pateikiamų duomenų skaičiaus mažinimas į kiekvieno algoritmo žingsnį: dimensijų erdvės sudarymas, klasifikatoriaus apmokymas, klasifikatoriaus testavimas. Duomenis galima mažinti linijiniu būdu – dimensijų mažinime naudoti pirmus 400 duomenų, 100 praleisti, nuo 501 iki 1000 paduoti klasifikatoriaus apmokymui ir nuo 1001 iki 1500 pateikti klasifikatoriaus tikrinimui. Taip pat duomenis galima pateikti kas kelintą n narį – kadangi iš viso yra trys algoritmo žingsniai, tuomet $n = 3$. Taip į dimensijų mažinimo algoritmą pateikiamas 1 rinkinys, 4 rinkinys, 7 rinkinys. Klasifikatoriaus apmokymui pateikiamas 2 rinkinys, 5 rinkinys, 8 rinkinys. Klasifikatoriaus tikrinimui pateikiamas 3 rinkinys, 6 rinkinys, 8 rinkinys. Toks duomenų pateikimas leidžia pasiekti “apibendrinto” metodo.

Taip pat, kadangi sprendime naudojamas laikinos atminties modulis, kuris reikalingas saugoti pirmąjį diskriminantą kiekvienos variacijos poros, eksperimentiniu būdu galima spręsti koks lango dydis nurodytai užduočiai atlikti geriausiai tinka. Atliekant tokį tikrinimą svarbu palaikyti kitų parametrų vienodumą, todėl į kiekvieną iš trijų modulių pateikiamas pastovus duomenų skaičius, kuomet keičiamas laikinos atminties dydis. Tikrinimas taip pat nuspręstas kiek minimaliai iš paciento eisenos turi būti išskirta savybės grupės, kad sistema gražintų geriausią klasifikavimo tikslumo ir taiklumo rezultata.

5.1 lentelė Klasifikavimo tikslumo ir taiklumo rezultatai, mažinant liniškai pateikiamų duomenų dimensijų mažinimo algoritmui

		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	0,750	0,400	0,462	0,500	0,467

5.2 lentelė Klasifikavimo tikslumo ir taiklumo rezultatai, mažinant liniškai pateikiamų duomenų klasifikatoriaus apmokymo algoritmui

		Taškų rinkinio skaičius					
		500	400	300	200	100	21
Co	Tikslumas	0,800	0,800	0,750	0,750	0,650	–
	Taiklumas	0,875	0,875	0,857	1,000	1,000	–
Pt	Tikslumas	0,800	0,800	0,750	0,750	0,650	–
	Taiklumas	0,750	0,750	0,692	0,667	0,588	–

5.2. Duomenų eksperimentams rengimas

Šiame poskyryje aptartas duomenų eksperimentams rengimas, aprašytas planuojamas duomenų kiekis, naudojamas tikrinimo metu.

Pradžioje yra tikrinamas dimensijų mažinimo algoritmas, mažinant pateikiamų duomenų skaičių. Struktūriškai nuspręsta, kad kiekvienas modulis turi lygiai po 500 rinkinių. Tikrinimas pradedamas nuo 500 rinkinių ir mažinamas kas 100 rinkinių. Vadinasi, iš viso dimensijų mažinimo algoritmui pateikiama 500, 400, 300, 200, 100 ir minimalus 21 rinkinių skaičius. Į apmokymo lygmenį pateikiamas pastovus duomenų skaičius – 500 (nuo 501 iki 1000) rinkinių. Laikinos atminties dydis palaikomas pastovus – 50 verčių. Po dimensijų mažinimo žingsnio tikrinimo, rinkinių skaičius tokia pat metodika mažinamas klasifikatoriaus apmokymui, dimensijų mažinimo metodui pateikiant pastovų rinkinių skaičių. Šioje stadijoje duomenų skaičius nėra mažinamas iki 21 rinkinio skaičiaus, kadangi neformuojamas naudojamas lango ilgio, 50, duomenų skaičius. Klasifikatoriaus testavimo atveju duomenų dydis nėra mažinamas.

Laikinos atminties ilgio tikrinimo metu panaudoti tokie laikinos atminties ilgiai: 10, 30, 50, 70, 100, 130 ir 150. Kiekvienas toks tikinimas atliekamas su duomenų kiekiu, kuris parodė geriausią klasifikavimo tikslumo ir taiklumo rezultatą, aprašytą ankstesniame paragrafe. Panaudoti rezultatai atitinkamai užtikrins optimalų duomenų kiekio pasiskirstymą, geriausiam klasifikavimo tikslumo ir taiklumo rezultatui pasiekti.

5.3. Programos patikros rezultatai

Šiame poskyryje pateikti ir aptarti patikros tikslumo ir taiklumo rezultatai, išanalizuoti galimi sistemos parametrai, kurie gali būti derinami: duomenų skaičius, kuris naudojamas

5.3 lentelė Klasifikavimo tikslumo ir taiklumo rezultatai, mažinant nelineiškai pateikiamų duomenų dimensijų mažinimo algoritmui

		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

5.4 lentelė Klasifikavimo tikslumo ir taiklumo rezultatai, mažinant nelineiškai pateikiamų duomenų klasifikatoriaus apmokymo 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	—

5.5 lentelė Klasifikavimo tikslumo ir taiklumo rezultatai, mažinant laikinosios atminties dydį

		Laikinosios atminties dydis						
		150	130	100	70	50	30	10
Co	Tikslumas	0,500	0,500	0,700	0,500	0,800	0,500	0,570
	Taiklumas	0,500	0,500	0,750	0,500	0,875	0,500	0,585
Pt	Tikslumas	0,500	0,500	0,700	0,500	0,800	0,500	0,570
	Taiklumas	0,500	0,500	0,667	0,500	0,750	0,500	0,559

kiekvienam programos moduliui, bei atminties dydis, kuris naudojamas pirmajai diskriminanto vertei saugoti.

Pirma patikra atlikta linijiškai mažinant duomenų kiekį, kuris yra paduodamas dimensių mažinimo algoritmui, taip siekiant nustatyti koks duomenų kiekis geriausiai tinka naujai savybių erdvei projektuoti. Eksperimentiniai duomenys yra pateikti 5.1 lentelėje. Vertinant dimensių sudarymą pagal klasifikatoriaus duodamą rezultatą, geriausias duomenų kiekis, skirtas naujai dimensių erdvei sudaryti yra 400 erdvės taškų. Tokiu atveju pasiekiamas 0,875 Pt ir 0,750 Co taiklumas.

Antra patikra 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 400. Tokiu atveju pasirenkamas didesnis duomenų kiekis, taip užtikrinant bendrų duomenų apibrėžtumą.

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

Ketvirta patikra 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 atlikta keičiant laikinosios atminties dydžio matmenis. Duomenų erdvei sudaryti panaudoti duomenys, remiantis pirma patikra, klasifikatoriaus apmokymui panaudoti duomenys, remiantis antra patikra. Klasifikavimo rezultatai yra pateikti 5.5 lentelėje. Geriausias laikinosios atminties lango dydis, prie kurio pasiekiamas geriausias klasifikavimo rezultatas yra 50 pirmojo diskriminanto verčių.

Iš pateiktos analizės, 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, 50 eilės.

Patikros metu nustatyta, kad programa geriausiai veikia 80 % tikslumu, kas įveda nepasitikėjimo faktorių, kuris lygus 1/5 visų rezultatų tikslumu. Toks rezultatas yra geras tik tuo



atveju, jeigu aprašytas diagnostikos įrankis panaudotas mažiausiai penkis kartus, norint pateikti galutinę diagnozę, kad pacientas turi arba neturi Parkinsono ligos, pagal eisenos sutrikimo simptomus.

6. Rezultatų apibendrinimas

Darbo metu ištirtos galimos žingsnio savybės, kuriomis remiantis galima sėkmingai atpažinti Parkinsono ligą pagal subjekto eiseną. Nustatyta, kad dažninės žingsnio komponentės, koreliacijos koeficientas, dviejų maksimumų ir vieno minimumo savybės neturi pakankamai informacijos Parkinsono ligos atpažinimui. Daugiausiai informacijos turi kojos prisilietimo prie žemės ir kojos pakilimo nuo žemės signalo laiko variacijos.

Turint duomenis, kurie turi daugiausiai informacijos ligos identifikavimui, toliau patikrinta galimų dimensių priskyrimo metodų pritaikymas. Linijiniai PCA ir LDA transformacijos savybių erdvę duomenys tinkamai nepraskyrė. Geriausiai užduotį LDA su Gauso branduoliu. Rezultatas parodė kaip branduolio metodo pritaikymas gali padidinti dimensių mažinimo algoritmo efektyvumą.

Turimus vienmačius duomenis realiu laiku geriausiai klasifikavo naivus Bayes klasifikatorius, tačiau klasifikavimo tikslumas ir taiklumas nepatenkinamas, tikslumas siekė tik 50,8 %. Pritaikius papildomą metodikos žingsnį – laikinosios atminties bloką, kuriame saugomas pirmas transformacijos diskriminantas, klasifikavimo tikslumas pagerintas iki 80 %. Rezultatas nurodo kas penkto diagnozavimo rezultato klaidingumą. Turint omenyje, kad klinikinės diagnostikos sprendimų taiklumas yra nuo 74 % iki 90 % [1], sistemos darbo rezultatas kitų produktų palyginime atrodo patenkinamai.

Pagrindinis darbo uždavinys buvo sukurti sistemą, kuri gebėtų atpažinti Parkinsono ligą pagal galimus ligos požymius, nagrinėjant subjektus pagal jų eisenos ypatybes. Toks uždavinys yra įvykdytas, tačiau egzistuoja 1/5 dalies netikslumas. Toks netikslumas argumentuojamas kiekvieno žmogaus eisenos unikaliomis savybėmis. Tokia neigiama metodo savybė yra pašalinama, atliekant diagnozę mažiausiai 5 kartus.

Darbe planotas naudoti dirbtinių neuronų tinklas panaudotas nebuvo. Duomenys savybių erdvėje buvo lengvai atskiriami linijine funkcija, todėl naudoti kompleksinio dirbtinių neuronų tinklų klasifikatorių nėra prasmės. Panaudotas paprastesnis naivus Bayes klasifikatorius.

Naudojamų subjektų skaičius klasifikatoriaus apmokymui ir tikrinimui panaudotas nebuvo. Iš viso, klasifikatoriaus apmokymui planuota panaudoti 60 Parkinsono liga sergančių ir 50 kontrolinių subjektų. Klasifikatoriaus tikrinimui planuota panaudoti 33 sergančių ir 23 sveikų subjektų. Kiekvienas iš subjektų generuoja skirtingą skaičių patikrintų žingsnių signalų, iš kurių toliau skaičiuojamos savybės. Duomenų analizės metu, norint suvienodinti ir tuo pačiu supaprastinti uždavinio nagrinėjimą, pirmiausiai iš kiekvieno subjekto išskirti žingsnio fazės signalai ir jie visi sujungti į vieną matricą. Tuomet egzistuoja dvi matricos, kurios priklauso skirtingai subjektų grupei. Toliau dalinti duomenis pagal kiekvieno subjekto sugeneruotų duomenų kiekį nėra prasmės, kadangi pirmiausiai yra naudojamas skirtingas subjektų skaičius kiekvienos grupės atžvilgiu ir yra garantuojamas duomenų skaičiaus neatitikimas. Skaičiaus atitikimas yra būtinas erdvės transformacijai atlikti, todėl reikia suvienodinti esamų duomenų



skaičių. Tikrinimo atveju turi būti atliekama tokia pati procedūra. Dėl šios priežasties buvo parinktas visų esamų duomenų dalinimas į tris lygias dalis kiekvieno programos modulio įgyvendinimui.

Sistema buvo planuojama dirbti nerealiu laiku. Toks uždavinys buvo įvykdytas. Darbe toks pasirinkimas argumentuojamas pačio eisenos variacijos chaotišku kitimu. Reikalingas ilgas, 50 ilgio vertės variacijos verčių vektorius, norint efektyviai nustatyti ar yra ligos požymiai ir galima teigti, kad subjektas turi Parkinsono ligą.

Gautus rezultatus galima pagerinti, išnagrinėjus daugiau kontrolinių subjektų eisenos ypatybių, bei Parkinsono liga sergančių subjektų eisenos ypatybes. Iš viso buvo išnagrinėti 93 sergantys subjektai ir 73 sveikas subjektas. Iš sveikų subjektų iš viso buvo išskirta 3543 kojos pakilimo nuo žemės signalų, 3583 kojos prisilietimo prie žemės signalai. Iš sergančių subjektų iš viso buvo išskirta 3241 kojos pakilimo nuo žemės signalų, 3217 kojos prisilietimo prie žemės signalai. Iš gautų signalų buvo paskaičiuota jų ilgiai ir naudojantis slankiojančio lango metodu – jų variacija. Iš viso, sveikų subjektų buvo požymių buvo 1790, sergančių subjektų požymių 1619 verčių. Nurodyto darbo rezultatus gali patikslinti tik dar didesnis subjektų skaičius, bei ilgesnis eisenos laikas, kuris šių duomenų atveju buvo tik 2 min.

Darbe panaudotos priemonės lūkesčius pateisino dalinai. Išskirta savybė identifikuoja ligos požymius tik esant ilgos eisenos prielaidai – subjektas turi atlikti eisenos patikrą ilgiau negu 2 min. Rekomenduojama eisenos trukmė yra 5 min. Tokiu atveju sistema geriausiai identifikuos ligos simptomą pagal parinktą savybę. Taip pat klasifikavimo rezultatus gali pagerinti abiejų kojų naudojimas savybių išskyrimo metu, kadangi eisenos nesimetriškumas gali galioti ne tik kairiai kūno pusei, tačiau ir dešinei.



Literatūros ir informacinių šaltinių sąrašas

- [1] Artūras; Griškevičius Julius [Griskevicius Julius]. Aubin, Patric M.; Serackis. A ground reaction force artificial neural network classifier for the diagnosis of parkinson's disease. *Proceedings of XXIIIrd ISB2011 Congress*, pages 428–437, July 2011.
- [2] Christopher J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.
- [3] Chih-Chung Chang and Chih-Jen Lin. Libsvm – a library for support vector machines.
- [4] Arnaud Delval, Julia Salleron, Jean-Louis Bourriez, Séverine Bleuse, Caroline Moreau, Pierre Krystkowiak, Luc Defebvre, Patrick Devos, and Alain Duhamel. Kinematic angular parameters in pd: reliability of joint angle curves and comparison with healthy subjects. *Gait & Posture*, 28(3):495–501, 2008.
- [5] Silvi Frenkel-Toledo, Nir Giladi, Chava Peretz, Talia Herman, Leor Gruendlinger, and Jeffrey Hausdorff. Effect of gait speed on gait rhythmicity in parkinson's disease: variability of stride time and swing time respond differently. *Journal of NeuroEngineering and Rehabilitation*, 2(1):23, 2005.
- [6] Jeffrey M. Hausdorff, Merit E. Cudkowicz, Renée Firtion, Jeanne Y. Wei, and Ary L. Goldberger. Gait variability and basal ganglia disorders: Stride-to-stride variations of gait cycle timing in parkinson's disease and huntington's disease. *Movement Disorders*, 13(3):428–437, 1998.
- [7] Jeffrey M. Hausdorff, Susan L. Mitchell, Renée Firtion, C. K. Peng, Merit E. Cudkowicz, Jeanne Y. Wei, and Ary L. Goldberger. Altered fractal dynamics of gait: reduced stride-interval correlations with aging and huntington's disease. *Journal of Applied Physiology*, 82(1):262–269, 1997.
- [8] T. Hofmann, B. Schölkopf, and A. J. Smola. Kernel methods in machine learning. *ArXiv Mathematics e-prints*, January 2007.
- [9] A.K. Jain, R.P.W. Duin, and Jianchang Mao. Statistical pattern recognition: a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):4–37, jan 2000.
- [10] 1998 Kevin Murphy. Hidden markov model (hmm) toolbox for matlab.
- [11] E. KNUTSSON. An analysis of parkinsonian gait. *Brain*, 95(3):475–486, 1972.
- [12] H.H. Manap, N.M. Tahir, and A.I.M. Yassin. Statistical analysis of parkinson disease gait classification using artificial neural network. In *Signal Processing and Information Technology (ISSPIT), 2011 IEEE International Symposium on*, pages 060–065, dec. 2011.



- [13] Vojtech Franc Michal. I. Schlesinger, Vaclav Hlavac. Statistical pattern recognition toolbox, September 2000.
- [14] G.B. Moody, R.G. Mark, and A.L. Goldberger. Physionet: a web-based resource for the study of physiologic signals. *Engineering in Medicine and Biology Magazine, IEEE*, 20(3):70–75, may-june 2001.
- [15] Maksim Norkin. Human activity recognition using Hidden Markov Models, September 2011.
- [16] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, feb 1989.
- [17] Irina Rish. An empirical study of the naive bayes classifier. In *In proceeding of IJCAI-01 workshop on Empirical Methods in AI*, pages 41–46, November 2001.
- [18] Jonathon Shlens. A tutorial on principal component analysis. <http://www.sn1.salk.edu/~shlens/pub/notes/pca.pdf>, December 2005.
- [19] J. Stamatakis, J. Cremers, D. Maquet, B. Macq, and G. Garraux. Gait feature extraction in parkinson’s disease using low-cost accelerometers. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 7900–7903, 30 2011-sept. 3 2011.
- [20] B. W. Stansfield, S. J. Hillman, M. E. Hazlewood, and J. E. Robb. *Gait & posture*, volume 23, chapter Regression analysis of gait parameters with speed in normal children walking at self-selected speeds, pages 288–294. Elsevier Scienccem, Apr 2006.
- [21] I. Tien, S.D. Glaser, and M.J. Aminoff. Characterization of gait abnormalities in parkinson’s disease using a wireless inertial sensor system. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 3353–3356, 31 2010-sept. 4 2010.
- [22] M. Welling. Fisher linear discriminant analysis. *Department of Computer Science, University of Toronto*, 2005.
- [23] Yunfeng Wu and S. Krishnan. Statistical analysis of gait rhythm in patients with parkinson’s disease. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 18(2):150–158, april 2010.



Santrauka

The goal of the Bachelor thesis was to develop an application, which is capable of Parkinson's disease recognition from gait analysis. The work was started from the alternatives methods review, with discussion of drawbacks and advantages. Based on this review, the methods, which will be used in this work, was derived. The second step was to find the features of the signals, which would most effectively separate the Parkinson's subjects from control subjects. The most logical feature had problems in feature space, so the dimensional reduction method had to be applied. The classification mechanism was chosen very simple, because of the one-dimensional data in feature space and classification only of two classes with no further information. System verification confirmed, that system is able to recognize Parkinson's subjects, but additional experiments with more data must be made.



PRIEDAI

1 priedas. PCA įgyvendinimas, panaudojus skirtingas metodikas.

6.1 programinis kodas. PCA įgyvendinimas, panaudojus SVD

```
function [signals,PC,V] = pca2(data)
% PCA2: Perform PCA using SVD.
% data - MxN matrix of input data
% (M dimensions, N trials)
% signals - MxN matrix of projected data
% PC - each column is a PC
% V - Mx1 matrix of variances
[M,N] = size(data);
% subtract off the mean for each dimension
mn = mean(data,2);
data = data - repmat(mn,1,N);
% construct the matrix Y
Y = data' / sqrt(N-1);
% SVD does it all
[u,S,PC] = svd(Y);
% calculate the variances
S = diag(S);
V = S .* S;
% project the original data
signals = PC' * data;
```



6.2 programinis kodas. PCA įgyvendinimas, panaudojus tikrinių vektorių dekompoziciją

```
function [signals,PC,V] = pca1(data)
% PCA1: Perform PCA using covariance.
% data – MxN matrix of input data
% (M dimensions, N trials)
% signals – MxN matrix of projected data
% PC – each column is a PC
% V – Mx1 matrix of variances
[M,N] = size(data);
% subtract off the mean for each dimension
mn = mean(data,2);
data = data – repmat(mn,1,N);
% calculate the covariance matrix
covariance = 1 / (N–1) * data * data';
% find the eigenvectors and eigenvalues
[PC, V] = eig(covariance);
% extract diagonal of matrix as vector
V = diag(V);
% sort the variances in decreasing order
[junk, rindices] = sort(–1*V);
V = V(rindices);
PC = PC(:,rindices);
% project the original data set
signals = PC' * data;
```