

Turinys

Žymenys ir santrumpos	8
1. Įvadas. Užduoties analizė	9
2. Informacinių Parkinsono ligos diagnostikos sistemų apžvalga	12
3. Signalų analizės programos kūrimas	17
3.1. Bendros programos struktūrinės schemos sudarymas	17
3.2. Pirminio signalų apdorojimo programos kūrimas	18
3.3. Požymių išskyrimo programos kūrimas	23
3.3.1. Dimensijų mažinimas	23
3.3.2. Savybių tyrimas	25
3.4. Požymių klasifikavimo programos kūrimas	33
3.5. Duomenų analizės programos kūrimas	35
4. Signalų analizės programos įgyvendinimas	37
4.1. Bendro programos algoritmo schemos sudarymas	37
4.2. Požymių klasifikavimo programos algoritmo schemos sudarymas	38
4.3. Signalų analizės programos įgyvendinimas	39
5. Signalų analizės programos patikra	42
5.1. Eksperimentų plano rengimas	42
5.2. Duomenų eksperimentams rengimas	42
5.3. Programos patikros rezultatai	43
6. Rezultatų apibendrinimas	44

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);

1. Įvadas. Užduoties analizė

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

Darbo tikslas yra sukurti programą, gebančią atpažinti jėgos jutikliais gautus signalus, priklausančius Parkinsono liga sergantiems pacientams. Programa bus paremta Matlab platforma. Tokia platforma buvo pasirinkta dėl didelių įrankių kiekių, kuris yra prieinamas Matlab aplinkoje. Taip pat sekančioje platformoje yra labai patogų ir greitą realizuoti signalų apdorojimo sistemas dėl jos architektūros - visi kintamieji yra matricos, kas yra labai patogų. Jėgos jutikliai parodo vertikalią žemės reakcijos jėgą (VŽRJ) (angl. Vertical Ground Reaction Force (vGRF)). Fizikoje, ir būtent biomechanikoje, VŽRJ nurodo kokia jėga žemė atsako ją slėgiančiam objektui. Kaip pavyzdžiui, stovinti žmogus slėgia žemę jėga, kuri lygi jo masei ir tuo pat metu, žemė slėgia žmogų priešinga, lygiai tokia pačia jėga. Parkinsono liga buvo pasirinkta dėl to, kad šiuo metu ji yra aktyviai tiriama [1], kadangi visuomenė moka labai didelius pinigus Parkinsono liga sergantiems asmenims. Per metus ši suma gali siekti 6 milijardus dolerių. Ligos rizika su amžiumi tik didėja, todėl analitikai prognozuoja, kad ateityje visuomenė mokės žymiai daugiau dėl žmonių populiacijos senėjimo.

Didžiausia ligos atpažinimo problema slypi savybių, kurios geriausiai atskirs sergantį Parkinsono liga nuo sveiko, nesergančio asmens. Darbo metu bus apžvelgtos kelios galimo savybės, kurios gali atskirti tokius pacientus. Blogiausias galimas variantas būtų nelinijinė funkcija atskirti didelių dimensijų duomenys. Geriausias galimas variantas būtų mažos dimensijos duomenys (iki 3 dimensijų) ir linijškai atskiriami duomenys (kadangi linijinę funkciją realizuoti yra lengviausia). Turint didelių dimensijų duomenis, planuojamos sistemos aparatiniai reikalavimai automatiškai padidėja, kadangi būtina apdoroti labai daug duomenų. Tokią problemą galima išspręsti panaudojus dimensijų mažinimo algoritmus, kurių dažniausiai taikomi: Principinių komponentų analizė (angl. Principal Component Analysis (PCA)), linijinė diskriminanto analizė (anl. Linear Discriminant Analysis (LDA)). Metodai bus aptarti vėlesnėse skyriuose. Jie taip pat vadinami dimensijų praskyrimo algoritmai. Jie yra taikomi, kuomet duomenis yra atskirti netiesiškai. Duomenims, atskirtu tiesiniu dėsniu, galima taikyti paprastą klasifikavimo algoritmą. Jeigu duomenis taip ir nepavyksta atskirti tiesiškai, tenka taikyti kompleksinį klasifikatorių, ko pasekoje gali labai sumažėti klasifikavimo rezultatas. Darbe bus panaudotas Matlab platformoje įgyvendintas įrankis, kuris skirtas suprojektuoti naują dimensijų plokštumą, kurį įgyvendino Boitech Franc savo magistriniam darbe. Darbas buvo apgintas 2000 metais, Čekijos technikos universitete, Prahoje. Tiesinio sklaidimo dirbtinių neuronų klasifikatorių bus

panaudotas iš Matlab “Neural Network Toolbox” įrankio. Likusi sistemos dalis bus aprašyta šiame darbe. Problema kyla pirminiam signalų apdorojimo pasirinkime: visą gaunamą signalą pilnai perduoti ar taikyti slankiojančio lango principą ar kitais, logiškai apibrėžtais būdais, apdoroti pirminį signalą.

Sukurtas produktas gebės pateikti paciento eisenos Parkinsono ligos eigos įvertinimą arba pateikti tikimybę kiek pacientas gali sirgti Parkinsono liga, neatsižvelgiant į kitus ligos simptomus: drebulys (rankų, kojų, žandikaulio, galvos), standumas (galūnių arba liemens sustingimas), bradikinezija (judesių lėtumas), pozicijos nestabilumas (arba sutrikęs balansas). Pati programa veiks nerealiu laiku. Tai reiškia, kad pirmiausiai duomenys bus surenkami, o vėliau įkeliami į programą tolimesniam apdorojimui.

Darbo tema, Parkinsono ligos eigos stebėjimo programa, reiškia, darbo rezultate bus sukurtas algoritmas, kuriuo bus parengta kompiuterinė programa. Pačiam kompiuteryje turi būti veikiantis Matlab programinis paketas. Programa bus rašoma Matlab 7.12.0 (R2011) versija su “Neural Network Toolbox” įrankiu. Eigos stebėjimas reiškia, kad visuomet egzistuoja neapibrėžtas, galimas programos netikslumas. Visiškai programa remtis, diagnozuojant Parkinsono ligą nėra galima, kadangi, kaip jau buvo 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 programa bus rašoma, paminėta anksčiau. Jėgos jutiklių generavimo programa (modulis) bus atsakinga už signalų nuskaitymą iš duomenų bazės ir jų pateikimą sistemos algoritmui. Taip pat signalų generavimo programa (modulis) bus naudojamas programos demonstracinei versijai įgyvendinti.

Kaip buvo minėta, duomenys sistemai bus pateikiami iš “PhysioBank” duomenų bazės. Duomenys duomenų bazėje buvo 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 Parkinsono liga subjektą, “Pt” nurodo Parkinsonu

sergantį subjektą. Pirmas numeris nurodo subjekto identifikacinį numerį, po brūkšnio einantis antras numeris nurodo subjekto duomenų rinkimo seką.

Programos veikimas bus vertinamas taiklumu ir jautrumu. Parametrai yra apskaičiuojami iš pasiklivimo matricos. Taiklumas apskaičiuojamas:

$$Taiklumas = \frac{T_P + T_N}{T_P + F_P + T_N + F_N}, \quad (1.1)$$

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:

$$Tikslumas = \frac{T_P}{T_P + F_P} \quad (1.2)$$

Programos veikimas grafiškai demonstruojamas kreive, kuri nurodo subjekto tikimybę sirgti Parkinsono liga.

2. Informacinių Parkinsono ligos diagnostikos sistemų apžvalga

Šiame skyriuje bus apžvelgtos analoginės informacinės sistemos arba bandymai sukurti sistemą, kuri, remiantis įvairių jutiklių pagalba, gebėtų atpažinti Parkinsono ligą.

Pirmas darbas, kuris bus apžvelgtas yra “Statistical Analysis of Gait Rhythm in Patients With Parkinson’s Disease” [10]. Šis darbas nagrinėja galimas žingsnio variacijos galimybes ir pateikia signalo nuokrypio skaičiavimą kaip pagrindinę savybę, atskiriančią kontrolinius subjektus nuo sergančių. Nagrinėjime buvo panaudoti žingsnio pakilimo nuo žemės ilgis, žingsnio prisilietimo prie žemės ilgis ir bendras žingsnio ilgis. Duomenys gaunami, iš vertikalią žemės reakcijos jėgos matavimų. Autorių teigimu, žingsnio pakilimo ir nusileidimo ilgiai turi aukštą koreliacijos koeficientą su bendro žingsnio ilgiu, todėl nagrinėjime buvo panaudotas tik bendro žingsnio ilgis (2.1 lentelė). Gautas savybių vektorius buvo 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 polinominiu branduoliu sugeba klasifikuoti duomenis su 90.32% tikslumu.

Nagrinėjime buvo naudoti duomenis, pateikti “Hausdorff et al” [3], kurie yra laisvai prieinami iš PhysioNet duomenų bazės [6]. Tokie duomenys bus 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 buvo visiškai sveiki neurologiškai, neturi jokių širdies ligų sutrikimų. Parkinsono subjektų buvo 15, kurių amžius variavo nuo 44 iki 80 metų. Ūgis ir svoris žymiai nesiskyrė tarp kontrolinių ir Parkinsono subjektų, todėl į tokius duomenis dėmesio nebuvo kreipiama. Parkinsono subjektai buvo 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 apskaičiuojama:

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

2.1 lentelė Koreliacijos koeficientai tarp σ bendro žingsnio (σ_r), kojos pakilimo (σ_w) ir kojos prisilietimo (σ_a) prie žemės ilgio parametrų [10].

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 [10].

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

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

Tyrimo metu buvo nustatyta, kad Parkinsono subjektų žingsnio ilgio variacija yra žymiai padidėjusi, lyginant su kontroliniais subjektais (2.2 lentelė), tačiau žingsnio vidurkis lieka beveik nepakitęs. Lentelėje taip pat pateikiami STC (angl. Signal Turn Counter) savybės vertės. 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 ypatybės. Tokio matematinio “derinimo” rezultate gali įvykti, kad algoritmas bus priderintas būtent prie nagrinėjime naudojamų duomenų, tačiau jis gali būti visiškai bevertis į jį pateikus visiškai nematytus algoritmui duomenis.

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

Sekančiam darbe, kuris bus pateiktas apžvalgai [8], nėra naudojama vertikali žemės reakcijos jėga, skaičiuojant savybes. Sprendimas orientuojamas naudoti pigius, keturis linijinio pagreičio jutiklius. Toks sprendimas yra paprastesnis, už prieš tai apžvelgtą, kadangi jame buvo panaudoti šešiolika vertikalios žemės reakcijos jutiklių. Nagrinėjamos Parkinsono ligos požymiai buvo eisenos stingimas ir nesimetriškumas. Iš pirminės sprendimo apžvalgos iškarto galimas

toks sprendimo trūkumas - tiriant galimas savybes, buvo panaudotas tik vienas Parkinsono liga sergantis subjektas ir vienas kontrolinis subjektas. Taip atliekant tyrimą, egzistuoja didelė tikimybė, kad visos gaunamos signalo eisenos savybės gali būti būdingos tik nagrinėjamam subjektui, tačiau gali visiškai negalioji kitiems subjektams ar subjektų grupei. Egzistuoja ir tokia tikimybė, kad ištirtos savybės, kurios galioja Parkinsono subjektui, gali galioji ir kontroliniam subjektui, kadangi buvo panaudotas tik vienas kontrolinis subjektas tyrimo metu.

Nepriklausomai nuo atlikti tyrimo siaurumo, darbe yra pateikta svarbios eisenos analizės išvados. Kontrolinio subjekto eisenoje nebuvo pastebėta jokių eisenos sustingimo požymių, kairės ir dešinės kojos motorinės operacijos buvo atliekamos simetriškai, išskyrus kairės kojos kelio didžiausios amplitudės pagreičio kojos susilietimo ir pakilimo su žeme metu. Abiejų kojų susilietimo su žeme laiko ilgis buvo simetriškas, kas ir patvirtina pusiausvyrą eisena. Parkinsono subjektas parodė dešinės kojos eisenos sąstingio nesimetriškumą. Pacientas naudojo kompensacijos mechanizmą, kadangi jo eisena automatiškai turėjo stoką. Jis peržengdavo dešinės pusės judesius, kuomet pradėdavo ėjimą. Kairės pusės judesiai buvo atliekami tik palaikant balansą. Pati eisena yra labai lėta, o sustingimas įvyksta tik dešinėje pusėje. Iš gautų rezultatų seka, kad analizuojamas pacientas visuomet bando kompensuoti eisenos pokytį, naudodamasis viena puse. Kadangi tyrimas buvo atliktas tik su vienu pacientu, nėra galima tiksliai nusakyti, kad kompensacija vyksta tik iš dešinės pusės, tačiau ką galima tvirtai sakyti, kad naudojantis kompensacijos mechanizmą - egzistuoja padidėjusi variacijos tikimybė. Tokią hipotezę sėkmingai patvirtina ir prieš tai atlikta darbo analizė [10]. Kompensacijos nebuvimas kontrolinio subjekto atveju garantuoja, kad žingsnio ilgio variacija nebus didelė, o atvirkščiai - artės į nulį.

Tolimesnė sistema apžvalgai siūlo jau konkretų algoritmą ir techninę įrangą jo įgyvendinimui [9]. Darbe panaudoti žingsnio, siūbavimo ir nuokrypio sukimo fizinės eisenos savybės ėjimo metu, panaudota principinė komponentų analizė (angl. Principal Component Analysis (PCA)) savybių erdvės sudarymui ir palaikymo vektorių mašina (angl. support vector machine (SVM)) klasifikavimo mechanizmui įgyvendinti. Naudojama sensorinė Micro-AHRS sensorinė platforma į kurią įeina 50 – g trijų ašių linijinio pagreičio jutiklis ir 1200 °/s trijų ašių kampinio pagreičio jutiklis. Gautas modelis sėkmingai veikia 93 % tikslumu ir taiklumu. Naudojamas svoriui jautrus (angl. cost-sensitive) apmokymas tam, kad nustatyti kokios savybės turi didesnę reikšmę formuojant savybių erdvę, bei atliekant klasifikavimą. Sprendimas buvo atliktas, panaudojus pakankamą skaičių subjektų - 23 subjektai su diagnozuota Parkinsono liga ir 16 kontrolinių subjektų, kurie neturi jokių neurologinių susirgimų ligos istorijoje. Tyrimo metu taip pat buvo nustatytas tikslas identifikuoti mažą ir didelį eisenos sutrikimo atvejį. Iš dalyvaujančių Parkinsono subjektų, 11 turi didelį eisenos sutrikimą, 12 turi mažesnį eisenos sutrikimą.

Tyrime buvo panaudoti iš viso 67 normalizuotos laiko srities savybės, kurios yra svarbios motorinei veiklai. Duomenys buvo normalizuoti dėl skirtingų savybių amplitudės srities. Normalizacija yra labai svarbi procedūra, prieš pateikiant duomenis dimensių analizavimo metodu. Taip yra užtikrinama, kad naudojamos savybės turi vienodą įtakos faktorių, projektuojant naują dimensių erdvę [5]. Po dimensių mažinimo metodo pritaikymo, liko tik 11 savybių, kurių

2.3 lentelė Klasifikavimo rezultatas, naudojant pastovų ir kintantį klaidingo klasifikavimo metodo kriterijų [9].

	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 Klasifikavimo rezultatas identifikuojant lengvą ir sunkią Parkinsono ligos eigos atvejį. [9].

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

bendra variacija yra 80.20 %. Parkinsono ligos eigos sunkumui nustatyti, prie bendros savybių erdvės buvo pridėti dar papildomos keturios savybės. Sprendimas, pateikti klasifikatoriui dar papildomas keturias dimensijas nebuvo tiksliai argumentuotas. Nėra nurodyta ar savybės buvo įterptos po dimensijų mažinimo ar prieš tai. Jeigu tai buvo atlikta po dimensijų mažinimo, tai kelia abejonių naujai sukonstruota erdvė - sistema nėra gerai apibendrinta, 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 klasifikatoriaus su radialiniu gauso branduoliu sugeba subendrinti savybių erdvę. Parkinsono subjekto atpažinimas yra pateiktas 2.3 lentelėje. Geriausias klasifikavimo rezultatas buvo pasiektas, naudojant kintamą klaidos vertinimo kriterijų. Nurodytas klasifikavimo metodas atpažįsta Parkinsono ligą labai gerai (100 % tikslumas). Toks rezultatas yra labai didelis iššūkis darbe naudojamai sistemai. Projektuojama sistema turi mažiausiai atsilikti nuo esamo rezultato 5 % tikslumo atžvilgiu, o idealiausiu atveju - jam prilygti.

Atliekant sudėtingesnę klasifikavimą tarp lengvos eisenos sutrikimo, sunkios eisenos sutrikimo Parkinsono ir kontrolinio subjekto, rezultatai gaunami prastesni. Sunkios eisenos sutrikimas atpažįstamas labai prastai, rezultatas siekia tik kiek daugiau negu pusę visų duomenų 52.4 %, Lengvos eisenos sutrikimas atpažįstamas apie 13 % geriau, tik 66.7 %. Geriausiai atpažįstamas tik kontrolinės eilės subjektas, net 91.7 %.

Aptartas metodas pasiūlo konkrečią įrangą, algoritmą, savybes, naudojamas savybių erdvei sudaryti, tačiau užsibrėžtas tikslas buvo pasiektas tikrais iš dalies - suprojektuota sistema, naudojanti pigius vertikaląus pagreičio jutiklius, gebanti atpažinti eigos sutrikimus pas Parkinsono sergančius subjektus, tačiau sistema labai blogai identifikuoja lengvą ir sunkų eisenos sutrikimą. Nurodyta kryptimi reikia atlikti labai didelį darbo kiekį, norint pilnai identifiukuoti

pagal eisena ir Parkinsono ligos stadiją ir atitinkamai pateikti HAY skalės vertinimą.

3. Signalų analizės programos kūrimas

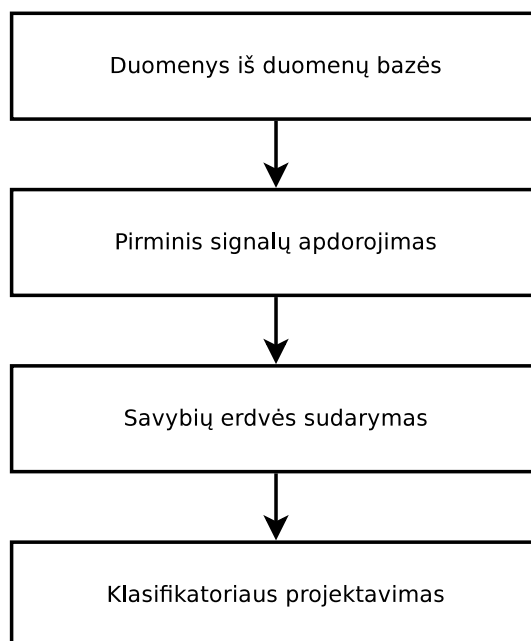
Šiame skyriuje apžvelgsime kylančias problemas, kuriant signalų analizės programa ir galimus jų sprendimus. Kiekvienas sprendimas bus išnagrinėtas, apžvelgta jo metodika, taikymo problemos bei rezultatai.

3.1. Bendros programos struktūrinės schemos sudarymas

Priklausomai nuo taikomos metodikos, programos struktūrinės schemos skiriasi, kadangi vienas sprendimas reikalauja vieno tipo duomenų struktūros, kitas - kitos struktūros. Programos projektavimo metu buvo išbandyti keli programos variantai ir suprastinus veikimo schemas, kiekvienas iš programos variantų veikė pagal bendrą schemą, kurią galima apibrėžti 3.1 pav. Reikalaujama, kad kiekvienas iš algoritmų blokų būtų nepriklausomas nuo žemiau ar aukščiau einančio bloko, kas reiškia, kad "Pirminio signalų apdorojimo" bloko algoritmo pakeitimas neturi turėti jokios įtakos po jo einančiam "Savybių erdvės sudarymas" blokui.

Kiekvieno bloko užduotis:

1. "Duomenys iš duomenų bazės" bloko užduotis yra nuskaityti iš naudojamos "PhysioBank" duomenų bazės duomenis ir juos įkrauti į Matlab aplinką. Kadangi duomenys pateikiami tekstiniu pavidalu, Matlab aplinkoje yra labai patogi funkcija tokiems duomenims nuskaityti - *dlmread*.
2. "Pirminis signalų apdorojimas" bloko užduotis yra pradinis signalų filtravimas, nuolatinės komponentės pašalinimas. Taip pat, į šį bloką įeina ir atskirų signalų išskyrimas iš bendrai gaunamo signalo: signalo išskyrimas kuomet koja turi susilietimą su žeme, signalo išskyrimas, kuomet koja praranda susilietimą su žeme. Į bloką taip pat įeina ir slankiojančio lango algoritmo taikymas.



3.1 pav. Pirminė programos schema.

3. “Savybių erdvės sudarymas” bloko paskirtis yra pateikti pasirinktam klasifikatoriui išskirtas signalų savybes po pirminio signalo apdorojimo. Tokiomis savybės gali būti signalų dažninės komponentės (Furjė koeficientai), koreliacijos koeficientai tarp dešinės ir kairės kojos, savi-koreliacijos koeficientai dešinės ar kairės kojos. Detaliau nagrinėjamas savybes bus aptartos tolimesniuose poskyriuose. Į bloką taip pat įeina ir darbas su dimensijomis - dimensijų skaičiaus mažinimas ar naujos erdvės paieška, kurioje nagrinėjami duomenys yra geriau tiesiškai atskiriami.
4. “Klasifikatoriaus projektavimas” bloko paskirtis yra klasifikatoriaus apmokymas ir jo testavimas. Klasifikatoriaus pasirinkimas yra labai svarbus klausimas nagrinėjamam darbe. Jis turi būti parinktas argumentuotai, pateikiant detalius klasifikavimo rezultatus, remiantis rezultato grafiku. Nepasitikėjimo matrica, bei iš jos apskaičiuotais įverčiais: taiklumu bei tikslumu. Dar svarbus klasifikatoriaus aspektas yra savybių erdvės bendro dėsnio radimas arba jo aproksimacija. Šį faktą galima apibrėžti iš klasifikavimo grafiko. Jeigu klasifikavimo rezultatas yra nestabilus, ilgą laiką neturintis pastovaus rezultato, vadinasi, galima teigti, kad klasifikatorius blogai atliko erdvės dėsnio aproksimavimą, jis nėra tinkamas nagrinėjamai savybių erdvei.

Tolimesniuose poskyriuose bus aptarti įgyvendinti blokinės struktūros elementai.

3.2. Pirminio signalų apdorojimo programos kūrimas

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

Programos kūrimo patogumui, buvo parinkta direktorių architektūra:

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

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

Pirmoji funkcija, priklausanti pirminio signalo apdorojimo programai yra *read_data*. Funkcijos kodas pateikiamas 3.2 pav.

```

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

```

3.2 pav. Duomenų nuskaitymo funkcija iš tekstinės duomenų bylos.

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 [2], 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.

Sekantis žingsnis, po duomenų nuskaitymo, yra jų pirminis apdorojimas. Nagrinėjimas bus pradėtas nuo slankiojančio lango metodo. Programos kodas pateikiamas 3.3 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, pasukines dvi vertes ištrins iš spartinančiosios atminties ir iš naujo lauks naujų dviejų reikšmių langui pilnai užpildyti.

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

```

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

```

3.3 pav. Slankiojančio lango algoritmo pritaikymas.

```

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

```

3.4 pav. Signalų filtravimas dviem Butterworth filtrais.

naudinga tolimesniame darbe.

Sekanti programos dalis atlieka paprastą signalų filtravimą su dvejais *Butterworth* skaitmeniniais filtrais. Pirmasis, aukštų dažnių filtras, skirtas pašalinti signalo nuolatinę komponentę. Antras, žemų dažnių filtras, skirtas pašalinti aukšto dažnio triukšmą, kuris neneša visiškai jokios naudingos informacijos. Filtras įgyvendinamas labai paprastai. Kodo pavyzdys pateikiamas 3.4 pav. Abiejų filtrų eilė yra 9-ta, žemų dažnių filtro ribinis dažnis parinktas 40 Hz. Duomenys diskretizuojami 100 Hz dažniu, vadinasi didžiausias galimas signalo dažnis yra 50 Hz. Didžiausias žmogaus generuojamas dažnis ėjimo metu, remiantis šaltiniu, yra 20 Hz. Užtikrintumui buvo parinktas 40 Hz dažnis. Nuolatinė dedamoji pašalinama su aukšto dažnio filtru, kurio ribinis dažnis yra 1 Hz. Nuolatinė dedamoji neneša jokios informacijos apie eisena, kadangi ji tiktais nurodo naudojamų jutiklių jautrumą.

Toliau seka, priklausomai nuo pasirinkto pirminio signalų apdorojimo būdo, signalo išskyrimas pagal fizinę veiklą. Dominančios signalo būsenos yra, kai subjekto koja turi kontaktą su žeme ir nurodytas subjektas neturi kontakto su žeme. Kontakto su žeme signalo išskyrimui programos kodas yra pateikiamas 3.5 pav.

Pirmiausiai, "Pt_t" kintamojo struktūroje yra saugoma kairės kojos Parkinsono liga sergančių subjektų duomenys. Kiekvienas signalas yra priskiriamas prie *signal* kintamojo, su

```

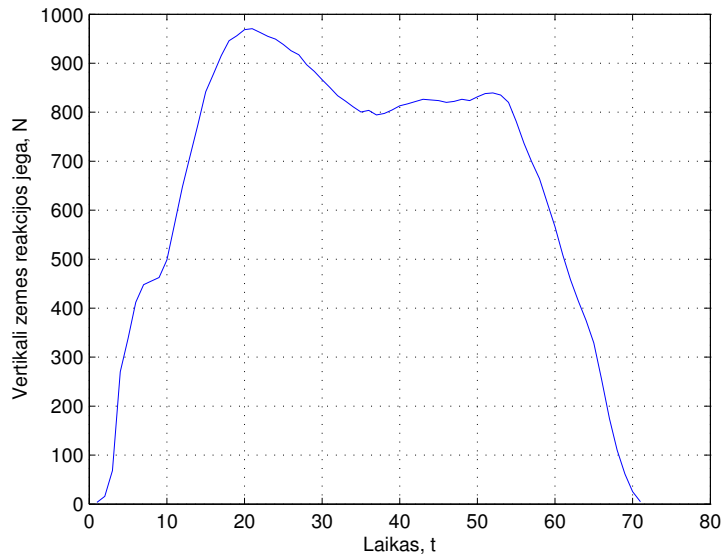
1  for i=1:size( Pt_t, 2 )
2      fprintf(' ');
3      signal = Pt_t{i};
4      buffer = [];
5      k = 0;
6      for j=1:size(signal, 1)
7          if ( signal(j) ~= 0 )
8              buffer = [ buffer signal(j) ]; %ok<AGROW>
9          else
10             if ~isempty(buffer) && length(buffer) > 10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

```

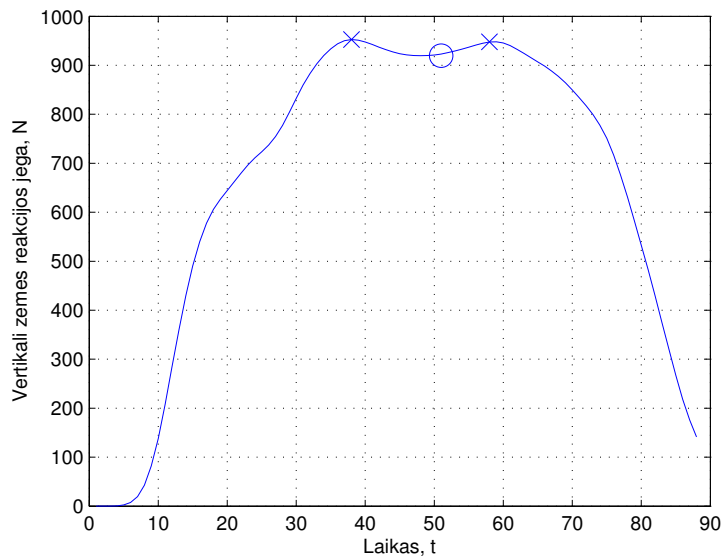
3.5 pav. Kontakto su žeme signalo išskyrimo programos kodo fragmentas.

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. Jeigu signalas yra trumpesnis už 10 verčių, arba turint omenyje, kad signalas buvo diskretizuojamas 100 Hz, tai 0.1 s, vadinasi, signalas yra tiesiog aukšto dažnio triukšmas arba blogas pavyzdys ir signalas yra atmetamas. Jeigu signalas yra ilgesnis už 200 verčių (2 s), vadinasi, duomenų rinkimo metu įvyko klaida ir koja per tokį laiką pakelta nebuvo. Tokia klaida gali būti sukelta, kuomet subjektas ne eina, o stovi ant dviejų kojų arba tik ant kairės kojos. Jeigu visi kriterijai patenkinami, signalas yra priimamas ir kraunamas į laikinąją atmintį, pavadinimu “*data.Pt*”. Signalas, kuriuo metu koja neliečia žemės, yra randamas kartu su ankščiau išnagrinėtu metu. Algoritmas patikrina kiek praėjo laiko (arba kiek verčių yra priskaičiuota) nuo paskutinio užskaityto kojos ant žemės signalo ir įrašo tą signalą į laikiną atmintį, jeigu po praeito signalo nepraėjo mažiau negu 50 verčių arba 0.5 s ir nedaugiau nei 200 verčių arba 2 s. Kojos signalas yra pateiktas 3.6 pav.

Dar vienas pirminis signalo apdorojimas, kurį teko panaudoti tiriant galima bendrą erdvę - kiekvieno kojos susilietimo su žeme signalo dviejų globalių maksimumų ir vieno lokalaus minimumo paieška. Algoritmo rezultatas yra pateiktas 3.7 pav. Kryžiais pažymėtos globalaus maksimumo vietos, apskritimu pažymėta lokalaus minimumo signalo vieta. Išanalizavus daugumos subjektų žingsnio signalus, buvo padaryta išvada, kad kiekviename žmogaus žingsnyje egzistuoja du maksimumai. Vienas maksimumas randamas, kai subjektas yra visiškai atsirėmęs galine pėdos dalimi į žemę, antras maksimumas randamas, kai subjektas visiškai atsiremia prie-



3.6 pav. Susilietimo su žeme signalas.



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

kinę pėdos dalimi. Tarp šių dviejų maksimumų yra pereinamasis laikotarpis, kuris yra signalo lokalus minimumas. Priežastis, dėl kurios algoritmas buvo įgyvendintas, yra vienas darbas, kuriame buvo pasiūlyta naudotis būtent tokiais savybėmis atpažinti Parkinsono liga sergančius subjektus nuo nesergančių subjektų. Pačio algoritmo kodo dalis yra pateikta 3.8 pav.

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

Pirminio signalų apdorojimo programa baigiasi nurodytais metodais. Sekantis žingsnis yra


```

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

```

3.8 pav. Dviejų globalių maksimumų ir vieno lokalaus minimumo radimo algoritmo fragmentas.

apdorotų duomenų perdavimas tolimesnei programos pakopai - požymių išskyrimui.

3.3. Požymių išskyrimo programos kūrimas

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

3.3.1. Dimensijų mažinimas

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

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

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

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

Matematiškai, PCA apibrėžiamas kaip statmena linijinė transformacija, kuri transformuoja duomenis į naują koordinačių sistemą, kurioje didžiausią variaciją projektuojama į pirmą ašį, antroji didžiausia variacija (statmena pirmajai) projektuojama į antrą ašį. Procesas tęsiasi tol, kol yra pasiekiamas norimas dimensijų skaičius.

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

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

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

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

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

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

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

kur S_B yra “išorinės klasės scatter matrix”, S_W yra “vidinės klasės scatter matrix”. Scatter matricų 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 [4]. Tarkim, egzistuoja tam tikri empiriniai duomenys:

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

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

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

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

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

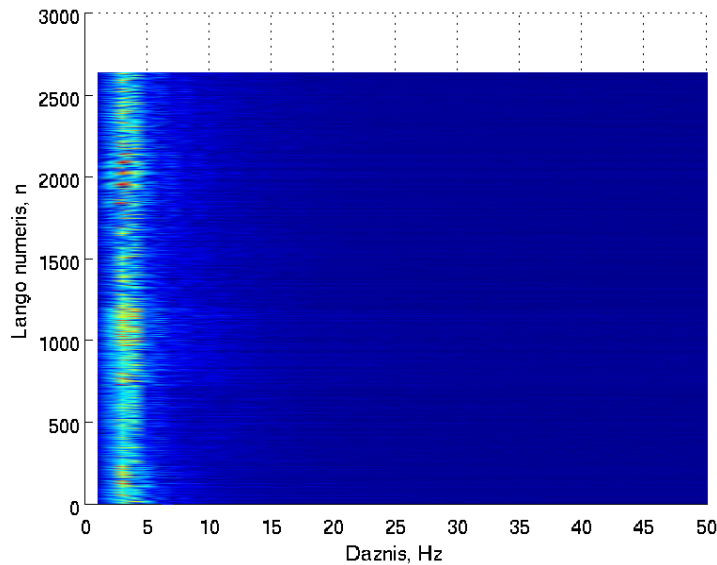
$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle, \quad (3.8)$$

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

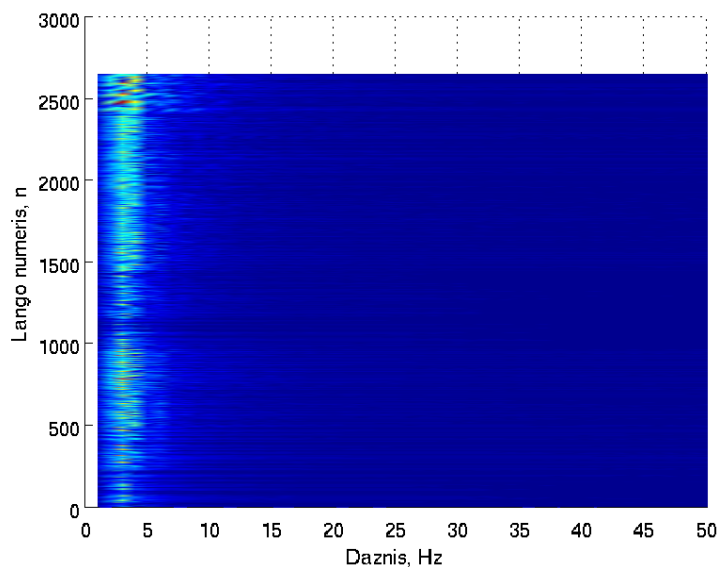
3.3.2. Savybių tyrimas

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

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



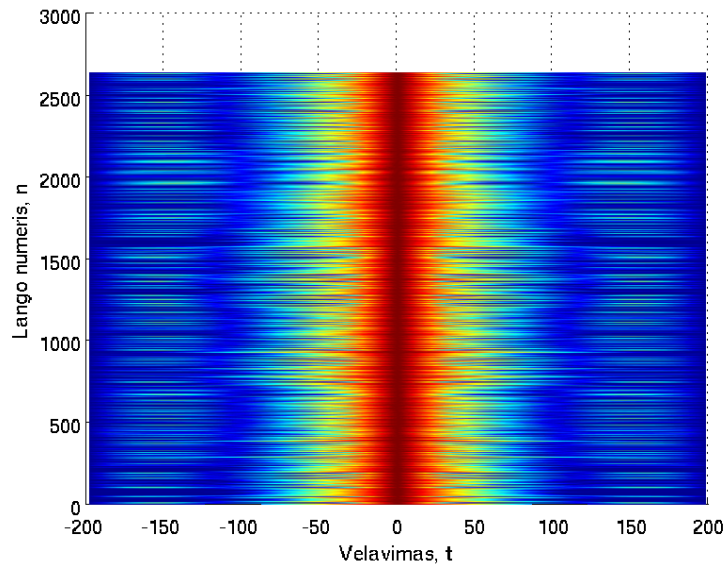
3.9 pav. Kontrolinių subjektų kairės kojos žingsnių signalų dažninės komponentės



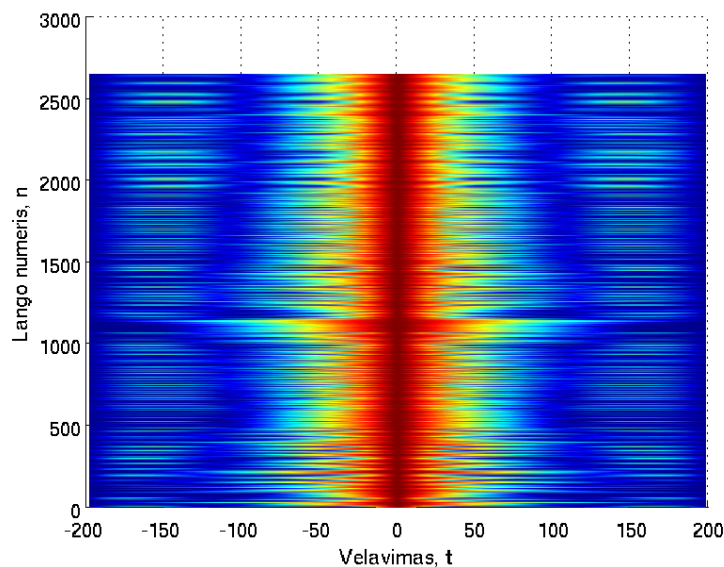
3.10 pav. Parkinsono liga sergančių subjektų kairės kojos žingsnių signalų dažninės komponentės

signalus tarpusavyje.

Toliau bus išanalizuoti Parkinsono liga sergančių subjektų ir kontrolinių subjektų žingsnių signalų dažninės komponentės. Kontrolinių subjektų kairės kojos žingsnių dažninės komponentės yra pateiktos 3.9 pav. Parkinsono liga sergančių subjektų kairės kojos žingsnių dažninės komponentės yra pateiktos 3.10 pav. Kaip matoma iš duotų komponentių grafikų - tiek kontrolinių subjektų, tiek Parkinsono liga sergančių subjektų pagrindinės dažninės komponentės išsidėsto iki 5 Hz ruože. Ties 0 Hz dažninių komponentių nėra, kadangi jos buvo pašalintos filtro pagalba. Už 10 Hz ribos, komponentės neneša visiškai jokios informacijos. Iš to galima padaryti išvadą, kad tiek Parkinsono liga sergančių subjektų, tiek kontrolinių subjektų eisenos yra visiškai vienodos dažnių srityje ir vien remiantis šita informacija nėra galima nustatyti ar



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

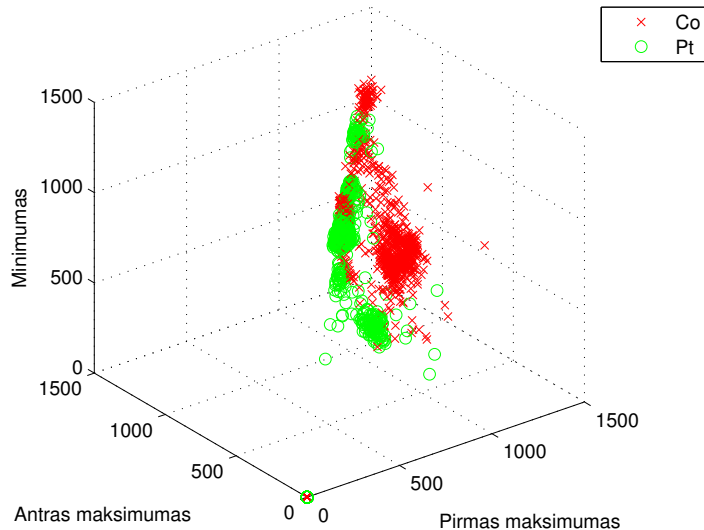


3.12 pav. Parkinsono liga sergančių subjektų kairės kojos žingsnių signalų savi-koreliacijos koeficientai.

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 labai gerus rezultatus ankstesniame tyrime, kuriame, remiantis akselerometro ir giroskopo jutiklių parodymais, reikėjo suprojektuoti algoritmą, gebantį atskirti tokias žmogaus veiklas: stovėjimas,ėjimas,ėjimas aukštyn laiptais,ėjimas žemyn laiptais.

Kontrolinių subjektų kairės kojos savi-koreliacijos koeficientai parodyti 3.11 pav. Parkinsono liga sergančių subjektų kairės kojos savi-koreliacijos koeficientai parodyti 3.12 pav. Kaip matosi



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

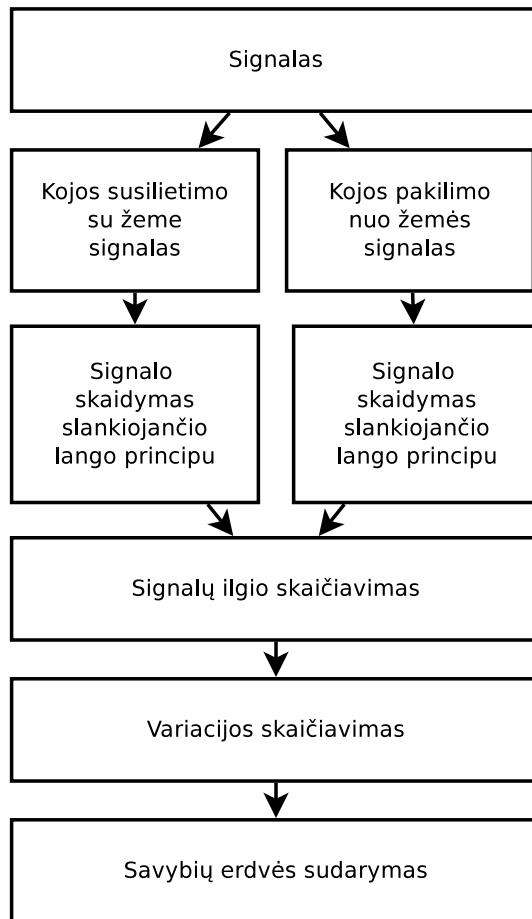
iš korelogramos, tiek sergantys, tiek kontroliniai subjektai turi panašias, o kai kuriais atvejais ir tokias pačias, koreliacijos reikšmes. Remiantis vien tik turima informacija, nustatyti ar subjektas serga Parkinsono liga ar ne, nėra įmanoma.

Sudėtingesnė analizė seka iš dviejų globalių maksimumų ir vieno lokalaus minimumo savybių erdvės. Šios tris savybės sudaro trijų dimensijų plokštumą, kurią galima lengvai pavaizduoti. Pilna savybių erdvė pavaizduota 3.13 pav. Kaip matosi iš duotos erdvės, duomenys neturi jokio koncentracijos centro. Erdvėje jie pasiskirstę pagal nežinomą dėsnį.

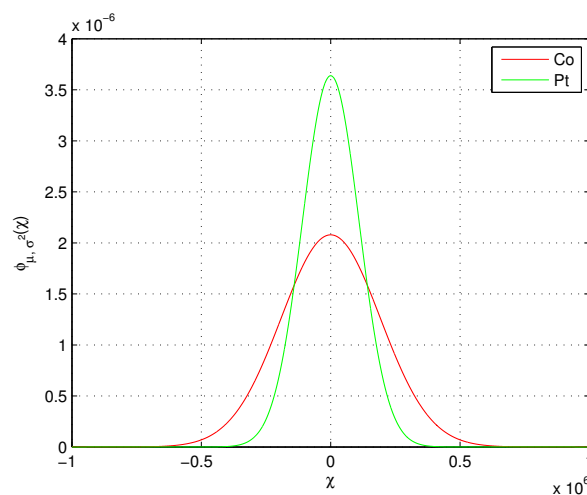
Tokių duomenų pateikti klasifikavimui nėra galima. Grafike “Co” taškai, pažymėti kryžiumi, parodo kontrolinį subjektą, “Pt” taškai, pažymėti apskritimu, parodo Parkinsono liga sergančius subjektus.

Sekančios savybės nagrinėjimui yra siūlomos atlikto tyrimo [2]. 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 3.14 pav. Sekantis žingsnis yra signalo išskaidymas slankiojančio lango principu. Abiejų signalų slankiojančio lango ilgis buvo parinktas 4 verčių ilgio, su 2 verčių perdanga. Kuomet tiek kojos susilietimo su žeme, tiek kojos pakilimo nuo žemės keturi signalai patenkina keliamus reikalavimus - iš jų yra apskaičiuojami signalų ilgiai. Vėliau, abiejų etapų signalams yra paskaičiuojama jų variacija ir taip yra sudaroma savybių erdvė. Toliau bus nagrinėjamos abiejų signalų variacijos vienos dimensijos plokštumoje.



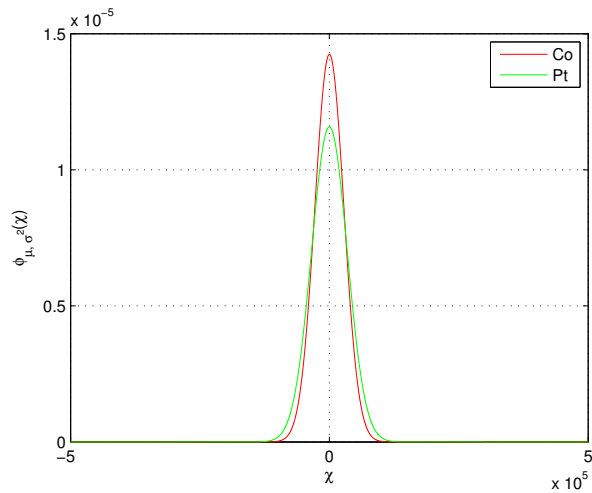
3.14 pav. Pirminis signalo apdorojimas, išskiriant savybę kaip kojos susilietimo su žeme laiko variaciją ir kojos pakilimo nuo žemės laiko variaciją.



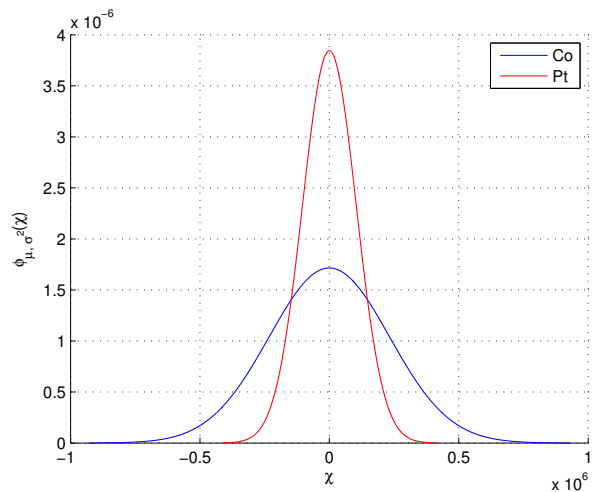
3.15 pav. Kojos prisilietimo prie žemės laiko variacija.

Abiejų signalų variacijų pasiskirstymai yra parodyti 3.15 pav. ir 3.16 pav. Abu pasiskirstymai turi vieną trūkumą - jų vidurkiai sutampa, tačiau gera žinia yra tai, kad jų variacijos skirtingos. Norint išspręsti iškilusią problemą, reikia taikyti dimensių priskyrimo metodus. Darbo metu buvo išbandyti tokie dimensių priskyrimo metodai:

- PCA;



3.16 pav. Kojos pakilimo nuo žemės laiko variacija.

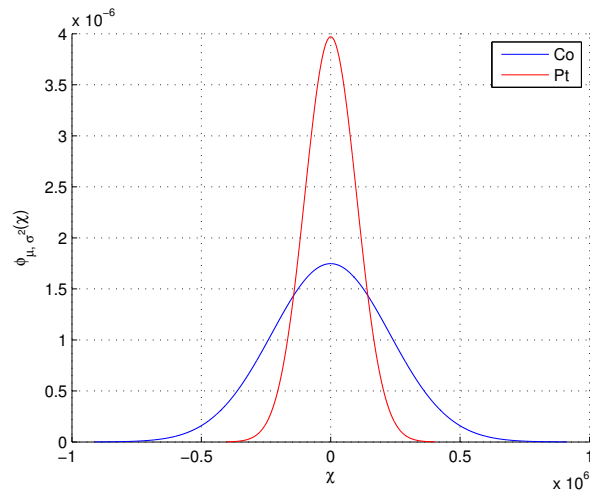


3.17 pav. Kojos prisilietimo prie žemės ir pakilimo nuo žemės laiko variacijos pasiskirstymas po PCA transformacijos.

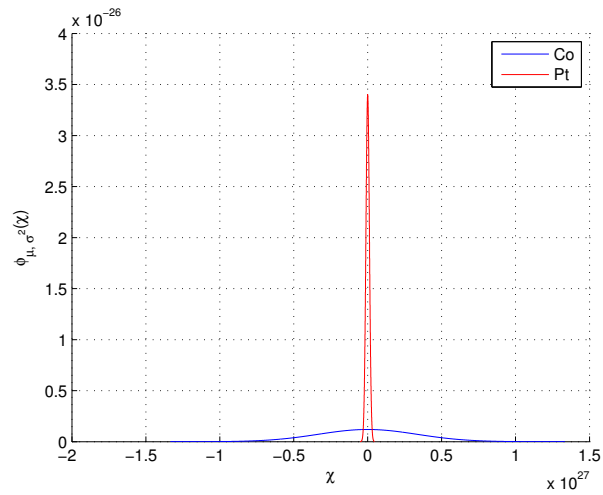
- LDA;
- Kernel PCA (su polinominiu, Gauso branduoliais);
- Kernel LDA (su polinominiu, Gauso branduoliais);

Vienmatė linijinė PCA transformacija pavaizduota 3.17 pav. Kaip matosi iš grafiko - vidurkis nepasikeitė, tačiau žymiai pasikeitė variacija. Vidurkio pokyčio nebuvimas pablogina situacija, kadangi to pačio vidurkio duomenis atskirti nėra įmanoma, todėl ir teikti duomenis klasifikavimui nėra prasmės.

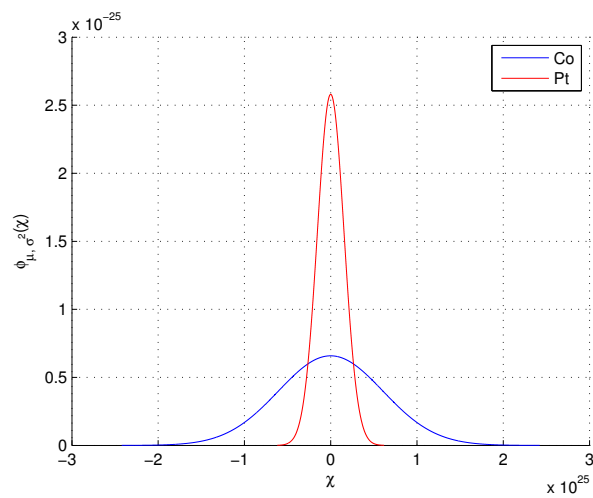
Sekantis metodas yra LDA. Transformacijos rezultatas yra pavaizduotas 3.18 pav. Kaip matosi - tiek po LDA, tiek po PCA duomenų pasiskirstymas nėra kritiškai skirtingas. Galima daryti hipotezę, kad linijinis duomenų praskirstymas šiuo atveju neveikia. Reikia taikyti branduolio metodą ir papildyti jau panaudotus transformacijos metodus polinominiu arba Gauso branduoliu.



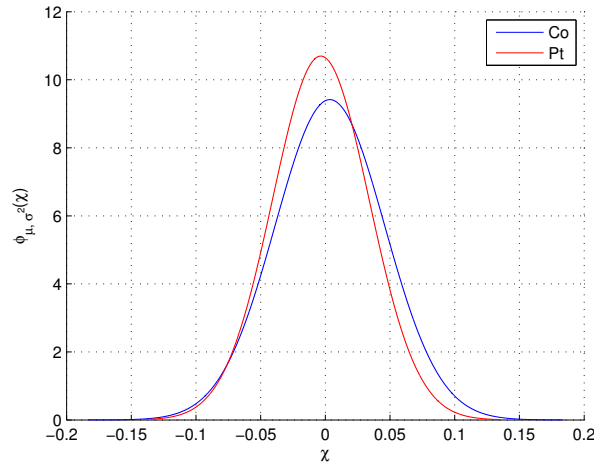
3.18 pav. Kojos prisilietimo prie žemės ir pakilimo nuo žemės laiko variacijos pasiskirstymas po LDA transformacijos.



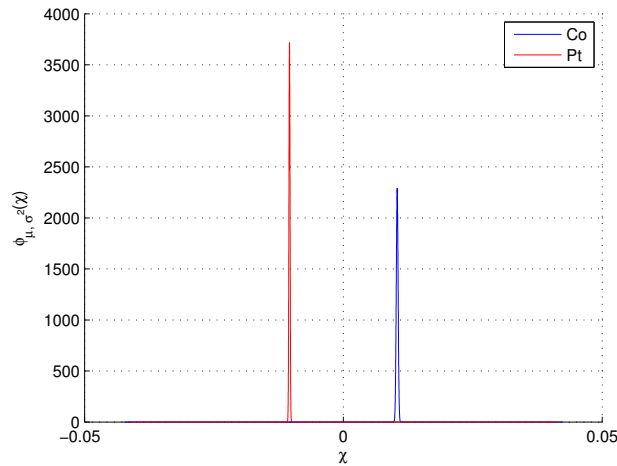
3.19 pav. Kojos prisilietimo prie žemės ir pakilimo nuo žemės laiko variacijos pasiskirstymas po PCA transformacijos, naudojant polinominį branduolį.



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



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



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

Branduolio metodas pritaikytas PCA transformacijai yra pavaizduotas 3.19 pav. ir 3.21 pav. Branduolio metodas pritaikytas LDA transformacijai yra pavaizduotas 3.20 pav. ir 3.21 pav. Polinominis branduolys abiemis atvejais buvo panaudotas su 4 branduolio argumentu.

Kaip matosi, panaudojus polinimoni branduolį, tiek PCA (3.19 pav.), tiek LDA (3.20 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 (3.22 pav.), PCA su tokiu branduoliu (3.21 pav.) duomenis tik dar labiau suvienodina.

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

3.4. Požymių klasifikavimo programos kūrimas

Šiame skyriuje bus išnagrinėti ir pritaikyti populiariausi šiuo metu naudojami klasifikatorių metodai. Tokie klasifikatoriai yra [5]:

- Vektoriaus palaikymo mašina (angl. Support Vector Machine (SVM));
- Paslėptas Markovo modelis (angl. Hidden Markov Model (HMM));
- Naivusis Bayes klasifikatorius (angl. NaiveBayes);
- Tiesioginio sklido 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.

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

Vektoriaus palaikymo mašina šiuo metu yra populiariausias klasifikatorius nagrinėjant netiesiškai atskiriamus duomenis. Klasifikatoriaus testavimo metu buvo naudojami tokie parametrai:

- SVM tipas - Sequential Minimal Optimization;
- Branduolio tipas - linijinis;
- Sureguliuavimo 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, abiems 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 yra vienintelis iš šiame darbe nagrinėjamų klasifikatorių, kuris turi laikinę informaciją. Tokia savybė suteikia “inkaro” galimybę - klasifikatorius gali užsilaikyti prie vienos klasės net ir tuomet, kai pagal savybių erdvę turi būti kitas klasifikavimo rezultatas.

3.1 lentelė Vektoriaus palaikymo mašinos tikslumas ir taiklumas.

	C1	C2
Tikslumas	0.489	0.489
Taiklumas	0.234	0.744

3.2 lentelė Paslėpto Markovo modelio tikslumas ir taiklumas.

	C1	C2
Tikslumas	0.500	0.500
Taiklumas	0.000	1.000

3.3 lentelė Naivaus Bayes klasifikatoriaus tikslumas ir taiklumas.

	C1	C2
Tikslumas	0.508	0.508
Taiklumas	0.764	0.252

Tokia klasifikatoriaus savybė buvo pritaikyta ankstesniame darbe, sudarant žmogaus eisenos atpažinimo sprendimą [7].

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

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

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

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

Lentelėje 3.4 pateikiami tiesioginio sklaidimo 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 sklaidimo neuronų tinklas užduotį atlieka blogai.

3.4 lentelė Tiesioginio sklaidimo dirbtinių neuronų tinklų tikslumas ir taiklumas.

	C1	C2
Tikslumas	0.282	0.287
Taiklumas	0.000	0.392

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

3.5. Duomenų analizės programos kūrimas

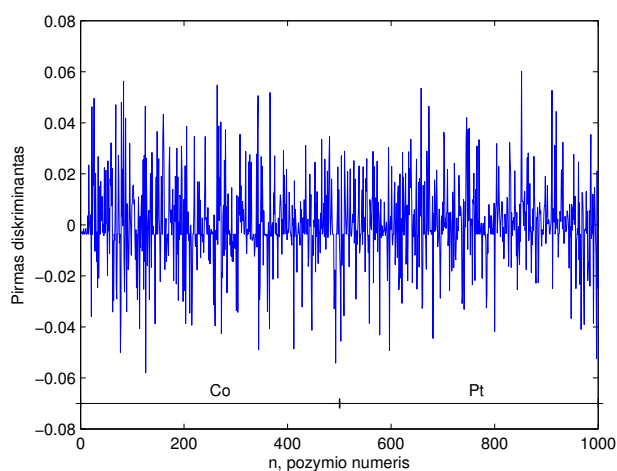
Šiame poskyryje bus aptartas duomenų analizės programos kūrimas. Pasinaudojus prieš tai esančių skyrių informacija bus pateiktas sprendimas, kuris leidžia efektyviai kiek įmanoma efektyviau atpažinti subjektų tipus.

Poskyryje “Požymių išskyrimo programos kūrimas” buvo atlikta savybių ir dimensijų mažinimo metodų analizė. Geriausiai dimensijų mažinimo klausimą išsprendė linijinė diskriminanto analizė, naudojant Gauso branduolį, tačiau iškilo klasifikavimo problema - klasifikuojant duomenis realiu laiku, klasifikavimo rezultatas buvo nepatenkinamas.

Pagrindinė problema, kodėl joks nagrinėjamas klasifikatorius neatliko korektiško klasifikavimo, tai dėl jam pateikiamų duomenų. Kaip pavyzdys yra pateikiamas klasifikatoriaus testavimo metu naudoti duomenis, pavaizduoti 3.23 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 3.22 pav., jie atrodo labiau chaotiški. Iš vaizdinės analizės nėra įmanoma spėti kuri signalo dalis kuriai subjektų grupei priklauso.

Tokios išvados verčia projektuoti kitą klasifikavimo mechanizmą, kuris duomenis klasifikuotų ne kiekvieną pagal kiekvieną nagrinėjamą savybę, o pagal savybės grupes - į laikiną atmintį yra rašoma požymio reikšmė, laukiama, kol laikina atmintis užsipildys iki tam tikros n eilės ir iš gautos sekos yra skaičiuojamas vidurkis (vidurkis skaičiuojamas todėl, nes pagal duomenų pasiskirstymą, kuris parodytas 3.22 pav., pasiskirstymų vidurkiai skiriasi, tačiau variacijos lieka tokios pačios) ir apskaičiuota reikmė naudojama kaip nauja savybė klasifikatoriaus apmokymui, bei testavimui.

Dėl gaunamų duomenų dimensijos ir jų mažo kiekio (kuris priklauso nuo laikinos atminties dydžio, į kurią rašomos naudos erdvės vertės), bus panaudotas paprasčiausias naivaus Bayes klasifikatoriaus mechanizmas. Dėl duomenų stokos, kai kurie duomenys buvo panaudoti visuose trijuose algoritmo veikimo žingsniuose. Viso erdvės sudarymui buvo panaudota 900, apmoky-



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

3.5 lentelė Gautas klasifikavimo rezultatas, naudojant naivų Bayes klasifikatorių.

	C1	C2
Tikslumas	0.875	0.875
Taiklumas	0.750	1.000

mui 500, testavimui 700 reikšmių. Laikinos atminties ilgis buvo pasirinktas 150 verčių, iš kurių buvo skaičiuojamas vidurkis. Apmokymo metu iš viso buvo 3 vidurkiai kiekvienai subjektų grupei, testavimo metu iš viso buvo 4 vidurkiai kiekvienai subjektų grupei. Gauti rezultatai pateikti 3.5 lentelėje.

Gauti klasifikavimo tikslumas yra 0.875, kas viršija prieš tai naudotų metodų tikslumą. Pirmos grupės klasifikavimo tikslumas yra 0.750, antros grupės klasifikavimo tikslumas 1.000. Nuodyti klasifikavimo mechanizmas klasifikavo blogai tik vieną kontrolinį subjektą ir priskyrė jį prie Parkinsono liga sergančių subjektų grupės.

4. Signalų analizės programos įgyvendinimas

Šiame skyriuje apžvelgsime įgyvendinta programinį kodą ir jo veikimo architektūrą. Anksčiau poskyriuose buvo argumentuotai pažvelgti galimi analizės metodai, požymiai ir klasifikatoriai. Visi rezultatai bus panaudoti projektuojant galutinį sprendimą.

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

4.1. Bendro programos algoritmo schemos sudarymas

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

Viso programa yra sudaryta iš trijų fazių:

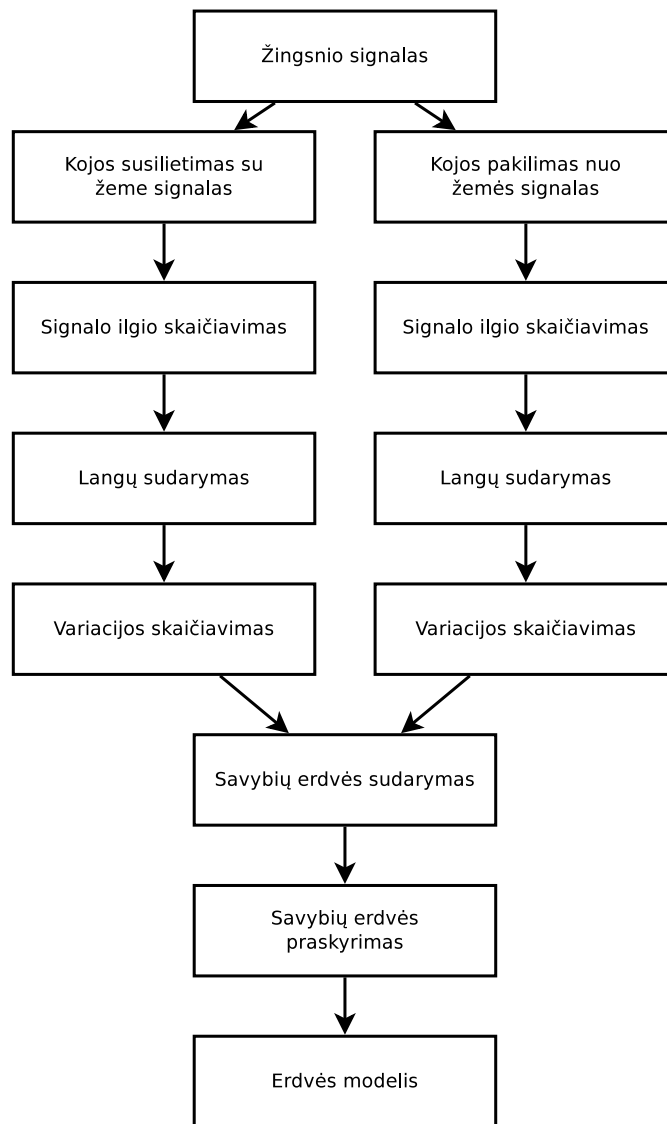
- Pirma fazė susideda iš savybių erdvės sudarymo;
- Antroji fazė skirta klasifikatoriaus apmokymui;
- Trečioji fazė skirta klasifikatoriaus tikrinimui.

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

Trečios, paskutinės fazės struktūrinė schema pavaizduota 4.3 pav. Struktūroje naudojamas erdvės modelis iš pirmos fazės ir klasifikatoriaus modelis iš antros fazės. Fazės įėjimas yra toks pats, kaip ir antrosios fazės, išskyrus tai, kad šios fazės išėjime yra klasifikavimo rezultatas, kurį galima pateikti tiek grafiškai, tiek skaitine išraiška.

Apibendrinus bendrą programos veikimo schemą, toliau paanalizuosim patį klasifikavimo mechanizmą - tiesioginio sklidimo dirbtinių neuronų tinklą.



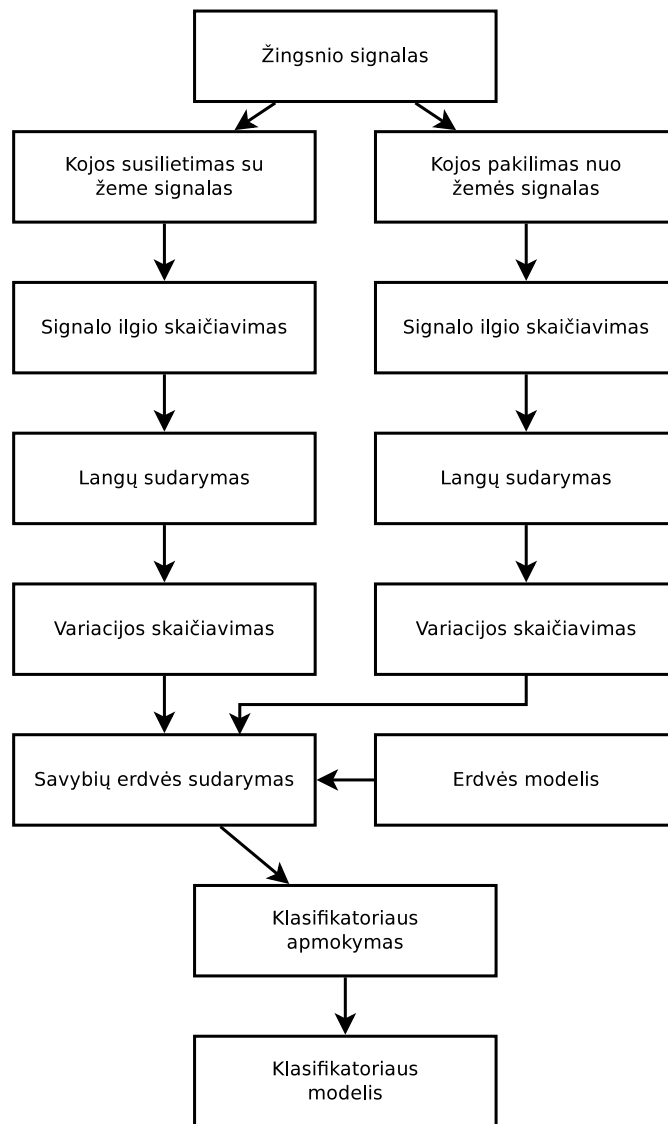
4.1 pav. Programos pirmos fazės grafikas.

4.2. Požymių klasifikavimo programos algoritmo schemas sudarymas

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

Darbe naudojamo tiesioginio sklidimo dirbtinių neuronų tinklo schema pavaizduota 4.4 pav. Įėjimo duomenys nagrinėjamu atveju yra vienos dimensijos, todėl įėjime reikalingas tik vienas neuronas. Bandymo metu, pasirinktas vienas paslėptas neuronas labai gerai atliko klasifikavimo užduotį, todėl jų skaičių didinti nėra prasmės. Išėjime yra du neuronai, kadangi tai yra binarinis klasifikavimo uždavinys.

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 dimensijų skaičių neliaka prasmės, ieškoti klasifikavimo algoritmo, kuris geriausiai apibendrintų turimus duomenis erdvėje nėra prasmės, kadangi duomenų apibendrinimą puikiausiai atliko LDA su Gauso branduoliu.



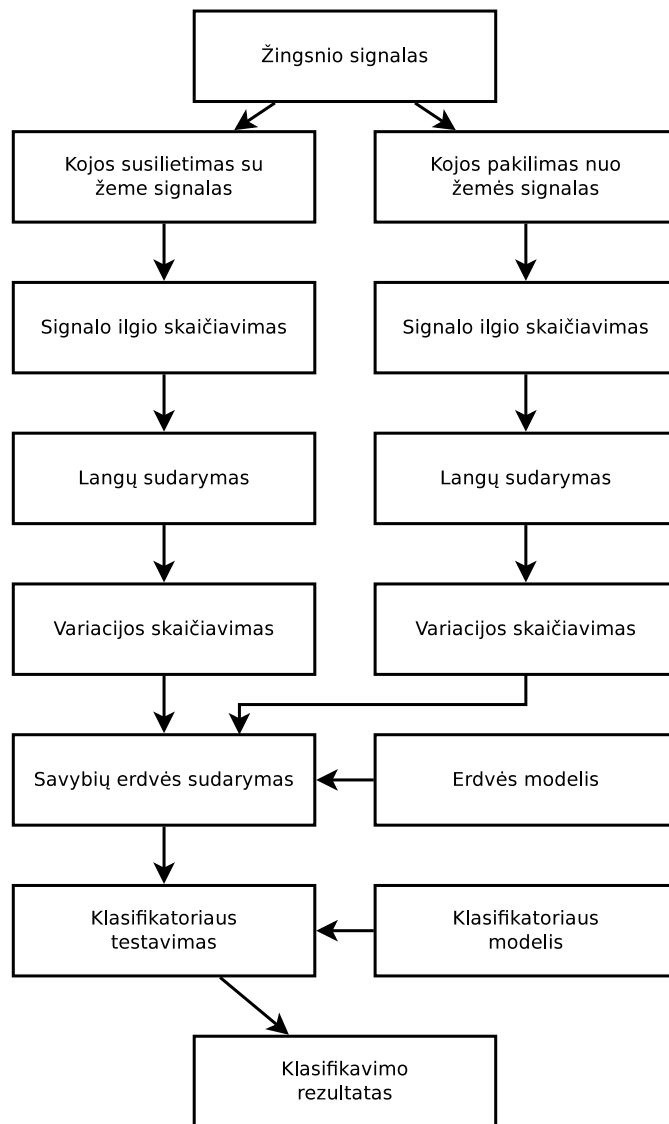
4.2 pav. Antros fazės grafikas.

Dažniausiai dirbtinių neuronų tinklai yra naudojami, kuomet nėra žinoma pagal kokį tiksliai dėsnį duomenys yra pasiskirstę savybių erdvėje. Taip klasifikatoriaus apmokymas vyksta tol, kol dirbtinių neuronų tinklas randa “paslėptą” duomenų priklausomybę. Tokiu atveju galima ieškoti optimaliausią paslėptų neuronų skaičių, galbūt keisti pačio tinklo struktūrą arba net pakeisti neuronų tinklo veikimo architektūrą. Taip pat, mechanizmas yra tiek lankstus, kad apmokymui galima taikyti ne tik “pažymėtus” duomenis, tačiau ir “nepažymėtus”, kuomet nėra žinoma kokie duomenys kokiai klasei priklauso. Tokie tinklai dar vadinami “patys save struktūrizuojantys”.

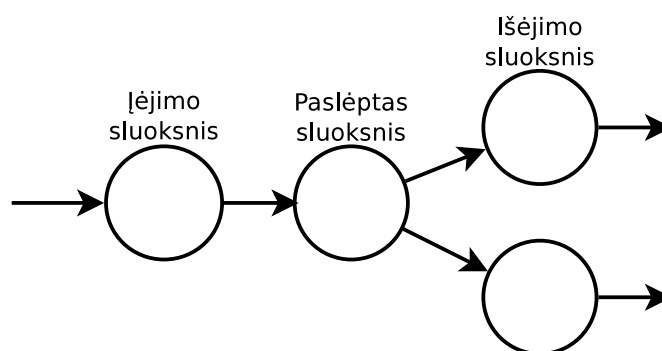
4.3. Signalų analizės programos įgyvendinimas

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

Galutinė programos schema yra pavaizduota 4.5 pav. Kaip buvo aptarta ankstesniuose poskyriuose - programa susideda iš trijų duomenų apdorojimo fazių. Kiekvienos fazės įėjimas



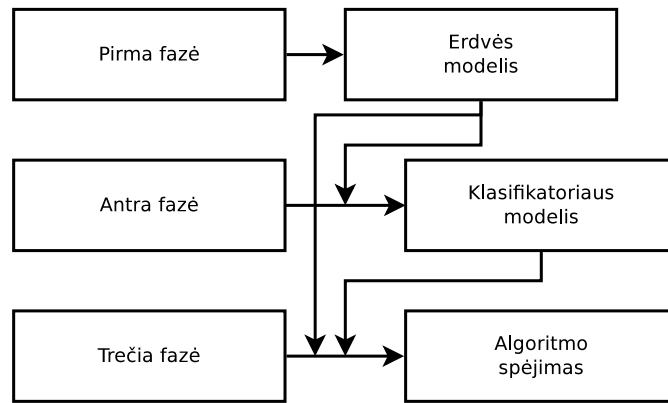
4.3 pav. Trečios fazės grafikas.



4.4 pav. Tiesioginio sklidimo dirbtinių neuronų tinklo schema.

yra toks pats, tačiau kiekvienos fazės išėjimas yra skirtingas. Kiekviena žemiau esanti fazė naudoja virš jos esančios fazės darbo rezultatą. Kritiškai svarbu yra kiekvienai fazei pateikti skirtingus duomenis.

Darbe pateiktas metodas leidžia efektyviai atpažinti ar subjektas serga Parkinsono liga. Didžiausias metodo trūkumas yra tas, kad ji neatsižvelgia į kitus Parkinsono ligos simptomus



4.5 pav. Galutinės programos struktūrinė schema.

- drebulys, eisenos sustingimas.

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

Darbas ties Parkinsono ligos identifikavimu ir jo 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 bus 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 bus 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 bus 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 buvo 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 [5]. 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 bus aptarti galimi eksperimentiniai algoritmo patikros planai.

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, 500 paduoti klasifikatoriaus apmokymui, 500 paduoti klasifikatoriaus testavimui. Taip pat duomenis galima pateikti praleidžiant kas kelintą n narį - kadangi iš viso yra trys algoritmo žingsniai, tuomet $n = 2$. Taip į dimensijų mažinimo algoritmą bus paduotas 1 rinkinys, 4 rinkinys, 7 rinkinys ir taip toliau. Klasifikatoriaus apmokymui bus pateiktas 2 rinkinys, 5 rinkinys, 8 rinkinys ir taip toliau. Klasifikatoriaus testavimui bus pateiktas 3 rinkinys, 6 rinkinys, 8 rinkinys ir taip toliau. Toks duomenų pateikimas leidžia pasiekti "apibendrinto" metodo.

Eksperimento rezultatas bus nusakomas klasifikavimo rezultatu. Kaip rezultato etalonas bus panaudotas ankstesnis klasifikavimo rezultatas, naudojant tiesioginio sklidimo neuronų tinklus (3.4 lentelė), kuomet į kiekvieną algoritmo žingsnį buvo paduota po lygų skaičių duomenų - 500 rinkinių.

5.2. Duomenų eksperimentams rengimas

Šiame poskyryje bus aptartas duomenų eksperimentams rengimas.

Pradžioje, bus pratestuotas dimensijų mažinimo algoritmas, mažinant pateikiamų duomenų skaičių. Kadangi struktūriškai buvo nuspręsta, kad kiekvienas lygmuo turi lygiai po 500 taškų,

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

		Taškų rinkinio skaičius					
		500	400	300	200	100	21
C1	Tikslumas	1.00	1.00	1.00	1.00	1.00	1.00
	Taiklumas	1.00	1.00	1.00	1.00	1.00	1.00
C2	Tikslumas	1.00	1.00	1.00	1.00	1.00	1.00
	Taiklumas	1.00	1.00	1.00	1.00	1.00	1.00

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

		Taškų rinkinio skaičius					
		500	400	300	200	100	21
C1	Tikslumas	1.00	1.00	1.00	1.00	1.00	1.00
	Taiklumas	1.00	1.00	1.00	1.00	1.00	1.00
C2	Tikslumas	1.00	1.00	1.00	1.00	1.00	1.00
	Taiklumas	1.00	1.00	1.00	1.00	1.00	1.00

testavimas bus pradėtas nuo 500 taškų rinkinių ir mažinamas kas 100 rinkinių, vadinasi, iš viso dimensijų mažinimo algoritmui bus pateikta 500, 400, 300, 200, 100 ir minimalus 21 rinkinių skaičius. Į kitus lygmenis bus pateikta po tiek pat rinkinių, kaip ir buvo nuspręsta struktūriškai analizuojant gautą algoritmą. Po dimensijų mažinimo žingsnio testavimo, rinkinių skaičius tokia pat metodika bus mažinamas klasifikatoriaus apmokymui ir klasifikatoriaus testavimo punktam.

5.3. Programos patikros rezultatai

Šiame poskyryje bus pateikti ir aptarti patikros rezultatai patikros rezultatai.

Duomenys patikrai buvo suformuoti tokiu būdu, koks buvo aptartas ankstesniame poskyryje. Patikros rezultatas bus įvertintas iš klasifikatoriaus darbo.

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

		Taškų rinkinio skaičius					
		500	400	300	200	100	21
C1	Tikslumas	1.00	1.00	1.00	1.00	1.00	1.00
	Taiklumas	1.00	1.00	1.00	1.00	1.00	1.00
C2	Tikslumas	1.00	1.00	1.00	1.00	1.00	1.00
	Taiklumas	1.00	1.00	1.00	1.00	1.00	1.00

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

		Taškų rinkinio skaičius					
		500	400	300	200	100	21
C1	Tikslumas	1.00	1.00	1.00	1.00	1.00	1.00
	Taiklumas	1.00	1.00	1.00	1.00	1.00	1.00
C2	Tikslumas	1.00	1.00	1.00	1.00	1.00	1.00
	Taiklumas	1.00	1.00	1.00	1.00	1.00	1.00

6. Rezultatų apibendrinimas

Darbo metu buvo ištirtos galimos žingsnio savybės, kuriomis remiantis galima sėkmingai atpažinti Parkinsono ligą pas subjektą. 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, sekantis pritaikytas žingsnis buvo galimų dimensijų priskyrimo metodų pritaikymas. Linijiniai PCA ir LDA reikiamo rezultato nepateikė. Geriausiai duomenis praskyrė LDA su Gauso branduoliu. Turimus vienmačius duomenis geriausiai klasifikavo tiesioginio sklaidimo dirbtinių neuronų tinklas.

Iš gautų klasifikavimo duomenų galima padaryti išvadą, kad klasifikatorius veikia ir veikia gerai. Užsibrėžtas tikslas buvo pasiektas - buvo suprojektuota sistema, kuri geba atpažinti Parkinsono ligą, remiantis žmogaus eisenos duomenimis.

Literatūra

- [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] 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.
- [3] 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.
- [4] T. Hofmann, B. Schölkopf, and A. J. Smola. Kernel methods in machine learning. *ArXiv Mathematics e-prints*, January 2007.
- [5] 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.
- [6] 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.
- [7] Maksim Norkin. Human activity recognition using Hidden Markov Models, September 2011.
- [8] 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.
- [9] 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.
- [10] 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.