



Kauno technologijos universitetas

Informatikos fakultetas

**Viršutinės kūno dalies sutrikimų aptikimo virtualioje
realybėje tyrimas**

Baigiamasis magistro projektas

Matas Vaitkevičius

Projekto autorius

Prof. dr. Tomas Blažauskas

Vadovas

Kaunas, 2026



Kauno technologijos universitetas

Informatikos fakultetas

**Viršutinės kūno dalies sutrikimų aptikimo virtualioje
realybėje tyrimas**

Baigiamasis magistro projektas

Dirbtinio intelekto informatika (6211BX007)

Matas Vaitkevičius

Projekto autorius

Prof. dr. Tomas Blažauskas

Vadovas

Prof. dr. Rytis Maskeliūnas

Recenzentas

Kaunas, 2026



Kauno technologijos universitetas

Informatikos fakultetas

Matas Vaitkevičius

Viršutinės kūno dalies sutrikimų aptikimo virtualioje realybėje tyrimas

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;
2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąrašė;
3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);
4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Matas Vaitkevičius

Patvirtinta elektroniniu būdu

Vaitkevičius, Matas. Viršutinės kūno dalies sutrikimų aptikimo virtualioje realybėje tyrimas. Magistro baigiamasis projektas / vadovas prof. dr. Tomas Blažauskas; Kauno technologijos universitetas, informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatikos mokslai, Informatika (B01).

Reikšminiai žodžiai: Virtualioji realybė, dirbtinis intelektas, rehabilitacija.

Kaunas, 2026. 60 p.

Santrauka

Šis baigiamasis magistro projektas skirtas viršutinės kūno dalies rehabilitacijos sistemos virtualiojoje realybėje kūrimui ir jai pritaikytų dirbtinio intelekto modelių tyrimui. Darbo eigoje buvo suprojektuota ir realizuota sistema, sudaryta iš virtualiosios realybės, dirbtinio intelekto ir reabilitologo sąsajos posistemių. Rankų judesiai buvo renkami naudojantis tik virtualiosios realybės įrangos plaštakų sekimo galimybėmis, todėl nėra reikalinga papildoma specializuota įranga, taip supaprastinant šį procesą.

Eksperimentų metu buvo nustatyta, kad pilna 26 kaulų plaštakos reprezentacija viršutinės kūno dalies sutrikimų aptikimui yra perteklinė, o poziciniai parametrai apsunkina modelių klasifikavimą. Klasifikavimo tyrimui buvo naudojami „LSTM“, „XGBoost“, „ST-GCN“ ir „ST-GIN“ modeliai. Geriausią tikslumą aptinkant sutrikimų aibes parodė „XGBoost“ ($\bar{F}_1 = 0,6100$), tačiau jo skaičiavimų trukmė, priešingai nei „LSTM“, yra per didelė, kad būtų galima klasifikavimą vykdyti realiu laiku. Sukurtas ansamblinis „LSTM-XGBoost“ modelis išlaiko aukštą tikslumą ($\bar{F}_1 = 0,6128$), o klasifikavimo laiką sumažina perpus.

Vaitkevičius, Matas. Research on Upper Body Impairment Detection in Virtual Reality. Master's Final Degree Project / supervisor prof. dr. Tomas Blažauskas; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Computer science, Informatics (B01).

Keywords: Virtual reality, artificial intelligence, rehabilitation.

Kaunas, 2026. 60.

Summary

This master's thesis focuses on the creation of upper body rehabilitation system using virtual reality and research of artificial intelligence models tailored for such a system. During this project, a system consisting of virtual reality, artificial intelligence and rehabilitation specialist interface subsystems was designed and developed. Hand tracking data was collected using only virtual reality hardware. For this reason, no additional specialised equipment was needed thus simplifying the process.

It was found that the hand representation of 26 bones is redundant for the use case of identifying upper body impairments. Similarly, positional features made it harder to classify the impairments. Classification experiments used "LSTM", "XGBoost", "ST-GCN" and "ST-GIN" models. "XGBoost" achieved the best performance when identifying the impairment combinations ($\bar{F}_1 = 0,6100$). However, unlike "LSTM", the model was too slow to run real-time. The proposed "LSTM-XGBoost" model manages to maintain high accuracy ($\bar{F}_1 = 0,6128$) while cutting the computational time in half.

Turinys

Lentelių sąrašas	7
Paveikslų sąrašas	8
Santrumpų ir terminų sąrašas	9
1. Įvadas.....	10
1.1. Projekto naujumas ir aktualumas.....	10
1.2. Tikslas ir uždaviniai	11
1.3. Dokumento struktūra	11
2. Viršutinės kūno dalies reabilitacijos sistemų analizė.....	12
2.1. Egzistuojančios sistemos	12
2.1.1. Virtualiosios realybės pratimai reabilitacijoje po insulto	12
2.1.2. Išmanioji gestų aptikimo pirštinė	14
2.1.3. Gestų klasifikavimas „Leap Motion“ įrenginiu.....	15
2.1.4. Nuotolinės reabilitacijos platforma	16
2.2. Dirbtinio intelekto metodai.....	19
2.2.1. Rekursiniai neuroniniai tinklai	19
2.2.2. Transformeriai	20
2.2.3. Sprendimų medžiai.....	20
2.2.4. Grafų konvoliuciniai tinklai	21
2.3. Posistemų įrankiai	21
2.3.1. Virtualiosios realybės posistemė	22
2.3.2. Dirbtinio intelekto posistemė	24
2.3.3. Reabilitologo sąsajos posistemė.....	25
3. Reikalavimų specifikacija	28
3.1. Nefunkcinių reikalavimų specifikacija.....	28
3.2. Funkcinių reikalavimų specifikacija.....	29
3.3. Techninių reikalavimų specifikacija.....	34
4. VR reabilitacijos sistemos ir DI modelių metodologija.....	36
4.1. Sistemos architektūra.....	36
4.2. Duomenų rinkimas ir apdorojimas	39
4.3. Dirbtinio intelekto modelių architektūra	44
4.3.1. Ilgos trumposios atminties modeliai.....	44
4.3.2. Erdviniai-laikiniai grafų konvoliuciniai tinklai	46
4.3.3. „XGBoost“	46
4.3.4. Ansamblis	47
5. Viršutinės kūno dalies judėjimo sutrikimų klasifikavimo rezultatai	48
5.1. Minimali kintamųjų aibė	48
5.1.1. Godus kaulų atmetimas	48
5.1.2. Pozicinių ir pasisukimo reikšmių atmetimas.....	50
5.2. Labiausiai tikėtino sutrikimo identifikavimas	50
5.3. Daugelio sutrikimų identifikavimas	53
Išvados	57
Literatūros sąrašas	59

Lentelių sąrašas

2.1 lentelė. VR posistemės įrankių analizė.....	23
2.2 lentelė. Dirbtinio intelekto posistemės įrankių analizė.....	25
2.3 lentelė. Reabilitologo sąsajos posistemės įrankių analizė	26
4.1 lentelė. Rankų duomenų failo pavyzdys.....	41
4.2 lentelė. Sesijas aprašantys bitai	41
5.1 lentelė. Svarbiausi „LSTM“ kaulai.....	49
5.2 lentelė. „XGBoost“ pirmi dešimt svarbiausių kintamųjų	49
5.3 lentelė. Modelių tikslumas atmetant pozicijos ir pasisukimo imtis.....	50
5.4 lentelė. Modelių rezultatai, identifikuojant labiausiai tikėtiną sutrikimą	51
5.5 lentelė. Modelių rezultatai, identifikuojant daugelį sutrikimų	53
5.6 lentelė. Ansamblinio modelio taisyklių rinkinys.....	54
5.7 lentelė. <i>F1</i> įverčiai kiekvienam sutrikimui.....	55
5.8 lentelė. Modelių vykdymo laikai apdorojant 600 kadru (20 sekundžių) seką.....	56

Paveikslų sąrašas

2.1 pav. Sukurti žaidimai [3]	13
2.2 pav. Integruota pirštinių sistema [7]	14
2.3 pav. Išmanioji pirštinė [7].....	15
2.4 pav. Apklauso dalyvių atstumas nuo gyvenamosios vietos iki artimiausio reabilitacijos centro [10]	17
2.5 pav. Apklauso dalyvių gyvenamasis aukštas [10]	17
2.6 pav. Vyresnio amžiaus apklauso dalyvių kompiuterinis raštingumas [10].....	18
2.7 pav. Fizioterapeuto vaizdas sesijos metu [10]	18
2.8 pav. Dviejų pacientų gydymo įvertinimas: aukščiau – mažiausias sulenkimo kampas tarp sesijų, žemiau – RMSE (<i>Root Mean Square Error</i>) įvertis tarp sesijų [10].....	19
2.9 pav. „LSTM“ sluoksnio vizualizacija.....	20
3.1 pav. Virtualiosios realybės posistemės panaudojimo atvejų diagrama	29
3.2 pav. Virtualūs pratimai: „Simon Says“ (a), atminties žaidimas (b), monetų skaičiavimas (c), žiedo perkėlimas (d).....	30
3.3 pav. "Pradėti sesiją" panaudojimo atvejo veiksmų diagrama.....	32
3.4 pav. „pradėti sesiją“ vaizdai iš reabilitologo perspektyvos (a) ir paciento perspektyvos (b).....	33
3.5 pav. Dirbtinio intelekto posistemės panaudojimo atvejų diagrama.....	34
4.1 pav. Diegimo diagrama.....	36
4.2 pav. Virtualiosios realybės posistemės paketų diagrama	37
4.3 pav. Sistemos serverio paketų diagrama	38
4.4 pav. Naudotojo sąsajos posistemės paketų diagrama	39
4.5 pav. Žaidimo objektų sekimo sistema	40
4.6 pav. Rankos modelis [20]	41
4.7 pav. Klasių pasiskirstymas	42
4.8 pav. Mokymo aibės sutrikimo kombinacijų surinktų kadru kieki prieš ir po DTW duomenų sintezės	43
4.9 pav. Kairės ir dešinės rankos duomenų pasiskirstymas (A) X ašyje, (B) Y ašyje ir (C) Z ašyje	44
4.10 pav. „LSTM“ architektūra	45
4.11 pav. Kairės plaštakos grafo struktūra	46
5.1 pav. Validavimo tikslumas godaus porų atmetimo metu	48
5.2 pav. „XGBoost“ susimaišymo matrica.....	51
5.3 pav. Atliekamo pratimo pavyzdys	52
5.4 pav. „LSTM“ susimaišymo matrica	52

Santrumpų ir terminų sąrašas

Santrumpos:

API – aplikacijos programavimo sąsaja (*angl. application programming interface*)

CNN – Konvoliucinis neuroninis tinklas (*angl. convolutional neural network, CNN*)

CSV – kableliais atskirtos reikšmės (*angl. comma separated values*)

DI – dirbtinis intelektas

DTW – dinaminis laiko iškraipymas (*angl. dynamic time warping*)

IMU – inercijos matavimo įrenginys (*angl. inertial measurement unit*)

LSTM – ilga trumpoji atmintis (*angl. long short-term memory*)

RMS – kvadratų vidurkio kvadratinė šaknis (*angl. root mean square*)

ST-GCN – erdvinis-laikinis grafų konvoliucinis tinklas (*angl. spatial-temporal graph convolutional network*)

ST-GIN – erdvinis-laikinis grafų izomorfizmo tinklas (*angl. spatial-temporal graph isomorphism network*)

VR – virtualioji realybė

Terminai:

XGBoost – atviro kodo biblioteka, kuri suteikia prieigą prie sprendimų medžiais grįstų optimizuotų metodų.

1. Įvadas

Reabilitacijos po insulto procesas yra ilgas ir reikalauja milžiniškų pastangų tiek iš paciento, tiek iš specialisto reabilitologo. Tačiau dėl žmogiškųjų išteklių trūkumo, klinikos neturi reikalingų galimybių skirti būtinojo dėmesio pilnam atsigavimui. Terapijai, dėl ribotų resursų, neretai yra skiriama mažiau laiko nei rekomenduotina, o didesnis dėmesis tenka apatinės kūno dalies pratimams. Tai lemia prastesnę viršutinės kūno dalies funkcijų atstatymą bei suprastėjusią gyvenimo kokybę ilguoju laikotarpiu.

Moksliniuose darbuose ši problema yra analizuojama dvejais aspektais. Vieni tyrimai kuria virtualiosios realybės sistemas, kuriuose pratimai yra pateikiami žaidimo formatu. Tai reabilitacijos procesą padaro įdomesniu, o pacientas pratimus gali atlikti su mažesne priežiūra. Kiti tyrimai naudoja specializuotą įrangą paciento judesių sekimui ir dirbtinio intelekto modelius gestų aptikimui. Tačiau tokių tyrimų, kurie apjungtų šias dvi sritis, trūksta.

Siekiant išspręsti šią spragą, šio tyrimo metu buvo kuriama viršutinės kūno dalies reabilitacijos sistema, kurią sudaro virtualiosios realybės, dirbtinio intelekto ir reabilitologo sąsajos posistemės. Čia pacientas pratimus gali atlikti virtualiojoje realybėje, o jo judesiai yra renkami dirbtinio intelekto modelių mokymo ir sutrikimų klasifikavimo tikslais. Duomenų rinkimas yra vykdomas pasitelkiant virtualiosios realybės įrangos plaštakų atpažinimo galimybėmis, todėl, priešingai nei panašiuose darbuose, nėra reikalinga papildoma brangi ir specializuota aparatūra.

Tyrimo metu buvo sprendžiamos dvi pagrindinės problemos. Pirmiausia, buvo ieškoma minimalios kintamųjų aibės, leidžiančios sumažinti duomenų saugojimo ir apdorojimo sąnaudas. Taip pat buvo tiriama galimybė aptikti viršutinės kūno dalies, t.y. rankų ir plaštakų, judėjimo sutrikimus naudojantis virtualiosios realybės aplinkoje surinktais duomenimis. Eksperimentuose skaičiavimai buvo atliekami naudojantis „LSTM“, „XGBoost“, „ST-GCN“ ir „ST-GIN“ modeliais. Iš jų geriausiai pasirodė pirmi du, todėl jie buvo papildomai kombinuojami į taisyklėmis grįstą ansamblinį modelį, apjungiantį „LSTM“ ir „XGBoost“ stipriąsias savybes – tikslumą bei galimybę skaičiavimus paskirstyti laike.

1.1. Projekto naujumas ir aktualumas

Reabilitacija po insulto yra sudėtinga, šie pacientai dažnai patiria dalinį viršutinės bei apatinės kūno dalies paralyžių, rijimo sutrikimus, bei sunkumus kalbant. Darbas, reikalingas grąžinti asmeniui pilnavertį gyvenimą, reikalauja daug pastangų tiek iš terapeuto, tiek iš paciento. Tačiau realybė yra tokia – dėl žmogiškųjų išteklių trūkumo, šiai problemai yra suteikiama per mažai laiko [1], o prioritetas yra skiriamas apatinei, o ne viršutinei kūno daliai [2].

Įvairios kuriamos virtualiosios realybės sistemos leidžia šį reabilitacijos procesą, kuris yra susijęs su viršutinės kūno dalies motorikos ir kognityvinių gebėjimų atstatymu, pajvairinti, pratimus pateikiant pramoginių, gyvenimo įgūdžių ar treniruočių žaidimų formatu. Šių užduočių atlikimas virtualiojoje realybėje ar tradicinėje aplinkoje neturi didelės įtakos galutiniam reabilitacijos rezultatui [3], tačiau įtraukčių žaidimų naudojimas gali paskatinti pastovumą, kuris yra reikalingas sėkmingam gebėjimų atstatymui ilguoju laikotarpiu [4].

Virtualiosios realybės sistemos taip pat suteikia tiesioginę prieigą prie daugelio dirbtinio intelekto mokymui reikalingų kintamųjų, tokių kaip judesio trajektorijos, sąnarių susilenkimo kampų ar sąnarių

kampinių greičių [5]. Šių kintamųjų ir dirbtinio intelekto algoritmų panaudojimas leistų palengvinti terapeutų darbą automatizuojant vieną ar kitą reabilitacijos procesą [1]. Tačiau ši sritis, t.y. virtualiosios realybės su integruotais dirbtinio intelekto metodais panaudojimas reabilitacijoje po insulto, vis dar yra nauja ir neišplėtota [6].

1.2. Tikslas ir uždaviniai

Tyrimo objektas – virtualiosios realybės ir dirbtinio intelekto metodų taikymas viršutinės kūno dalies judėjimo sutrikimų atpažinimui.

Šio projekto tikslas yra sukurti viršutinės kūno dalies motorikos lavinimo sistemą ir iširti dirbtinio intelekto panaudojimo galimybes identifikuojant judėjimo sutrikimus. Siekiant šį tikslą įgyvendinti yra iškeliami tokie uždaviniai:

1. Išanalizuoti egzistuojančius sprendimus,
2. apibrėžti judėjimo sutrikimų klases, pradinius bei išvestinius duomenis, kurie bus naudojami modeliams mokytis,
3. suprojektuoti ir realizuoti smulkiosios motorikos lavinimo virtualiosios realybės sistemos prototipą,
4. surinkti ir apdoroti mokymo duomenų aibę,
5. surasti įvesties parametrų aibę, kuri leistų minimizuoti saugomų duomenų kiekį bei modelių kompleksiskumą,
6. sudaryti ir išmokyti pradinės analizės modelį, kuris klasifikuotų paciento sutrikimus, bei atlikti eksperimentinius tyrimus, kurie įvertintų modelio kokybę.

1.3. Dokumento struktūra

Šiame dokumente yra pateikiama viršutinės kūno dalies reabilitacijos sistemų analizė, kurioje apžvelgiami egzistuojantys sprendimai, dirbtinio intelekto metodai bei įrankiai, kuriais bus kuriama sistema. Sekantis skyrius aprašo nefunkcinius, funkcinis ir techninius reikalavimus taikomus šiai sistemai.

Toliau yra aprašoma virtualiosios realybės reabilitacijos sistemai aktuali metodologija – sistemos architektūra, duomenų rinkimas ir jų apdorojimas bei panaudotų modelių praktiniai aprašymai, pritaikius kuriamai sistemai. Gale yra aprašomi įvykdyti eksperimentai ir jų rezultatai – modelių tikslumo pokyčiai atmetant kaulus ir tam tikrus įvesties poaibius bei sutrikimų klasifikavimo kokybę naudojant skirtingus modelius.

Rengiant šį rašto darbą generatyvinio dirbtinio intelekto įrankiai nebuvo naudoti.

2. Viršutinės kūno dalies reabilitacijos sistemų analizė

Šiame skyriuje yra pateikiama viršutinės kūno dalies reabilitacijos sistemų analizė – apžvelgiamos egzistuojančios sistemos, dirbtinio intelekto metodai bei posistemų įrankiai, kuriais bus kuriama ši sistema.

2.1. Egzistuojančios sistemos

Sistemų, kurios pasitelkia virtualiąją realybę, fokusuojasi į reabilitaciją po insulto ar naudoja dirbtinio intelekto metodus, galima rasti tikrai nemažai – egzistuoja virtualūs lavinimo žaidimai, gestus aptinkantys specializuoti įrenginiai ir nuotolinės reabilitacijos priemonės. Tačiau tokie sprendimai, kurie apimtų visas tyrimo dalis, šioje srityje yra nauji.

2.1.1. Virtualiosios realybės pratimai reabilitacijoje po insulto

Virtualiąją realybę, kaip įrankį, kuris gali palengvinti reabilitacijos po insulto kognityvinį procesą, tyrė Kinijos mokslininkai [3]. Šio tyrimo metu buvo sukurta programinė įranga, kuri veikimo principu yra labai panaši į tą, kurią kuriame šio projekto metu. Tyrimo metu buvo suformuotos dvi grupės – kontrolinė, kuri dalyvavo tradicinėse treniruotėse, ir virtualiosios realybės, kuri naudojo jų sukurta programinę įranga.

Abiejose grupėse dalyvavo iš viso 30 pacientų. Grupės dalyvių atrankai buvo išskirti tokie kriterijai – asmenys yra patyrę su insultu susijusius kognityvinius sutrikimus, o insultas atitinka Kinijos kardiologinių ligų prevencijos ir gydymo kriterijus, asmenys vyresni nei 60 metų bet jaunesni nei 90, insultas juos turėjo ištikti prieš daugiau nei 6 mėnesius, asmenys taip pat turėjo turėti bent 9 metų išsilavinimą. Be to, pradiniai kognityvinių gebėjimų testai turėjo atitikti tam tikras ribas.

Sudarant grupes taip pat buvo išskirti atmetimo kriterijai – asmuo nesupranta instrukcijų ir jų negali sekti, asmuo turi klausos, matymo ar protinių sutrikimų, yra patyrę epilepsijos priepuolių, patiria intensyvų svaigulį (*angl. vertigo*). Taip pat buvo atmetami asmenys, kurie tyrimo metu jau dalyvavo kituose tyrimuose.

Tyrimo metu visi 30 individų dalyvavo kasdieniuose 90 minučių trukmės fizioterapijos ir ergoterapijos užsiėmimuose. Kognityviniai pratimai, kurie dviem grupėms buvo skirtingi, truko 15 minučių per dieną, 6 kartus per savaitę iki 6 savaičių. Tradiciniai užsiėmimai apėmė šiuos pratimus: Šultė lentelės, atminties lavinimas atpasakojant nuotraukas, atminties ir sudėties veiksmų atlikimas iki 100, problemų sprendimo gebėjimo lavinimas pasitelkiant žaislinius blokelius. Virtualiosios realybės užsiėmimams buvo naudojama 16 žaidimų, o jie suskirstyti į 3 kategorijas – gyvenimiški įgūdžiai, treniruočių žaidimai (*angl. exergames, exercise games*) bei pramoginiai žaidimai (žr. 2.1 pav.). Asmenys patys pasirinkdavo kategoriją, kuri juos domino labiausiai, o terapeutas parinkdavo atitinkama žaidimą iš šios kategorijos.



(I)



(J)



(K)



(L)



(M)



(N)



(O)



(P)

2.1 pav. Sukurti žaidimai: (A–D) gyvenimiški įgūdžiai: maisto ruošą (A), langų valymas (B), perėjos kirtimas (C), ir gėlių laistymas (D); (E–H) treniruočių žaidimai: skvošas (E), priešlėktuvinių ginklų šaudymas (F), skrydis sklandytuvu (G), ir beisbolas (H); (I–P) pramoginiai žaidimai: kiaušinių mušimas (I), vabzdžių gaudymas (J), fejerverko uždegimas (K), „whack-a-mole“ (L), baliono priputimas (M), dangaus žibinto paleidimas (N), „Fruit Ninja“ (O), ir muilo burbulų sprogdinimas (P) [3].

Tyrimo metu buvo naudojami paprasti virtualiosios realybės įrenginiai, kurie pasižymėjo tik trijų laisvės ašių pajėgumais (t.y. įrenginys seka tik pasisukimą, bet ne poziciją).

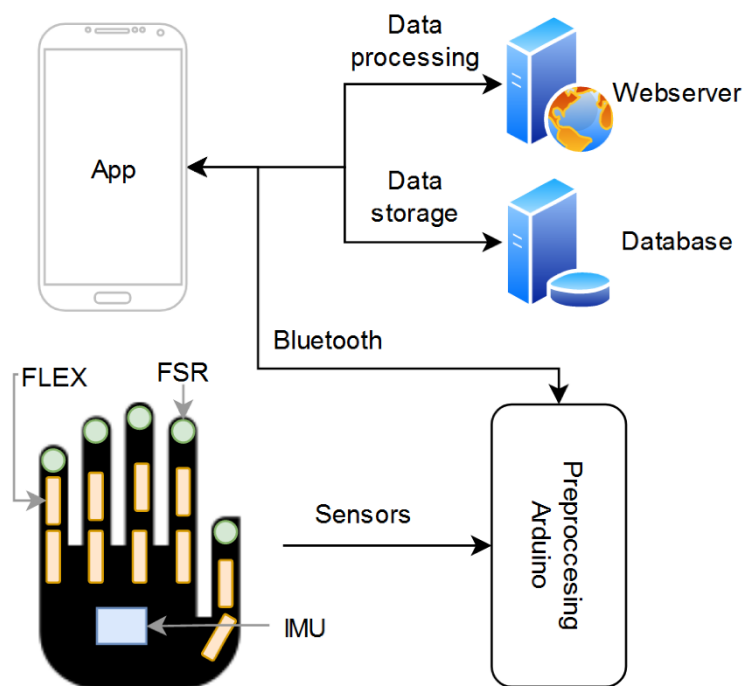
Prieš terapiją ir po šešių savaitių terapijos intervalo buvo atlikti gausūs kognityvinių gebėjimų testai. Daugumos, apart vieno, testų rezultatai buvo lygiaverčiai lyginant su tradicine terapija, todėl tyrėjai daro išvadą, kad virtualiosios realybės terapija yra naudingesnė vizualinės koordinacijos gebėjimų (*angl. visual-spatial cognitive characteristics*) reabilitavimui.

Po užsiėmimų pacientai taip pat užpildydavo klausimyną, kuris įvertindavo asmenų patiriamus šalutinius pojūčius, asmens istoriją bei tikimybę, kad tokią terapiją rekomenduotų kitiems. Du asmenys teigė patyrę nežymų svaigimą be vėmimo jausmo, dar du skundėsi išsausėjusiomis akimis ir trys teigė jaučiantys akių nuovargį. Kiti tyrime dalyvavę asmenys šalutinių efektų nepatyrė. Didžioji dauguma į klausimą ar rekomenduotų šią terapiją atsakė „sutinku“ (66,67 %) arba „labai sutinku“ (6,67 %). „Nesutinku“ arba „labai nesutinku“ neatsakė nei vienas.

Taigi, šis sprendimas patvirtina kuriamos sistemos aktualumą, įrodydamas, kad virtualiosios realybės įrenginių taikymas reabilitacijai yra lygiavertis tradiciniams metodams, o sprendimo naujumas paskatina pacientus naudotis įranga ir ją rekomenduoti kitiems.

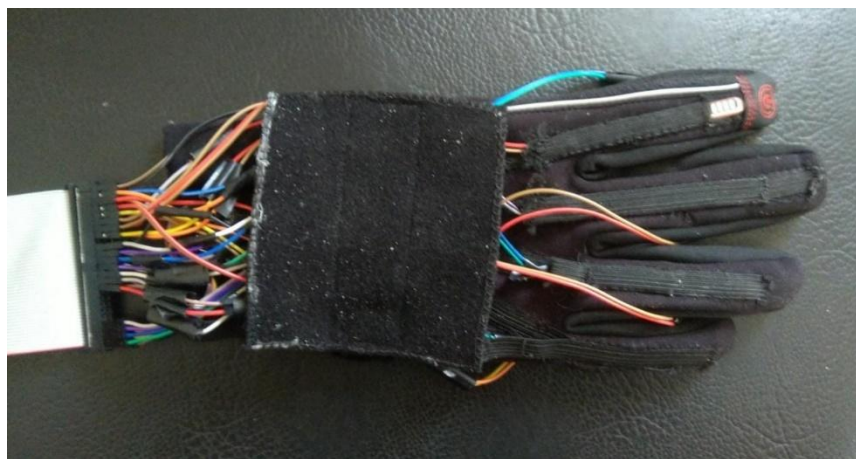
2.1.2. Išmanioji gestų aptikimo pirštinė

Straipsnio autoriai, siekdami palengvinti ir automatizuoti rankos judesių progreso įvertinimą po reabilitacijos nuotoliniu būdu, sukūrė išmaniąsias pirštines [7]. Išmaniosios pirštinės gali sekti asmens pirštų judesius, o dirbtinio intelekto modelis sugeba nustatyti asmens pajėgumus atlikti tam tikrus gestus. Šie duomenys yra perduodami terapeutui internetu, kaip iliustruota 2.2 pav.



2.2 pav. Integruota pirštinių sistema [7]

Šių autorių sukurta pirštinė turi šešiolika jutiklių. Dešimt iš jų yra sulenkimo (*angl. flex*) jėgą aptinkantys jutikliai, kurie naudojami asmens sąnarių sulenkimo kampui nustatyti. Penki – jėgos jutimo davikliai (*angl. force-sensing resistor, FSR*), skirti sugniaužimo bei griebimo jėgai nustatyti. Taip pat buvo naudojamas vienas inerciją matuojantis įrenginys (*angl. Inertial measurement unit, IMU*). Šis įrenginys turi galimybę sekti rankos poziciją šešiose laisvės ašyse (pozicija ir pasisukimas).



2.3 pav. Išmanioji pirštinė [7]

Siekdami sumažinti sesijos informaciją mašininio mokymosi algoritmams, straipsnio autoriai iš turimų duomenų išgavo svarbiausias savybes (*angl. feature extraction*). Sulenkimo jutikliai naudojo vidutinę reikšmę, vidutinės reikšmės kvadrato šaknį (*angl. root mean squared, RMS*), standartinį nuokrypį bei minimalią reikšmę. Jėgos jutimo davikliams buvo išgautos RMS, standartinio nuokrypio bei maksimalios reikšmės. IMU įrenginiui: vidurkis, RMS, standartinio nuokrypio bei signalo didžio ploto (*angl. signal magnitude area, SMA*) reikšmės.

Duomenims apdoroti ir gestams įvertinti straipsnio autoriai panaudojo du metodus – „XGBoost“ [8] bei logistinę regresiją. Straipsnyje tirti trys gestai – kumščio sugniaužimas, sužnybimas ir mojavimas. Kiekvienas gestas buvo įvertinimas 0–2 skalėje, kur 0 reiškia, kad gesto nepavyko padaryti, 1 pavyko dalinai, o 2 – gestas atliktas visiškai. Šiame tyrime dalyvavo trys pacientai, visi galėjo atlikti kumščio gestą, tačiau vienas pacientas nepajėgė atlikti sužnybimo ir mojavimo gestų.

Naudojant tik geriausiai veikiančias savybių kombinacijas (*angl. best performing feature combination, BPF*), „XGBoost“ pasiekė 53 % tikslumą atpažįstant kumštį, 78 % – sužnybimą ir 90 % – mojavimą. Logistinė regresiją pasiekė 87 % atpažįstant kumštį, 78 % – sužnybimą ir 70 % – tikslumą atpažįstant mojavimą. „XGBoost“ geriau sugebėjo atpažinti mojavimą, tačiau logistinė regresija geriau veikė kumščio atpažinimui.

Rankos sąnarių sekimą, tokį, kokį atliko straipsnio autoriai, galima atlikti be papildomos pirštinės kūrimo, naudojant tik virtualiosios realybės akinius. Turint prieigą prie sąnarių informacijos, mes galime duomenų rinkimą atlikti pigiau ir greičiau, naudojantis jau komerciškai pasiekiamą įranga.

2.1.3. Gestų klasifikavimas „Leap Motion“ įrenginiu

Vieno straipsnio autoriai susidūrė su labai panašia problema, kurią tiriamo mūsų darbo metu – jie taip pat kuria smulkiosios motorikos lavinimo, konkrečiai reumato reabilitacijos, sužaidybintą sistemą bei joje naudoja dirbtinio intelekto metodus [9]. Straipsnio autorių sukurta sistema naudojasi „Leap Motion“ įrenginiu, kuris geba sekti rankų sąnarių pozicijas, o tai jiems leidžia rinkti sesijos informaciją, fiksuoti gestus.

Naudotojai „Leap Motion“ įrenginiu formuoja rankų gestus, kurie yra atpažįstami dirbtiniu intelektu. Įvairūs gestai leidžia naudotojams valdyti žaidimą, o tai skatina rankų naudojimą ir reabilitaciją. Gestų aptikimui straipsnio autoriai naudoja du metodus – atsitiktinius miškus (*angl. random forest*) ir naivųjų Bajeso klasifikatorių (*angl. naive Bayes classifier*).

Testavimo metu straipsnio autoriai turėjo du gestų tipus – 1456 vieno tipo ir 38 antro tipo. Galima drąsiai teigti, kad jų testavimo aibė nėra subalansuota, o jų teigiamas pasiektas 98 % (atsitiktiniai miškai) ir 97 % (naivusis Bajeso klasifikatorius) tikslumas nėra objektyvus. Pavyzdžiui, naudojant atsitiktinius miškus, iš 38 antro tipo gestų vos 18 buvo atspėti teisingai. Naiviojo Bajeso klasifikatoriaus atveju, jis sugebėjo teisingai atskirti visus 38 antro tipo gestus, tačiau 49 pirmojo tipo gestai buvo neteisingai priskirti antrajam tipui. Taigi, dėl nesubalansuotos testavimo aibės į jų aprašytus rezultatus privalu žiūrėti skeptiškai. Iš jų klaidų taip pat galima ir pasimokyti, tad šiame projekte bus naudojami tokie įverčiai kaip F_1 , kurie padeda atsižvelgti į nesubalansuotas aibes.

Įdomu pastebėti tai, kad straipsnio autoriai užsimena apie dinaminį žaidimo sudėtingumo lygio keitimą naudojant mašininį mokymąsi, tačiau apie šią metodologiją straipsnyje nėra kalbama plačiau.

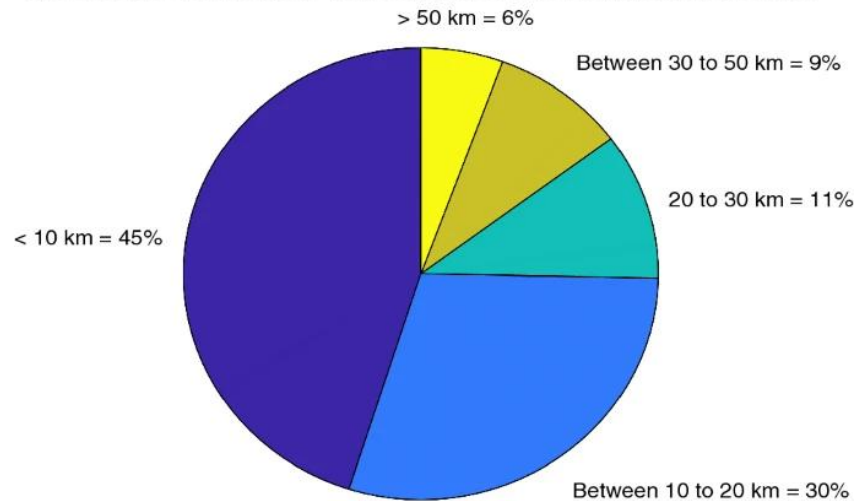
2.1.4. Nuotolinės reabilitacijos platforma

Vykstant COVID-19 pandemijai, vieno straipsnio autoriai buvo užsibrėžę sukurti nuotolinę reabilitacijos platformą, kuria naudojantis bet koku išmaniuoju įrenginiu, turinčiu kamerą, būtų galima sudalyvauti terapijos sesijoje [10]. Prieš pradėdami šios svetainės kūrimą, mokslininkai sukūrė apklausą siekdami įsitikinti šios poreikiu. Didžioji dalis apklaustųjų į ją atsakė elektroniniu būdu, dalis – raštiškai. Apklausą sudarė tokie klausimai:

1. Kiek Jums metų?
2. Gyvenate mieste, kaime ar vienkiemyje?
3. Koks atstumas iki artimiausio reabilitacijos centro?
4. Kokio tipo name gyvenate? (Kadangi tyrėjai siekė atskirti asmenis, kurie gyvena individualiuose namuose, nuo tų, kurie gyvena daugiabučiuose, jiems taip pat buvo aktualu būto aukštas.)
5. Ar turite asmeninį kompiuterį?
6. Kaip įvertintumėte savo kompiuterinį raštingumą?
7. Ar turite internetą namie?

Dalis apklausos rezultatų pateikiami toliau (2.4-2.6 pav.).

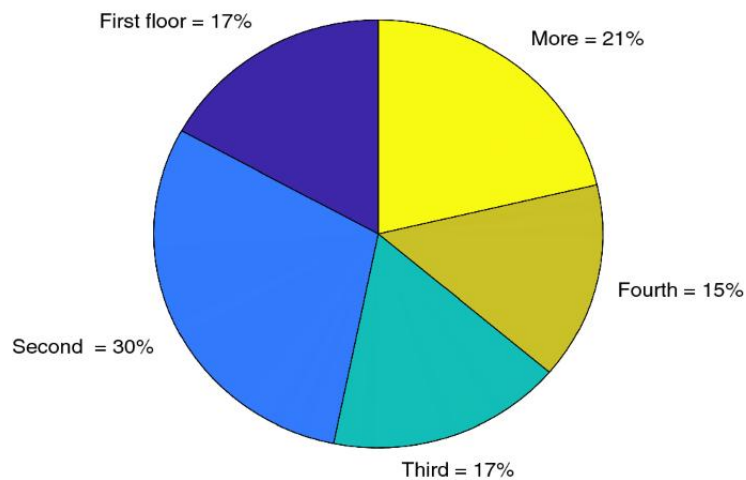
The distance between the home and the nearest rehabilitation center



2.4 pav. Apklaustos dalyvių atstumas nuo gyvenamosios vietos iki artimiausio reabilitacijos centro [10]

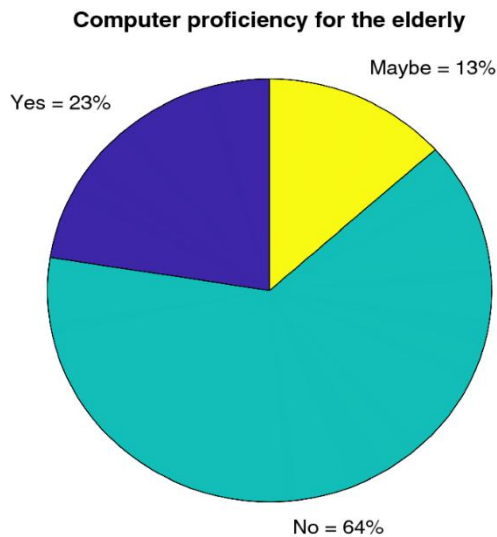
Didžioji dalis apklaustųjų (55 %) gyvena toliau nei 10 kilometrų nuo artimiausio reabilitacijos centro. Tai reiškia, kad tie asmenys, kuriems yra reikalinga reabilitacija, negali vairuoti patys, yra priklausomi nuo artimųjų, kad juos nuvežtų ir parvežtų iš klinikos. Nuotolinės reabilitacijos platforma šiems asmenims padėtų būti savarankiškesniems, leistų asmenims neturintiems artimųjų patogiau dalyvauti reabilitacijoje.

Apartment floor



2.5 pav. Apklaustos dalyvių gyvenamasis aukštas [10]

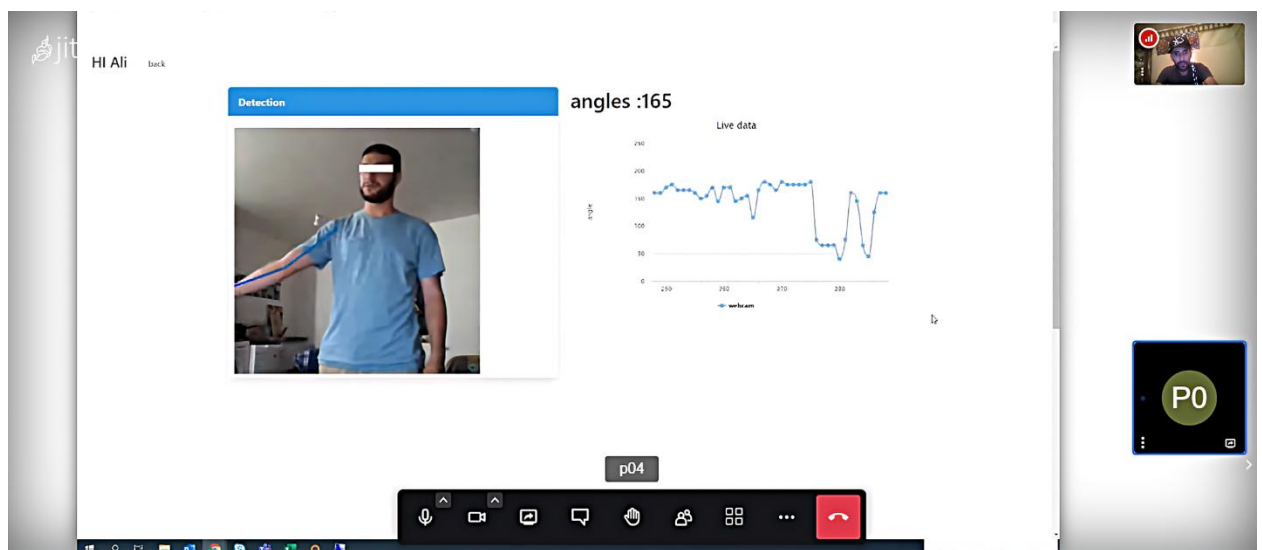
Tačiau net jei asmenys gyvena arti klinikos, išėjimo iš namų problema vis vien išlieka. Dauguma daugiaaukščių Alžyre, kur buvo parašytas šis straipsnis, yra senos statybos ir neturi liftų. Jei asmenys gyvena antrame aukšte ar aukščiau, yra senesnio amžiaus ar turi judėjimo sutrikimų, palikti savo būta tampa iššūkiu.



2.6 pav. Vyresnio amžiaus apklausos dalyvių kompiuterinis raštingumas [10]

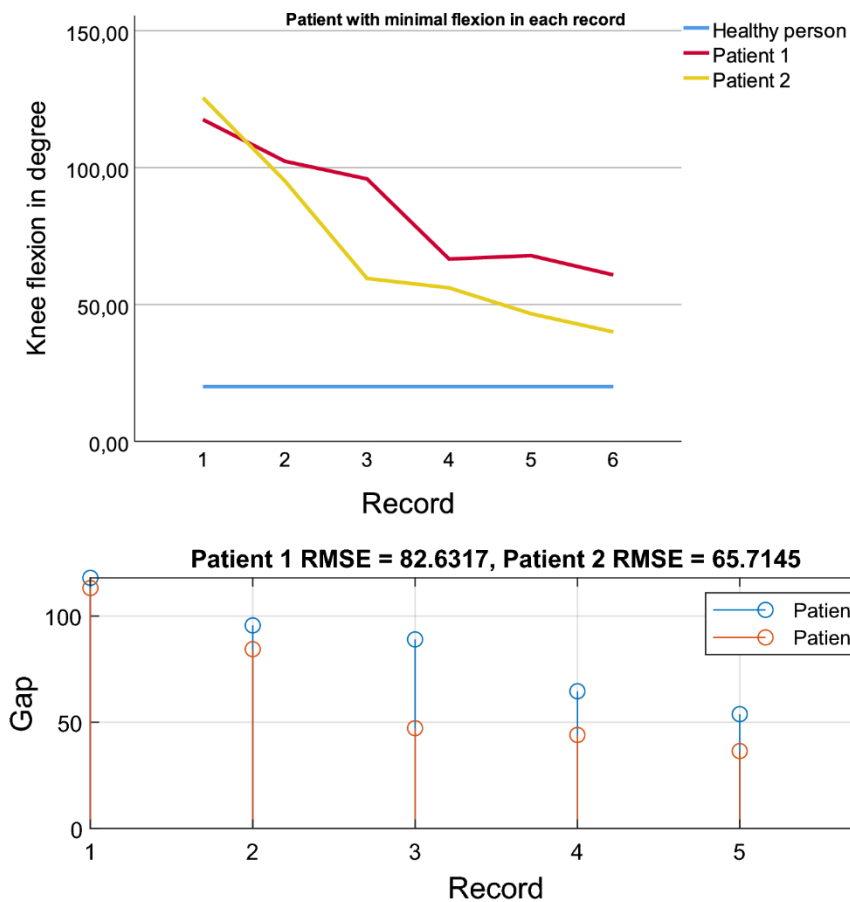
Verta pastebėti, kad didžioji dalis senyvo amžiaus apklausos dalyvių teigė, kad neturi tinkamų kompiuterinio raštingumo įgūdžių. Suprantama, tai tampa didele problema svetainės populiarinimui, kadangi šie asmenys yra pagrindinė naudotojų demografija. Tačiau nepaisant to, straipsnio autoriai nusprendė kurti šią sistemą.

Kuriant šią sistemą, darbo autoriai panaudojo kompiuterinį matymą siekdami automatiškai fiksuoti sąnarių kampus bei sekti juos laike (žr. 2.7 pav.).



2.7 pav. Fizioterapeuto vaizdas sesijos metu [10]

Šiam uždaviniui pasiekti straipsnio autoriai naudojo „MediaPipe“, Google sukurtą karkasą, kuris, naudodamas dirbtinį intelektą, kadre gali aptikti žmogaus sąnarių koordinatas (*angl. human pose estimation*). Straipsnio autoriai naudojo jau išmokytą modelį, alternatyvų nenagrinėjo, tikslumo netikrino.



2.8 pav. Dviejų pacientų gydymo įvertinimas: aukščiau – mažiausias sulenkimo kampas tarp sesijų, žemiau – RMSE (*Root Mean Square Error*) įvertis tarp sesijų [10]

Straipsnio autoriai įvykdė sistemos veikimo tyrimą, kaip matome iš 2.8 pav., pacientų būklė tarp terapijos sesijų gerėjo, rodmenys artėjo prie sveiko paciento įverčių. Verta pastebėti tai, kad tyrime nebuvo naudojama kontrolinė grupė, todėl palyginti ar šis progresas yra lygiavertis gyvoms sesijoms neįmanoma.

Nors šis sprendimas nėra techniškai inovatyvus, jo metu atlikta apklausa patvirtina kuriamos sistemos aktualumą. Apklausos rezultatai parodo poreikį išmaniai virtualiosios realybės reabilitacijos sistemai, kuri būtų automatizuota dirbtinio intelekto metodais – pacientams, turintiems mobilumo sutrikimų, atvykti į reabilitacijos įstaigą gali būti iššūkis, o tokia sistema leistų pratimus atlikti iš namų.

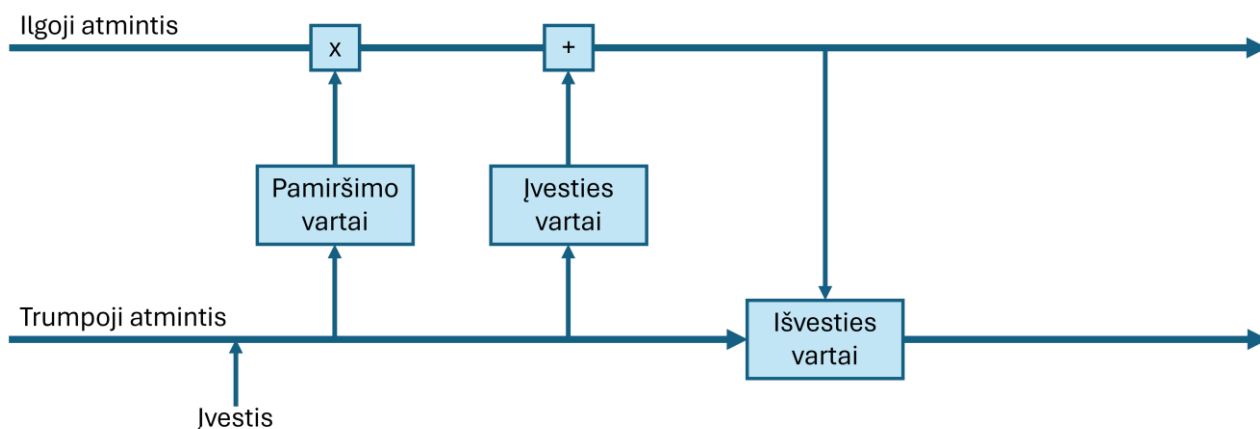
2.2. Dirbtinio intelekto metodai

Siekiant atpažinti sutrikimus viršutinės kūno dalies reabilitacijos sistemoje, panaudoti galima labai įvairius dirbtinio intelekto metodus ir modelius. Šiame skyrelyje bus apžvelgiami rekursiniai neuroniniai tinklai (*angl. Recursive Neural Network, RNN*), transformeriai, sprendimų medžiai bei grafų konvoliuciniai tinklai (*angl. Graph Convolutional Network, GCN*).

2.2.1. Rekursiniai neuroniniai tinklai

RNN – tai tokie neuroniniai tinklai, kurių praeito etapo išvestis gali būti iš naujo panaudojama kaip papildoma įvestis sekančio etapo metu. Šie modeliai pasižymi tuo, kad rekursiją galima atlikti tiek kartų, kiek yra elementų įvesties aibėje. Tai reiškia, kad šių modelių įvestis neturi tiesioginių dydžio suvaržymų, o mūsų sistemoje jie gali apdoroti kintamo ilgio reabilitacijos sesijas.

Tačiau paprasti RNN neturi efektyvių atminties mechanizmų, todėl ilgesnės sesijos yra linkusios susidurti su nykstančio ir sprogstančio gradiento problemomis. Jas išspręsti padeda ilgos trumposios atminties (*angl. Long Short-Term Memory, LSTM*) modifikacija [11], kuri įveda pamiršimo, įvesties ir išvesties vartų terminus. Pamiršimo vartai reguliuoja, koks procentas ilgosios atminties yra išsaugomas rekursinio etapo metu. Įvesties vartai valdo ilgosios atminties papildymą, o išvesties vartai apjungia trumpąją ir ilgąją atmintį kitam rekursiniam etapui (žr. 2.9 pav.).



2.9 pav. „LSTM“ sluoksnio vizualizacija

Toks modelis geriau geba apdoroti ilgesnes duomenų sekas, atsiminti senus įvykius. Tai yra ypatingai naudinga rehabilitacijos sistemai skirtam sutrikimų klasifikatoriui.

2.2.2. Transformeriai

Transformeriai [12] – tai vienas plačiausiai naudojamų šiuolaikinių modelių tipų. Jie taip pat geba apdoroti kintamojo dydžio sekas, tačiau, priešingai nei RNN, nėra rekursiniai. Šis skirtumas leidžia skaičiavimus paskirstyti lygiagrečiai, todėl sistemos, kurios naudoja transformerius, yra lengviau plečiamos horizontaliai.

Svarbiausia transformerių savybė yra dėmesio (*angl. attention*) mechanizmas, kuris leidžia modeliui įvertinti skirtingų sekos elementų svarbą tarpusavyje (žr. 1 lygtį). Dekodavimo metu, kai yra priimami sprendimai, jis leidžia atsižvelgti į aktualias sekos pozicijas, o tai gali lemti aukštesnį tikslumą.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

1 lygtis Dėmesio mechanizmas, kur Q yra užklausų matrica, K – raktų matrica, V – reikšmių matrica, o d_k – raktų vektoriaus dimensija

Tačiau transformerių modeliai yra dideli, o jų mokymas yra sudėtingas procesas, kuriam labai yra svarbus duomenų kiekis [13]. Siekiant išmokyti tokį transformeriais grįstą modelį, mums reikia daugiau duomenų nei jų turime šio projekto apimtyje. Dėl šios priežasties, transformeriai nėra naudojami eksperimentų metu.

2.2.3. Sprendimų medžiai

Sprendimų medžiai – tai vieni iš paprasčiausių dirbtinio intelekto modelių. Jie yra sudaromi generuojant binarinius rekursinius mazgus. Kiekvieno mazgo viduje duomenys yra atfiltruojami

viena ar kita kryptimi. Patys mazgai yra sudaromi sukuriant tokius sprendimus, kurie minimizuotų entropiją ar suteiktų didžiausią informacijos gavimą.

Verta pastebėti, kad sprendimų medžiai, priešingai nei RNN ar transformeriai, negeba tiesiogiai apdoroti kintamo dydžio aibių. Todėl dirbant su tokio tipo duomenimis, iš aibių turi būti išgaunamos statistinės reikšmės, tokios kaip vidutinė reikšmė, standartinis nuokrypis, minimalios ir maksimalios reikšmės ir panašiai. Iš vienos pusės, tai sumažina kintamųjų skaičių – aibės yra sutraukiamos į trumpus aprašymus. Kita vertus, tai gali paslėpti smulkesnius judesius, kurie gali būti naudingi viršutinės kūno dalies sutrikimų klasifikavimui.

Miškai – tai irgi sprendimų medžiais grįstas metodas. Pagrindinis skirtumas yra tas, kad sprendimo priėmimo dalyvauja daugiau nei vienas unikalūs medis, o galutinis rezultatas yra nusprendžiamas demokratinio balsavimo būdu. Tai leidžia atskiriems medžiams išmokyti skirtingą informaciją, o bendram modeliui geriau prisitaikyti prie išimčių. Šia modifikacija yra pagrįstas ir „XGBoost“ [8], kuris mišką formuoja iteraciniu būdu, kiekvieną kartą atsižvelgiant į prieš tai darytas klaidas.

2.2.4. Grafų konvoliuciniai tinklai

CNN yra ypatingai naudingi dirbant su duomenimis, kurie yra pateikiami langelių formatu, pavyzdžiui, nuotraukomis. Konvoliucija sumažina vaizdo apimtį, o tai leidžia efektyviau atpažinti objektus, klasifikuoti vaizdus. Tačiau tokia pati mintis gali būti pritaikoma ir ne tik vaizdams, bet ir grafams ar kaulų struktūroms.

Jei kiekvieną plaštakos sąnarį laikome kaip grafo viršūnę, o kaulus kaip grafų kraštines, mes galime sudaryti gretimumo matricą (*angl. adjacency matrix*), kuria naudojant grafų konvoliuciniai tinklai išveda kiekvieno sąnario kaimyninių reikšmių vidurkį [14]. Kaip kaimynus grafiui galima priskirti ir gretimus kadrus.

Tyrimo metu taip pat buvo naudojama GCN modifikacija – grafų izomorfizmo tinklai (*angl. Graph Isomorphism Network, GIN*) [15]. Vietoje to, kad grafų kaimyninės reikšmės būtų išvedamos į vidurkį, GIN šias reikšmes susumuoja ir jas panaudoja kaip įvestį papildomam neuroniniam tinklui, kas yra matoma 2 lygyje. Teoriškai, ši modifikacija turėtų būti stipresnė nei paprasti GCN tinklai.

$$h_v^{(k)} = MLP^{(k)} \left((1 + \epsilon^{(k)}) \cdot h_v^{(k-1)} + \sum_{u \in N(v)} h_u^{(k-1)} \right)$$

2 lygtis GIN sluoksnio formulė, kur $MLP^{(k)}$ yra daugiasluoksnis perceptronas, $h_v^{(k-1)}$ yra prieš tai buvęs sluoksnis, $N(v)$ yra taško kaimynai, o $\sum_{u \in N(v)} h_u^{(k-1)}$ yra kaimyninių reikšmių suma.

2.3. Posistemių įrankiai

Šio magistrinio projekto darbo metu kuriama sistema susidės iš trijų posistemių, kurias reikės sukurti, suderinti ir apjungti tarpusavyje. Siekiant realizacijos etapą padaryti kuo lengvesnį, mums yra svarbu iš anksto numatyti kuriamų posistemių naudojamas technologijas – tai leidžia mūsų žingsnius planuoti į ateitį.

2.3.1. Virtualiosios realybės posistemė

Pirmoji posistemė, kuri bus sukurta – virtualiosios realybės posistemė. Šioje posistemėje bus praleidžiama daugiausiai laiko, pacientai atliks pratimus, bus vykdoma terapija ir renkami duomenys. Dėl šių priežasčių mums yra svarbu sukurti tokį įrankį, kuris būtų patogus naudoti, turėtų visas mums reikalingas funkcijas. Šios posistemės kūrimui tinka visi žaidimų varikliai, kurie palaiko virtualiosios realybės aplikacijų kūrimą, gestų aptikimą. Šiuo metu rinkoje tokie populiariausi yra trys – „Unity“, „Unreal Engine“ bei „Godot“.

Populiariausias variklis, turintis daugiausiai projektų ir žaidimų – „Unity“. Jis suteikia visas reikalingas funkcijas siekiant sukurti pilnavertę terapijos įrangą. Šis variklis yra puikiai sudokumentuotas, pasižymi gausa pagalbinių išteklių, modelių ir įskiepių prieinamumu „Unity Asset Store“. Didžiausias šio variklio minusas yra kaina – „Unity“ savo kūrėjus apmokestina nuo pakankamai žemos kartelės, t.y. vos \$200 000. Dėl to reikia atsižvelgti į mūsų kuriamos sistemos apimtį.

Sekantis populiarus pasirinkimas yra „Unreal Engine“. Tai galingas variklis, naudojamas naujausių ir galingiausių žaidimų kūrimui. Jis pasižymi itin aukštą grafine kokybe, beveik trigubai didesniu „Unreal Marketplace“ siūlomų produktų skaičiumi. Kita vertus, aukšta grafinė kokybė, mūsų atveju, nėra prioritetas. Priešingai, tai gali reikšmingai sumažinti virtualiosios realybės posistemės kadru per sekundę įvertį.

Šią problemą galima būtų išspręsti naudojantis lengvo svorio variklį „Godot“. Tai pakankamai naujas, bet mažiau populiarus variklis. Tačiau šio variklio naujumas mums sukeltų papildomų problemų – virtualiosios realybės palaikymas darbo pradžios metu dar nebuvo iki galo išvystytas.

Atlikus išsamią įrankių analizę (žr. 2.1 lentelė) buvo nuspręsta naudoti „Unity“ variklį. Nors tai yra uždaro kodo ir potencialiai mokamas įrankis, jo suteikiamos funkcijos virtualiosios realybės srityje mums leis užtikrinti galimybę laiku sukurti siekiamą posistemę, kuri veiks ir su paprastesniais virtualiosios realybės įrenginiais.

2.1 lentelė. VR posistemės įrankių analizė

Kriterijus	Unity	Unreal Engine	Godot
Kaina	2030€/metus vienam programuotojui (jei metinės pajamos viršija \$200 000)	5 % nuo pardavimų, viršijančių \$1 000 000	Nemokama
Atviro kodo	Ne	Ne	Taip
Programavimo kalba	C#	C++	GScript, palaiko C#
Naudojimo lengvumas	Daug funkcijų ir lankstumo, dideli pagalbinių išteklių	Galingas variklis, reikia įgūdžių pilnam išnaudojimui	Lengvas variklis su paprasta kodavimo kalba
Įskiepių ir modelių prieinamumas	„Unity Asset Store“ suteikia gausų nemokamų ir mokamų resursų pasirinkimą (117 901 produktai [16])	„Unreal Marketplace“ suteikia gausų nemokamų ir mokamų resursų pasirinkimą (~342 000 produktų [17])	„Asset Library“ suteikia mažą pasirinkimą (5 016 produktai [18])
Išleistų žaidimų skaičius Steam platformoje [19]	63 183	19 696	4 048
Optimizavimas mobiliems įrenginiams	Veikia puikiai su dauguma VR realybės įrenginių	Veikia prastai su įrenginiais, turinčiais mažesnius skaičiavimo pajėgumus	Dėl lengvo svorio, veikia puikiai su dauguma įrenginių
VR/XR palaikomumas	UnityXR biblioteka suteikia reikalingą VR prieigą	Unreal Engine suteikia visą reikalingą prieigą	Godot suteikia prieigą prie bazinių VR funkcijų, tačiau ji yra eksperimentinė, trūksta funkcionalumo
Gestų palaikymas	XR Hands suteikia visą reikalingą prieigą prie rankų atvaizdavimo	Variklis palaiko gestus	Gestus palaiko, tačiau funkcionalumas yra eksperimentinis

2.3.2. Dirbtinio intelekto posistemė

Sukūrus virtualiosios realybės posistemę jau galima būtų kurti dirbtinio intelekto posistemę bei atlikti šių metodų tyrimą. Tačiau, čia taip pat yra svarbu pasirinkti tinkamus įrankius – lanksčios bibliotekos mums leis atlikti eksperimentus greičiau ir užtikrinčiau. Kita vertus, netyčia pasirinkus netinkamą kalbą, įrankius ar bibliotekas, mums gali tekti tvarkyti jų sukurtas problemas, vietoje to, kad visą dėmesį skirtume tyrimams.

Taigi, prieš pradėdant šios posistemės kūrimą, mes turime atsakyti kelis svarbius klausimus. Visu pirma, mums reikia žinoti ar mes norime likti prie .NET architektūros – tokiu atveju dirbtinio intelekto modelius, kurie buvo sukurti C# kalboje, būtų galima naudoti lokaliai virtualiosios realybės įrenginyje. Jei mūsų atsakymas yra „taip“, mes turime nuspręsti ar modelius kursime patys, nuo nulio, ar pasitelksime tokiems „Unity“ karkasais, kaip „Sentis“.

Kita vertus, pasirinkus Python ekosistemą, mes turėtume prieigą prie populiariausių dirbtinio intelekto karkasų, tokių kaip „TensorFlow“ ar „PyTorch“, kurie leistų greitai koreguoti mūsų modelius, išbandyti skirtingas realizacijas.

Verta pastebėti, kad modelius, sukurtus Python aplinkoje, yra įmanoma konvertuoti į „ONNX“ formatą, kurį palaiko anksčiau minėtas „Unity“ karkasas „Sentis“. Tai reiškia, kad modelius sukurti ir išmokyti galime Python aplinkoje, o tokiu atveju, jei eksperimentai yra sėkmingi, modelius panaudoti praktiškai būtų galima jau lokaliai, virtualiosios realybės akiniuose.

Tačiau modelius leidžiant lokaliai, mes galime susidurti su išteklių nepakankamumo problema – nors virtualiosios realybės akiniai turi pakankamai galingus grafinius procesorius, jie ir taip yra apkrauti virtualiosios aplinkos atvaizdavimo darbu. Siekiant užtikrinti stabilią smulkiosios motorikos lavinimo sistemos greitaveiką, modelius geriausia būtų paleisti debesyse. Šį funkcionalumą mes galime pasiekti naudojant PyFlask biblioteką. Ši mums leistų sukurti atskirą serverį su aplikacijų programavimo sąsaja (*angl. application programming interface, API*), kuris suteiktų tiesioginę prieigą prie dirbtinio intelekto modelių. Tai mums leistų apdoroti siunčiamus duomenis nepriklausomai nuo virtualiosios realybės įrenginio skaičiavimų pajėgumo.

Taip pat yra verta pastebėti, kad modelių prieinamumas per aplikacijų programavimo sąsają mums suteiktų didelį lankstumą. Ateityje mes nebūtume pririšti prie „Unity“ realizacijos, o duomenis būtų galima pasiekti per skirtingas naudotojo sąsajas, kas gali būti naudinga suteikiant terapeutams pilną prieigą prie informacijos, rezultatų ir modelių.

Bendra dirbtinio intelekto posistemės įrankių analizė pateikiama toliau (žr. 2.2 lentelė). Iš jos sprendžiamame, kad mūsų geriausi pasirinkimai yra arba „PyTorch“, arba „TensorFlow“ su „PyFlask“ integracija. Abu yra populiarūs ir kokybiški karkasai, kuriuos būtų galima pilnai panaudoti mūsų tikslams. Galų gale modelių kūrimui buvo pasirinktas „TensorFlow“ karkasas – šį sprendimą lėmė turima patirtis programuojant modelius Python aplinkoje.

2.2 lentelė. Dirbtinio intelekto posistemės įrankių analizė

Kriterijus	C# modelis nuo nulio	Sentis	PyTorch	TensorFlow
Kalba	C#	C#	Python	Python
Kūrimo greitis	Lėtas, reikia sukurti viską	Greitas, pilnai integruotas „Unity“ aplinkoje	Greitas	Greitas
Galimybė išmokyti	Įmanoma	Nėra, reikia importuoti modelius	Yra	Yra
GPU palaikymas	Įmanoma	Taip	Taip	Taip
„Unity“ integracija	Taip	Taip	Ne	Ne
Integracija į egzistuojantį įmonės C# API	Taip	Ne	Ne	Ne

2.3.3. Reabilitologo sąsajos posistemė

Siekiant reabilitacijos procesą padaryti patogesnę tiek pacientams, tiek reabilitologams, yra reikalinga ir pastarųjų sąsajos posistemė. Naudojantis šia sąsaja, specialistas gali koreguoti užsiėmimo procesą – pradėti sesiją ir susieti jos duomenis su konkrečiu pacientu, valdyti jo poziciją virtualioje aplinkoje ir pateikiamus pratimus.

Prieš pradėdant kurti šią posistemę, lygiai taip pat reikia teisingai pasirinkti įrankius, kuriais sieksime šio tikslo. Grafinių sąsajų kūrimo įrankių pasirinkimas yra labai didelis, o visų alternatyvų apžvelgti praktiškai nėra įmanoma. Dėl šios priežasties apžvalgoje apsiribosime keliais pasirinkimais – bus apžvelgta konsolinė aplikacija, „TKinter“, „Flutter“ bei „MAUI“.

Paprasčiausia ir greičiausia realizacija būtų įmanoma kuriant primityvią konsolinę aplikaciją. Tai mums leistų sutaupyti laiko bandant integruoti posistemę prie dirbtinio intelekto posistemės. Tačiau toks variantas tinkamas tik prototipų kūrimui – mūsų tikslinei auditorijai, t.y. fizioterapeutams, tokios sistemos naudojimas nebūtų patogus.

Norinti pilnai išlikti Python ekosistemoje būtų galima pasinaudoti „TKinter“ biblioteka. Ji mums leistų naudotojo sąsają pilnai integruoti į dirbtinio intelekto posistemę. Naudojantis šia biblioteka mes galėtume kurti paprastus informacinius langus. Tai būtų optimaliausias sprendimas siekiant sukurti balansą tarp kūrimo sudėtingumo bei kuo geresnio panaudojamumo galutiniam vartotojui. Tačiau būtina pastebėti, kad Python aplikacijas mes galime paleisti tik „Windows“, „Linux“ ir „macOS“

platformose. Siekiant aplikaciją padaryti kuo lengviau prieinamą, mums reikia tokių įrankių, kurie palaikytų mobiliuosius įrenginius.

Vienas iš įrankių, kuris palaiko didelę aibę platformų yra „Flutter“. Šiuo įrankiu sukurtos programos gali būti sukompilijuojamos „iOS“, „Android“, „Windows“, „Linux“ bei „macOS“ platformoms. Šis įrankis taip pat suteikia galimybę kurti internetines aplikacijas, kurias būtų galima naudoti nepriklausomai nuo platformos. Naudojantis „Flutter“ įrankiu mes taip pat galėtume kurti aplikacijas, kurios turi įvaizdį, tinkamą tikram, išbaigtam ir galutiniam produktui. Savaime aišku, nušlifotos naudotojo sąsajos kūrimas užtruktų daugiau laiko nei konsolinės aplikacijos kūrimas. Tačiau tai būtų vertinga investicija, siekiant užtikrinti, kad sukurta programa bus panaudota mūsų tikslinės auditorijos.

Alternatyva „Flutter“ įrankiui galėtų būti „Microsoft“ kompanijos kuriamas „MAUI“. Šis taip pat palaiko gausią platformų aibę, galima programą kurti „iOS“, „Android“, „Windows“, „macOS“ įrenginiams. Deja, šis įrankis nepalaiko „Linux“ įrenginių, negalima kurti internetinių aplikacijų. Taigi, pasirinkdami „MAUI“ mes apribotume savo galimybes vystyti kuriamą sistemą ateityje.

Gale buvo atlikta išsami įrankių, kuriuos būtų galima panaudoti naudotojo sąsajos posistemės kūrimui, analizė (žr. 2.3 lentelė). Konsolinė aplikacija ir „TKinter“ buvo atmesti dėl savo minimalaus funkcionalumo, mobilių įrenginių nepalaikymo. Abi alternatyvos, t.y. „Flutter“ ir „MAUI“, buvo lygiai tinkamos galutiniam produktui. Pasirinkimą tarp šių dviejų platformų lėmė tik asmeniniai įgūdžiai – su „Flutter“ platforma buvo susidurta anksčiau, joje jau yra kuriama dalis įmonės įrankių. Tai reiškia, kad sistema nebūtų kuriama nuo nulio, mes turėtume atspirties tašką.

2.3 lentelė. Reabilitologo sąsajos posistemės įrankių analizė

Kriterijus	Konsolinė aplikacija	TKinter	Flutter	MAUI
Kalba	Python	Python	Dart	C#
Kūrimo sudėtingumas	Labai paprasta	Paprasta	Sunkiau	Sunkiau
Naudojimo sudėtingumas	Sunku, netinka terapeutams	Minimalus funkcionalumas gali pakenkti panaudojamumui	Kuriamos modernios aplikacijos	Kuriamos modernios aplikacijos
Grafiniai pajėgumai	Maži	Minimalūs, palaikomi mygtukai bei paprastos drobės	Modernūs, palaiko visą reikalingą funkcionalumą	Modernūs, palaiko visą reikalingą funkcionalumą
Palaikomos platformos	Windows, Linux, macOS	Windows, Linux, macOS	iOS, Android, Web, Windows, Linux, macOS	iOS, Android, Windows, macOS

Kriterijus	Konsolinė aplikacija	TKinter	Flutter	MAUI
Integracija su DI sisteme	Tiesioginė, abu Python	Tiesioginė, abu Python	Per API	Per API
Atviro kodo	Taip	Taip	Taip	Taip

3. Reikalavimų specifikacija

Siekiant sukurti programą, kuri būtų naudinga ir tinkama galutiniam vartotojui, kad žinotume kuris funkcionalumas mums yra svarbus, mes negalime apsieiti be reikalavimų specifikacijos. Šiame skyrelyje bus išnagrinėti tiek funkciniai, tiek nefunkciniai reikalavimai, aprašytos išsamios jų specifikacijos. Taip pat bus išnagrinėta techninė reikalavimų specifikacija susijusi su virtualiosios realybės įrenginiais.

3.1. Nefunkcinių reikalavimų specifikacija

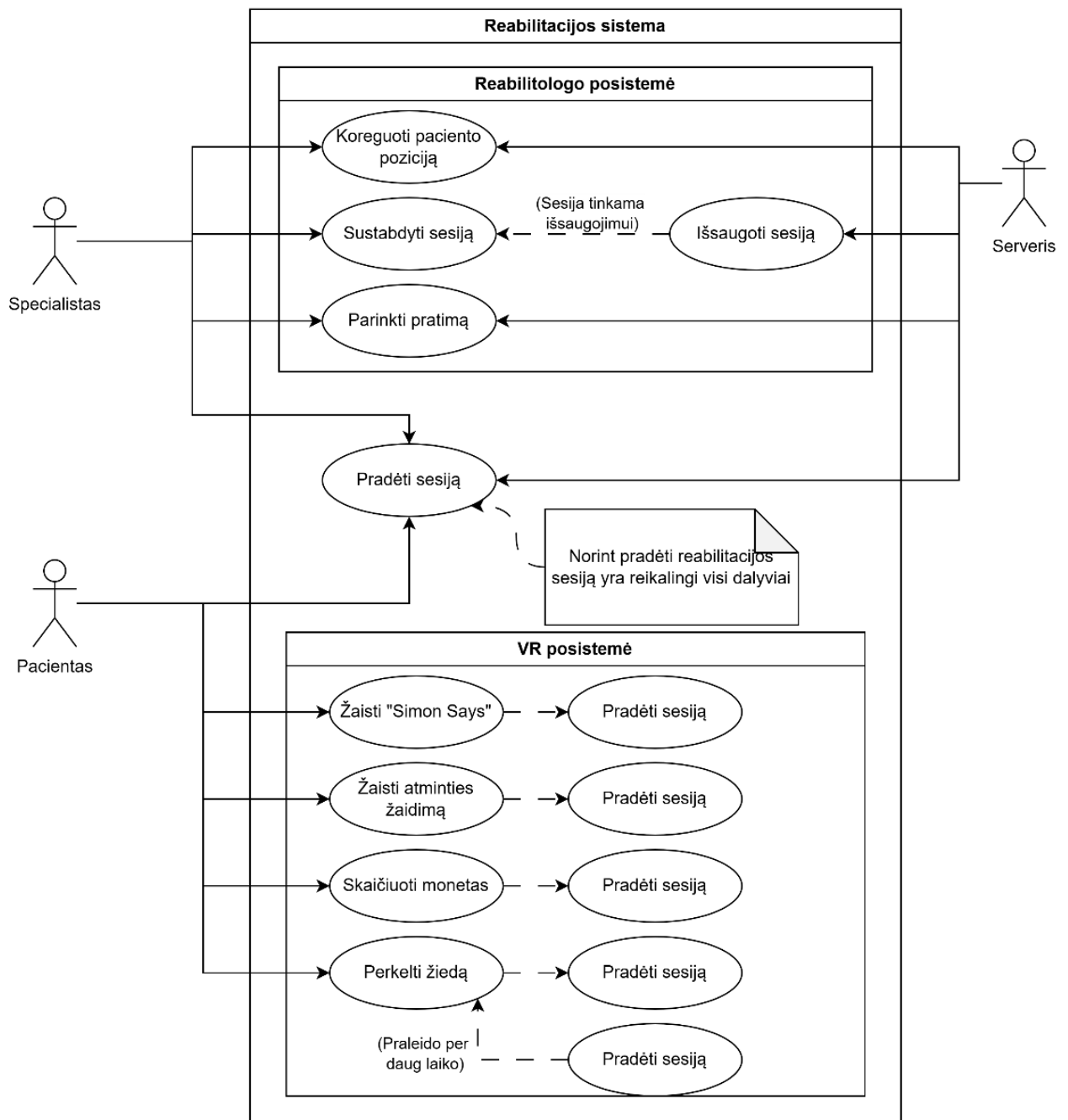
Kuriant smulkiosios motorikos lavinimo sistemą buvo užsibrėžti tokie nefunkciniai reikalavimai, kurie užtikrins kuriamos sistemos kokybiškumą:

1. naudotojo sąsaja turi būti intuityvi ir prieinama visiems naudotojams:
 - a. virtualiosios realybės naudotojo sąsaja turi turėti aiškias atsakomąsias reakcijas – mygtukų paspaudimai ar sąveika su virtualiosios realybės objektais pasitelkia spalvines ir garsines atsakomąsias reakcijas,
 - b. virtualiosios realybės naudotojo sąsajoje tekstas yra naudojamas minimaliai, šriftas turi būti didelis, o spalvos kontrastingos,
 - c. virtualiosios realybės naudotojo sąsaja turi būti pilnai prieinama iš sėdimos pozicijos,
 - d. dirbtinio intelekto naudotojo sąsajos posistemė turi būti aiški visiems naudotojams – čia neturi būti naudojamas techninis žargonas,
 - e. atliekant vieną veiksmą dirbtinio intelekto naudotojo sąsajoje turi būti panaudoti ne daugiau nei trys langai,
2. virtualiosios realybės sistema turi minimizuoti tikimybę, kad sistemos naudojimas sukels pykinimo pojūtį, kuris yra dažnas simptomas naudojantis virtualiosios realybės akiniais:
 - a. virtualiosios realybės scenose objektai, kurie yra fone, turi judėti minimaliai,
 - b. virtualiosios realybės sistema turi išlaikyti stabilų arba didesnę nei 90 kadru per sekundę atnaujinimo greitį,
 - c. virtualiosios realybės aplinkoje dirbtinio judėjimo (t.y. kai judama ne fiziškai, o pasitelkiant išorines įvestis) turi būti kuo mažiau,
 - d. jei virtualiosios realybės aplinkoje yra naudojamas dirbtinis judėjimas, jis turi būti diskretus, o ne tolyginis,
 - e. virtualiosios realybės aplinkoje atliekami pratimai turi būti trumpi (mažiau nei 5 minutės), siekiant leisti naudotojui daryti pertraukas,
 - f. virtualiosios realybėje atliekamus pratimus turi būti įmanoma atlikti ir papildytos realybės aplinkoje,
3. naudotojo sąsają turi būti galima paleisti tiek „Android“, tiek „Windows“ įrenginiuose,

4. dirbtinio intelekto posistemė turi gebėti palaikyti daugiau nei vieno naudotojo užklaudas rezultatų gavimui,
5. dirbtinio intelekto posistemėje vienu metu mokomas tik vienas modelis.

3.2. Funkcinių reikalavimų specifikacija

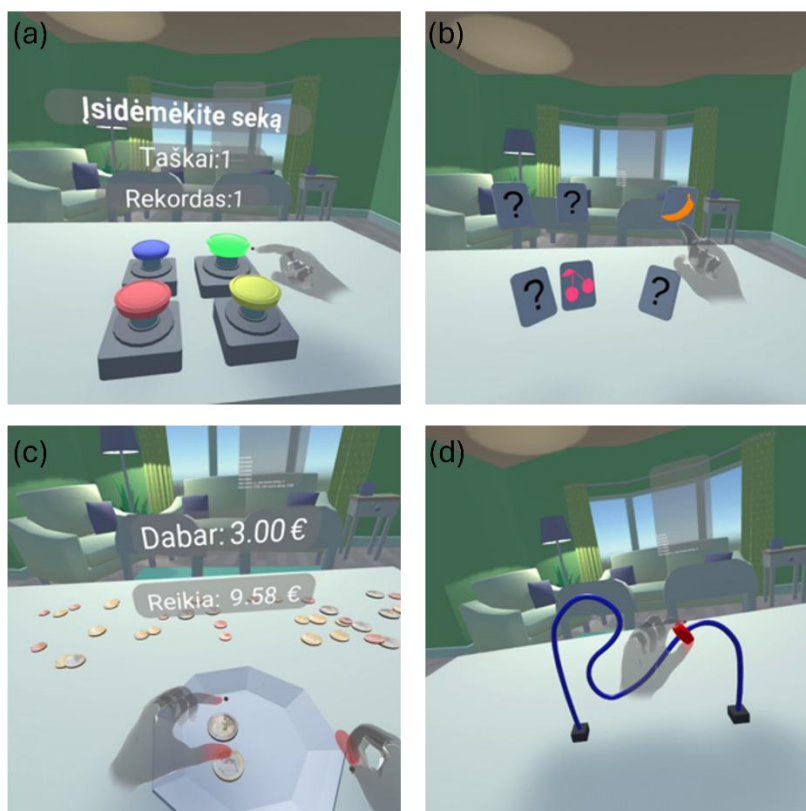
Siekiant tikslingai panaudoti turimą laiką kuriant šią sistemą buvo reikalinga funkcinių reikalavimų specifikacija, nusakanti tikslų funkcionalumą, kurį turi vykdyti galutinis produktas. Ši funkcinių reikalavimų specifikacija buvo aprašyta „UML“ panaudojimo atvejų diagramomis (3.1 pav. ir 3.5 pav.).



3.1 pav. Virtualiosios realybės posistemės panaudojimo atvejų diagrama

Pirmoji diagrama (žr. 3.1 pav.) aprašo virtualiosios realybės posistemės panaudojimo atvejus. Kadangi ši sistema yra glaudžiai susijusi su specialisto darbu, šalia yra atvaizduojama dalis reabilitologo sąsajos panaudojimo atvejų. Čia specialistas turi galėti pradėti sesiją, ją sustabdyti. Jam taip pat yra suteikiama prieiga prie funkcijų, leidžiančių koreguoti paciento poziciją (t. y. jo koordinatas ir aukštį) virtualiosios realybės aplinkoje. Specialistui taip pat yra suteikiama galimybė parinkti ir užkrauti atitinkamus pratimus, kurie reikalauja skirtingų gebėjimų, lavina skirtingas reabilitacijai aktualias sritis. Pabaigiant sesiją, specialistas taip pat gali išsaugoti sesijos duomenis, kurie galės būti panaudoti tolimesniems modelių mokymams.

Pacientas su šia posisteme sąveikauja veikdamas virtualiosios realybės aplinkoje. Specialistui parinkus atitinkamą pratimą, jis turi galėti juos atlikti. Šioje posistemėje egzistuoja keturi pagrindiniai pratimai (žr. 3.2 pav.), tačiau šis skaičius gali keistis.



3.2 pav. Virtualūs pratimai: „Simon Says“ (a), atminties žaidimas (b), monetų skaičiavimas (c), žiedo perkėlimas (d)

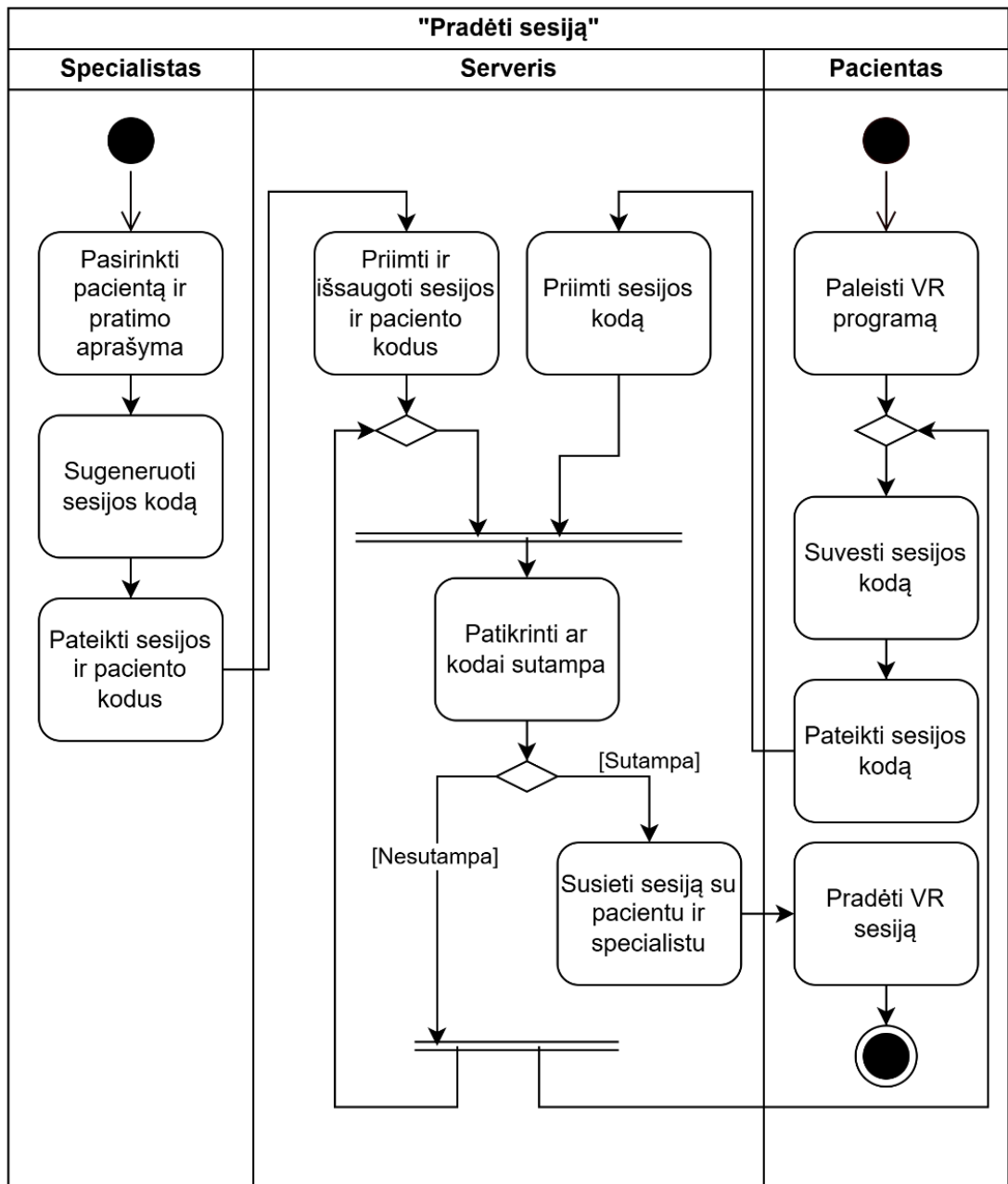
„Simon“ as sako“ (*angl. „Simon Says“*) pratimas reikalauja tiek rankų koordinacijos, tiek atminties. Šis pratimas yra paremtas to paties pavadinimo žaidimu, kuris yra plačiai naudojamas panašios paskirties lavinimo sistemose. Šio žaidimo metu, pacientas mato keturis skirtingų spalvų mygtukus, kurie užsidega tam tikra seka. Pratimo pradžioje yra pažymimas vienos spalvos mygtukas, kurį pacientas turi įsidėmėti ir jį paspausti. Šį žingsnį įvykdžius sėkmingai, prie sekos yra pridemas dar vienas mygtukas, o seka yra pakartojama dar kartą. Pacientas dabar turi įsidėmėti du elementus ir juos teisingai paspausti. Žaidimas tęsiasi tol, kol pacientas suveda neteisingą seką, o laikas, skirtas įsidėmėti seką ir naują elementą jos gale kiekvienos iteracijos metu vis trumpėja.

Atminties žaidimas, dar dažniau žinomas kaip „Matching Pairs“, „Match Up“ arba „Concentration“ yra dar vienas pratimas, kuris yra paremtas realiu atminties lavinimo uždaviniu. Čia pacientui yra

atvaizduojama aibė užverstų kortų. Jis vienu metu gali atversti tik dvi korteles. Jei kortelės sutampa, jos yra pašalinamos. Jei kortelės nėra vienodos, jos vėl yra užverčiamos. Šį atvertimo procesą pacientas kartoja tol, kol nebelieka užverstų kortelių. Pašalinus visas korteles yra sukuriama nauja kortelių aibė, kurios dydis tampa didesnis nei prieš tai.

Monetų skaičiavimo pratimas reikalauja išlavintos smulkiosios motorikos bei gerų kognityvinių gebėjimų, simuliuoja realią parduotuvės situaciją. Šio pratimo metu pacientui yra atsitiktinai „išpilama“ gausybė monetų, eurų (1ct., 2ct., 5ct., 10ct., 20ct., 50ct., 1€ ir 2€). Jo tikslas – surinkti tokias monetas, kurių suma atitiktų tam tikrą nurodytą sumą. Surinkus teisingą sumą, monetas yra iš naujo paskirstomos žaidimo aplinkoje, kas skatina kognityvinių gebėjimų vystymą ieškant reikiamos denominacijos monetas.

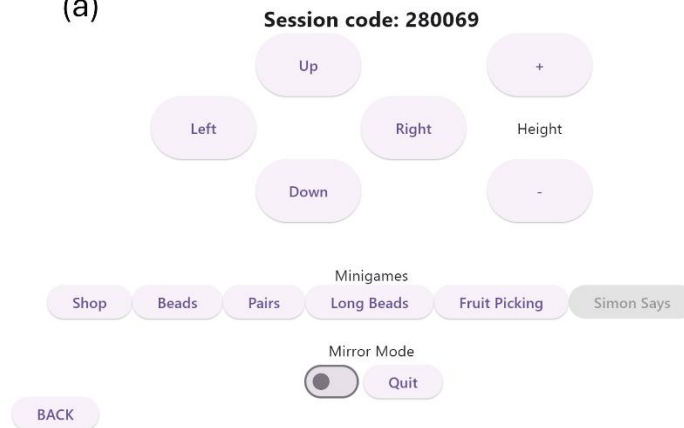
Paskutinis pratimas, žiedo perkėlimas, irgi atitinka realų pratimą. Šio pratimo metu, pacientui yra pateikiamas tam tikras kompleksinis kelias, strypas. Viename jo gale yra žiedas. Pratimo tikslas yra tą žiedą perkelti į priešingą pusę. Šis žiedas yra prirakintas prie strypelio, tad pacientas turi atsargiai ranka sekti virtualų kelią. Virtualiosios realybės žiedo reprezentacijai nutolus per toli nuo paciento rankos, žiedas yra paleidžiamas. Žiedas simuliuoja gravitaciją ir nukrenta atitinkamai ten, kur yra strypelio lokalus minimumas. Jei pacientas prie pratimo užtrunka per ilgai, jam yra leidžiama paspausti mygtuką, kuris įkelia naują žiedo ir strypelio konfigūraciją.



3.3 pav. "Pradėti sesiją" panaudojimo atvejo veiksmų diagrama

Vienas sudėtingesnių panaudojimo atvejų, kuriame dalyvauja net trys aktoriai bei kuriam reikėjo apgalvoti teisingą funkcionalumą buvo „Pradėti sesiją“. Šis panaudojimo atvejis buvo detalizuotas veiksmų diagramoje (žr. 3.3 pav.). Kadangi duomenų suvedimas virtualiosios realybės aplinkoje yra sudėtingas ir nepatogus, o pacientai gali turėti judėjimo sutrikimų, buvo pasirinkta nenaudoti standartinio prisijungimo. Vietoje to, ši sistema pasitelks vienkartinio kodo principą. Prieš pradėdant pratimus, specialistas turi užregistruoti sesijos pradžią – jis pasirenka pacientą, su kuriuo dirbs, bei pratimų aprašymą. Tai atlikus, „Flutter“ programėlė sugeneruoja atsitiktinį šešių skaitmenų kodą ir jį išsiunčia į serverį, kuris šį kodą saugos ir lauks žinutės iš virtualiosios realybės programos. Šį procesą galima matyti 3.4 pav..

(a)

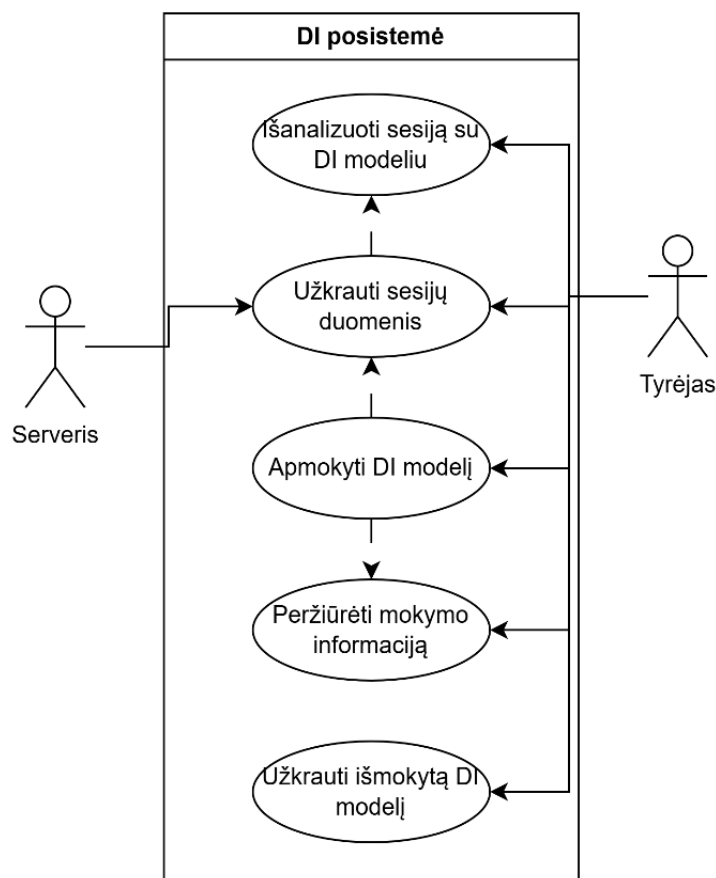


(b)



3.4 pav. „pradėti sesiją“ vaizdai iš rehabilitologo perspektyvos (a) ir paciento perspektyvos (b)

Pacientas, kuris turėtų būti toje pačioje patalpoje kaip ir specialistas, žinos šį sugeneruotą kodą – jį jam gali arba parodyti, arba padiktuoti specialistas. Šį kodą jis turi suvesti virtualiosios realybės posistemėje ir išsiųsti į serverį. Serveryje kodai yra sulyginami tarpusavyje ir jei buvo rasta atitiktis, pacientui yra leidžiama toliau atlikti pratimus, o specialistas įgauna prieigą prie pratimų parinkimo ir paciento pozicijos virtualiojoje aplinkoje koregavimo.



3.5 pav. Dirbtinio intelekto posistemės panaudojimo atvejų diagrama

Dirbtinio intelekto posistemėi taip pat buvo sukurta panaudojimo atvejų diagrama (žr. 3.5 pav.). Čia posistemė turi gebėti atlikti standartines dirbtiniam intelektui aktualias funkcijas. Tyrėjas, kuris turės prieigą prie šios posistemės, turi gebėti užkrauti mokymo duomenis iš serverio, mokyti naują modelį bei peržiūrėti jo mokymo istoriją. Turint sėkmingai išmokytą modelį, ji reikia gebėti užkrauti į atmintį bei jam pateikti sesijos duomenis tolimesniam apdorojimui.

3.3. Techninių reikalavimų specifikacija

Siekiant užtikrinti tinkamą sistemos veikimą, buvo išskirti pagrindiniai techniniai reikalavimai. Jie padeda apibrėžti tikslus aparatinės ir programinės įrangos apribojimus, reikalingus stabiliai sistemos veiklai.

1. Rankų judesiai fiksuojami 30 kadrų per sekunde greičiu. Toks dažnis leidžia tiksliai užfiksuoti paciento atliekamus judesius. Mažesnis dažnis gali paslėpti smulkius judesius, tokius kaip rankų tremorą, o tai lemtų prastesnį modelių tikslumą.
2. Rankų judesiai turi būti saugomi CSV formatu. Šis formatas yra plačiai naudojamas įvairiose srityse, o egzistuojančios „Python“ bibliotekos leidžia greitai ir paprastai užkrauti ir apdoroti tokius failus.
3. Sistema turi veikti su „Meta Quest 3“ virtualiosios realybės akiniais. Šis modelis šiuo metu yra naujausias „Quest“ serijos įrenginys ir vienas populiariausių virtualiosios realybės modelių. Papildomai, šis įrenginys palaiko tikslų rankų sekimą be išorinių valdiklių ar jutiklių

naudojimo, o tai leidžia įgyvendinti smulkiosios motorikos pratimus bei rinkti plaštakos sąnarių informaciją.

4. Sistema turi palaikyti realaus laiko sutrikimų sekimą. Tai leidžia gyvai stebėti atliekamų pratimų kokybę bei imtis pagalbinių priemonių aptikus tam tikrus sutrikimus. Be to, realaus laiko sekimas užtikrina, kad pacientui nereikės laukti modelio įverčių. Šiuos įverčius taip pat būtų galima naudoti integruojant į virtualiosios realybės posistemę ir sukuriant sistemą, kuri dinamiškai reguliuoja pratimų sudėtingumą.

Visi šie reikalavimai yra svarbūs siekiant sėkmingai įgyvendinti anksčiau užsibrėžtus tikslus.

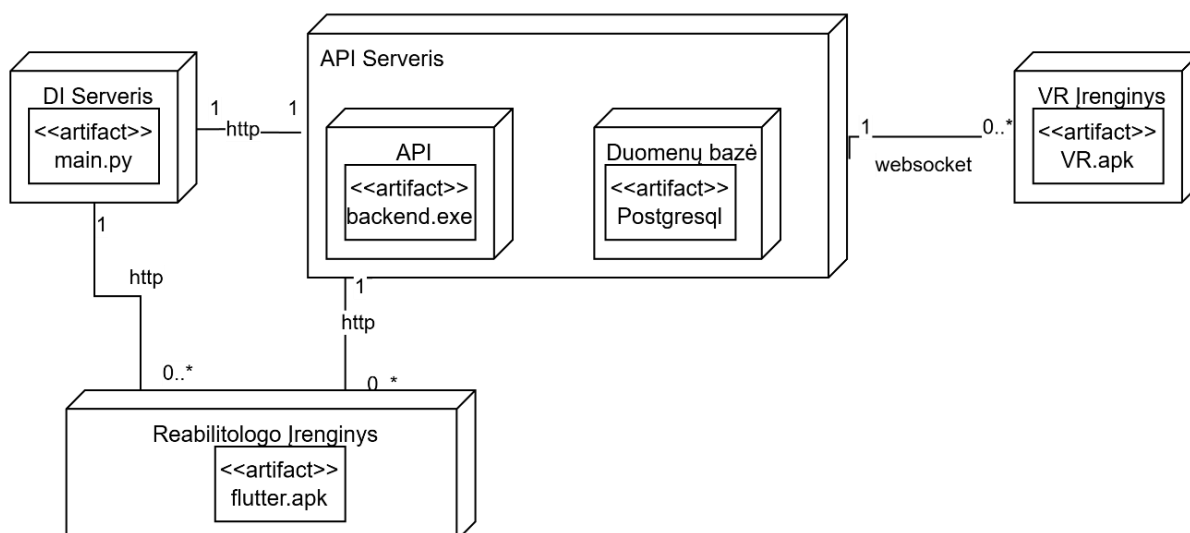
4. VR reabilitacijos sistemos ir DI modelių metodologija

Siekiant užtikrinti tikslingą darbo kryptį tyrimo metu, buvo sudaryta metodologija. Sistemos architektūros skyrelyje yra aprašomi architektūriniai sprendimai, kurie buvo priimti kuriant reabilitacijos sistemą. Čia taip pat yra aprašomi duomenų rinkimo ir apdorojimo niuansai, tokie kaip duomenų disbalansas ir sintetinių įrašų generavimas bei panaudotų dirbtinio intelekto modelių architektūrai aprašymai.

4.1. Sistemos architektūra

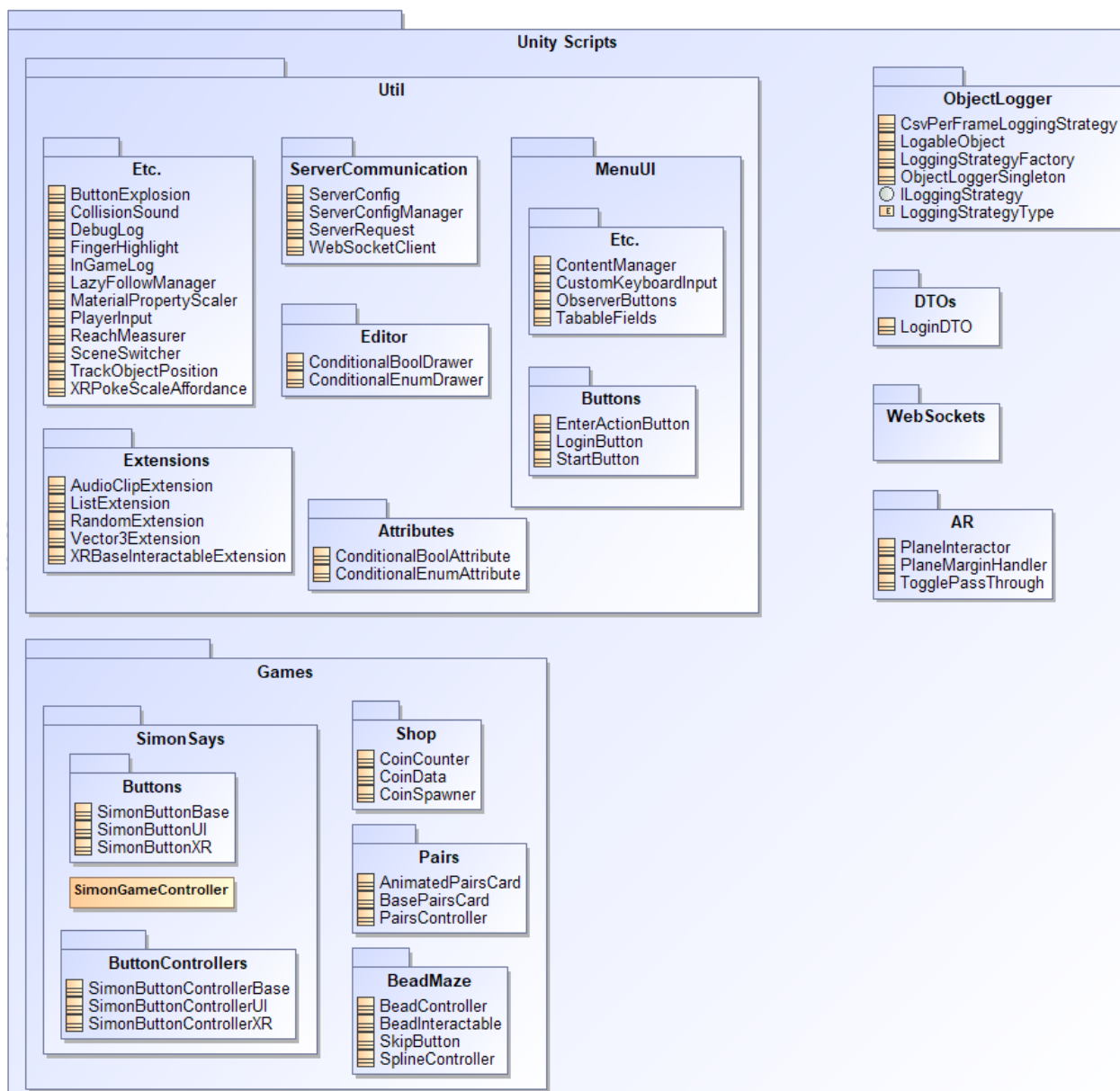
Kuriant šią sistemą, reikia iš anksto numatyti galimą jos architektūrą. Toliau (žr. 4.1 pav.) yra pateikiama diegimo diagrama, kurioje nusakomas galimas jos išsidėstymas serveriuose ir įrenginiuose.

Šioje sistemoje, virtualiosios realybės įrenginys su serveriu palaiko tiesioginį, dvipusį ryšį „SignalR“ karkasu arba „websocket“ protokolu. Tai terapeutui leidžia realiu laiku komunikuoti iš mobilaus įrenginio per serverį su virtualiosios realybės įrenginiu, keisti pratimus, paciento poziciją.



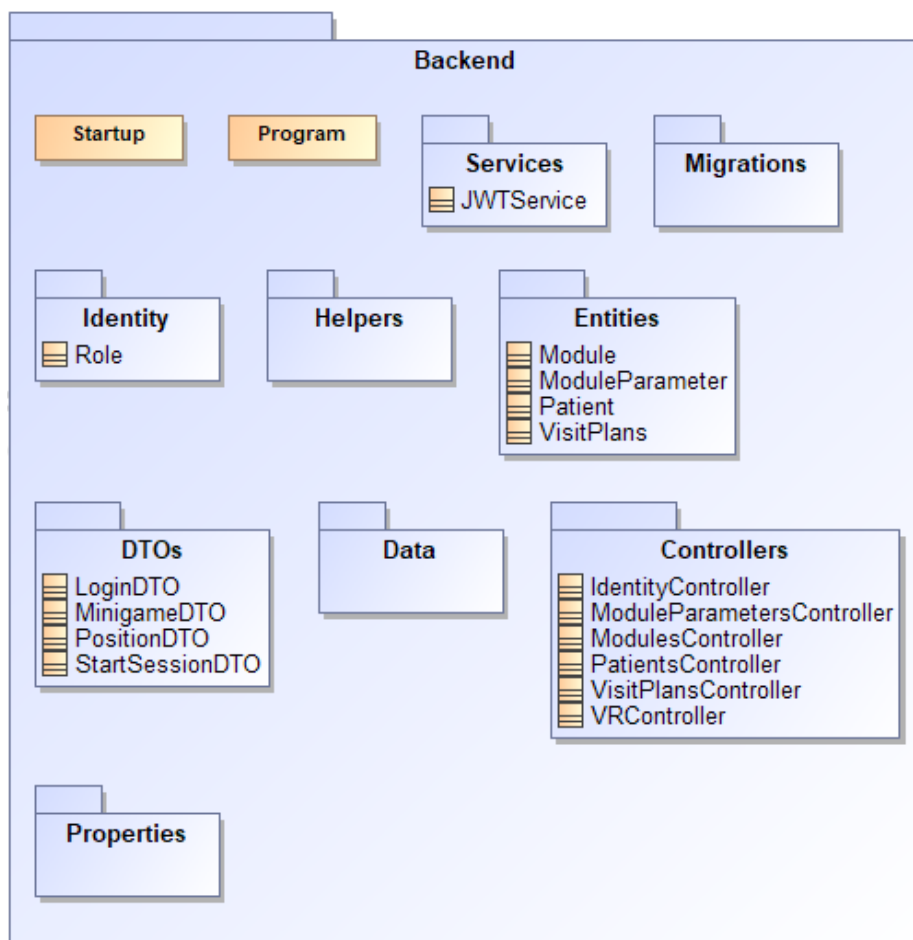
4.1 pav. Diegimo diagrama

Tyrimo metu taip pat buvo paruoštas virtualiosios realybės posistemės prototipas su dalimi pratimų. Konkreti šios posistemės realizacijos architektūra pateikiama paketų diagramoje (žr. 4.2 pav.). Joje galima matyti sistemos apimtį, sukurtus įrankius užtikrinančius posistemės veikimą bei pratimų realizacijas.



4.2 pav. Virtualiosios realybės posistemės paketų diagrama

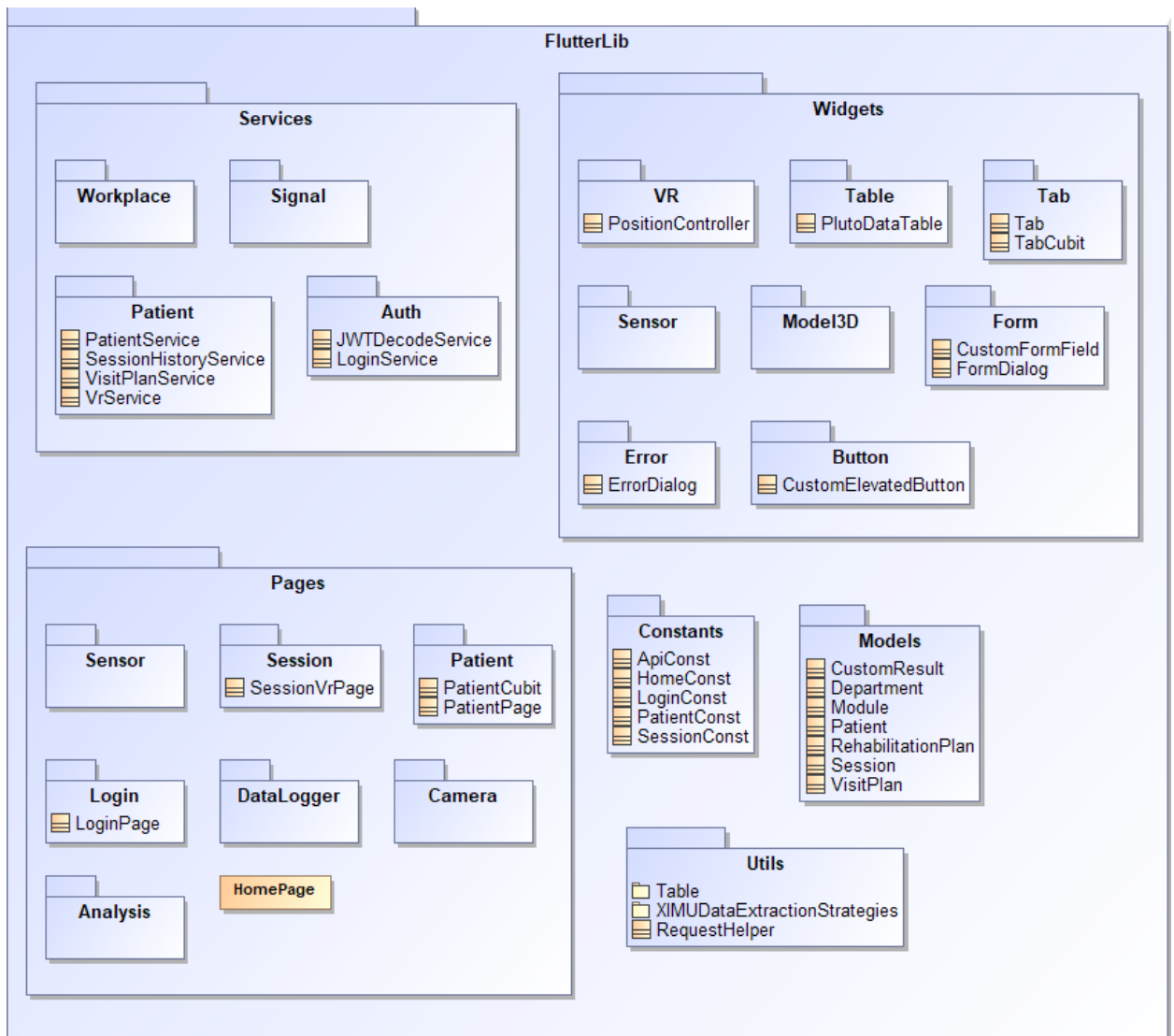
Darbo metu taip pat buvo suprojektuota ir realizuota serverio architektūra. Jos realizacija pateikiama paketų diagramoje (žr. 4.3 pav.). Kadangi serveris buvo kuriamas komandoje komerciniais tikslais, verta pastebėti, kad serverio architektūra yra didesnė nei pateikiama diagramoje – čia paliktos tik smulkiosios motorikos lavinimo sistemai aktualios funkcijos.



4.3 pav. Sistemos serverio paketų diagrama

Siekiant, kad terapeutas galėtų prijungti pacientą prie sistemos bei redaguoti jo poziciją, taip pat reikėjo sukurti naudotojo sąsajos posistemę. Šios paketų diagrama pateikiama toliau (4.4 pav.). Kadangi ši sistemos dalis buvo kuriama komerciniais tikslais, o jos panaudojimo atvejai aprėpia daugiau nei užsibrėžta šio darbo metu, diagramoje pateikiamos tik smulkiosios motorikos lavinimo sistemai aktualios dalys.

Realizuota naudotojo sąsajos posistemė leidžia terapeutui prisijungti prie savo paskyros naudojantis mobiliuoju įrenginiu. Čia jis gali pasirinkti pacientą, atliekamus pratimus. Atliekamų pratimų yra daugiau nei jų apibrėžiama virtualiosios realybės posistemėje. Jei pasirinktas pratimas yra virtualiosios realybės pratimas, terapeutui yra rodomas virtualiosios realybės valdymo langas su pacientui skirtu vienkartinio prisijungimo kodu.

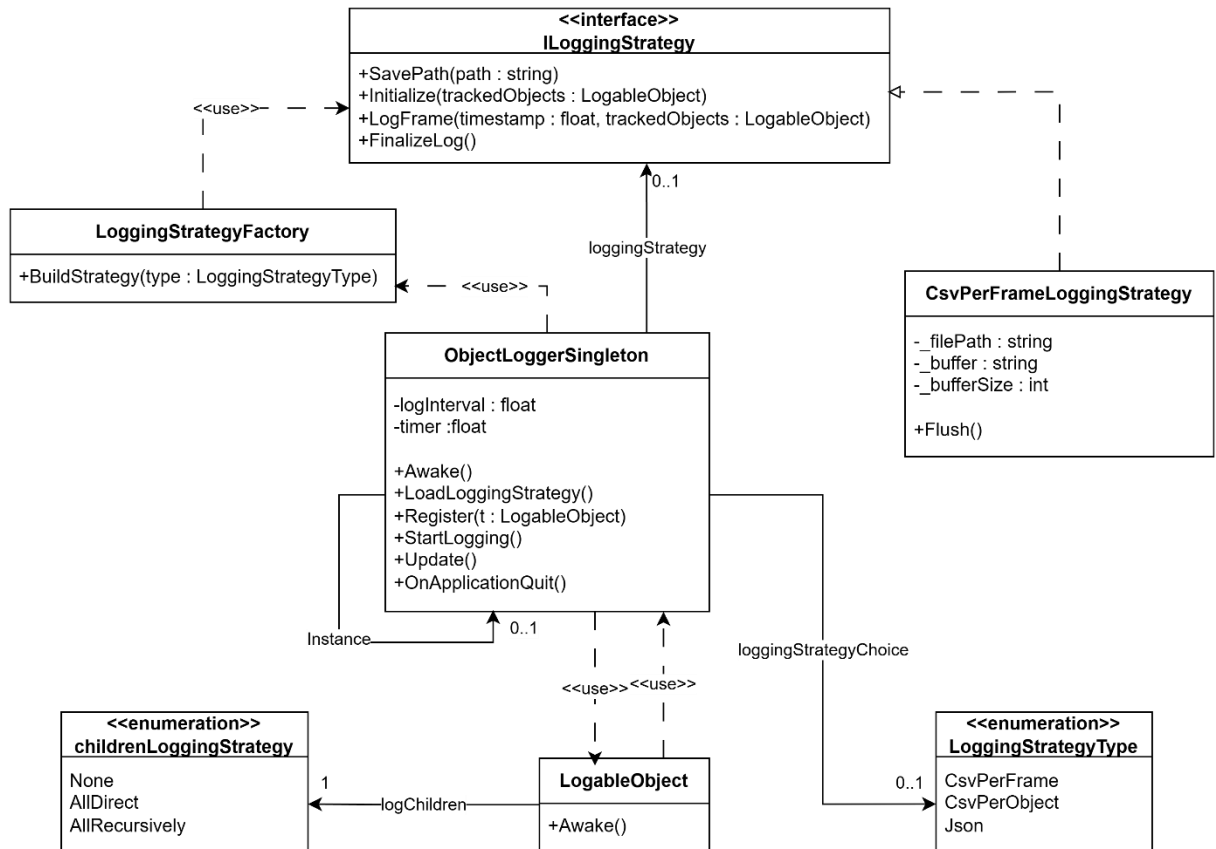


4.4 pav. Naudotojo sąsajos posistemės paketų diagrama

4.2. Duomenų rinkimas ir apdorojimas

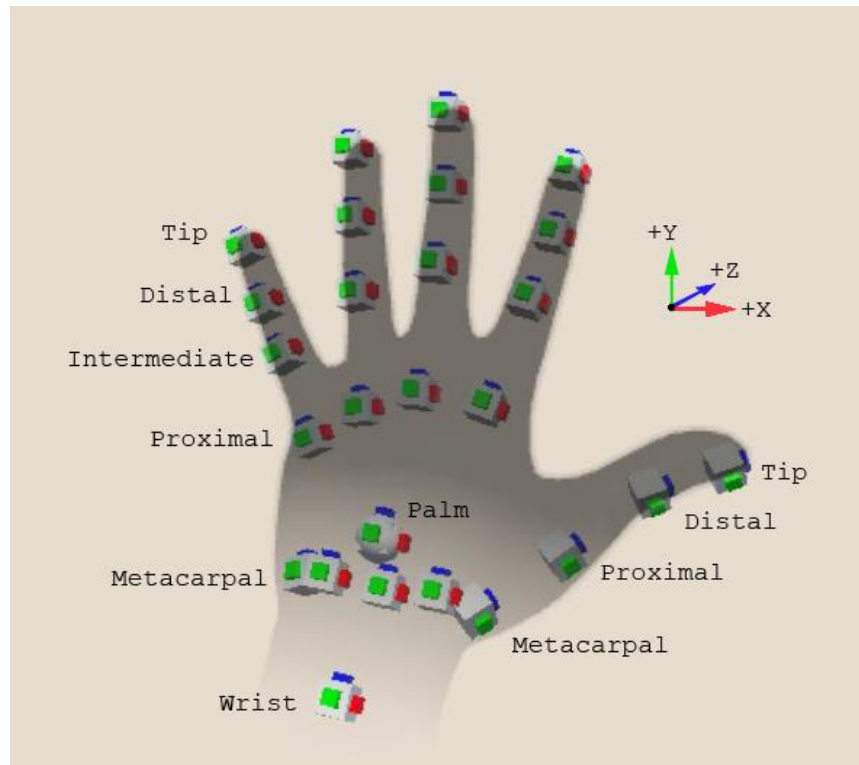
Siekiant surinkti duomenis, kurie bus reikalingi dirbtinio intelekto modeliams, taip pat buvo suprojektuota „Unity“ žaidimo objektų sekimo sistema. Jos klasių diagrama yra pateikiama toliau (4.5 pav.).

Pagrindinis šios sistemos veikimo taškas yra vienetinis „ObjectLoggerSingleton“ objektas. Jis suregistruoja visus „LogableObject“ žaidimo objektus ir jų hierarchijoje esančius vaikus ir kas tam tikrą numatytą laiko intervalą, kuris buvo užsibrėžtas keliant sistemos reikalavimus, jų poziciją nusakančius duomenis įrašo į failą. Sistema palaiko CSV duomenų failus. Duomenų įrašymui buvo panaudotas strategijos programavimo šablonas, tad norint duomenis įrašyti į kitokį failo tipą ar pakeisti duomenų formatavimą, mums pakanka realizuoti šios strategijos sąsają.



4.5 pav. Žaidimo objektų sekimo sistema

Sesijos metu surinkti duomenys yra įrašomi į CSV failą, kurio pavyzdys renkant rankų duomenis yra pateikiamas toliau (žr. 4.1 lentelė). Gautas failas turi iš viso 365 stulpelius – vienas stulpelis laiko žymei ir po septynis stulpelius kiekvienam sekamam objektui – trys stulpeliai nusakantys objekto poziciją trimatėje erdvėje ir keturi stulpeliai aprašantys objekto pasukimą. Kadangi virtualiosios realybės aplinkoje rankos yra apibrėžiamos naudojantis 26 sekimo taškais (žr. 4.6 pav.), faile reikia išsaugoti 52-ių objektų pozicijos ir pasukimo duomenis.



4.6 pav. Rankos modelis [20]

4.1 lentelė. Rankų duomenų failo pavyzdys

Timestamp	R_Wrist_Pos X	R_Wrist_Pos Y	... viso 365 stulpeli ai	L_ThumbTip_Ro tZ	L_ThumbTip_Rot W
100.4617	-1.824469	0.7762566	...	-0.5384204	-0.6836388
100.7126	-1.832777	0.7611768	...	-0.6880619	-0.6552775

Prieš renkant duomenis, taip pat yra svarbu užsibrėžti tikslines klases, kurios bus naudojamos modelių mokymui. Čia buvo išskirti keturi pagrindiniai judėjimo sutrikimai – sumažėjęs judesių diapazonas X ašyje (horizontalioje plokštumoje), sumažėjęs judesių diapazonas Y ašyje (vertikaloje plokštumoje), pirštų sustingimas ir drebulys. Šie sutrikimai yra sekami atskirai kiekvienai rankai, todėl siekiant identifikuoti įrašo sutrikimo tipą galima naudoti 8 binarines bitų vėliavėles (žr. 4.2 lentelė).

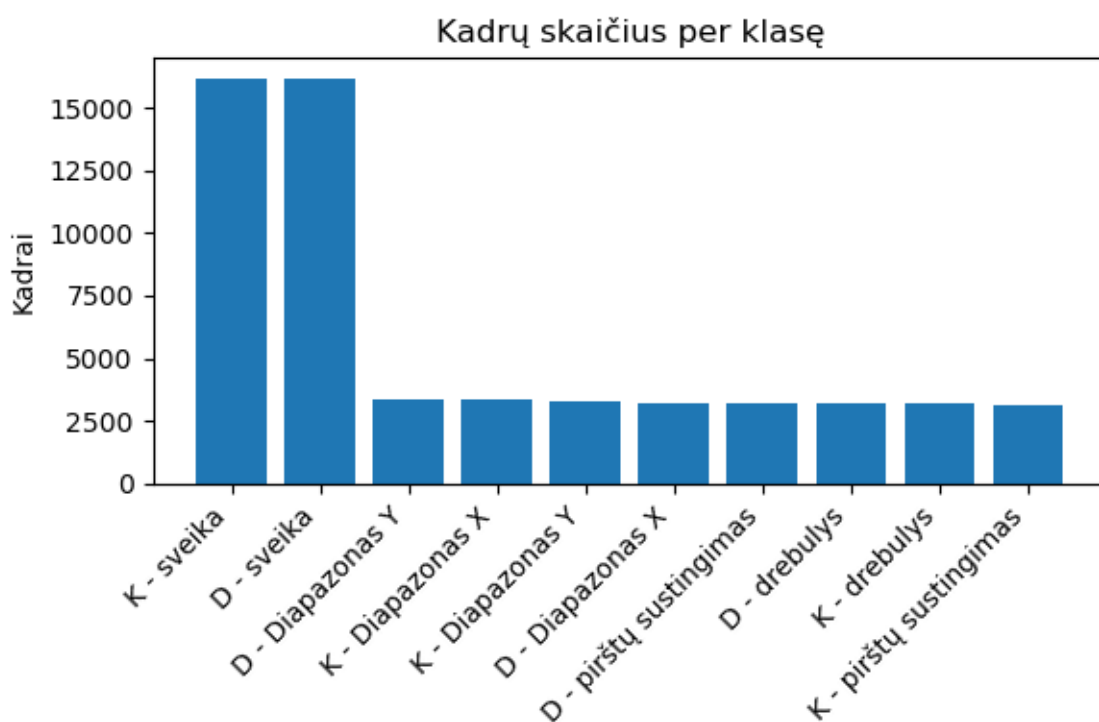
Verta pastebėti, kad toks duomenų rinkinys neatsižvelgia į sutrikimo sunkumą, o tai gali pakenkti modelio tikslumui. Toks ribojimas buvo pasirinktas sąmoningai, kadangi sunkumas, kaip įvertis, yra subjektyvus, o jo įtraukimas apsunkintų teisingą duomenų rinkimą.

4.2 lentelė. Sesijas aprašantys bitai

Bitas	Reikšmė
0	Kairės rankos sumažėjęs judesių diapazonas X ašyje
1	Kairės rankos sumažėjęs judesių diapazonas Y ašyje

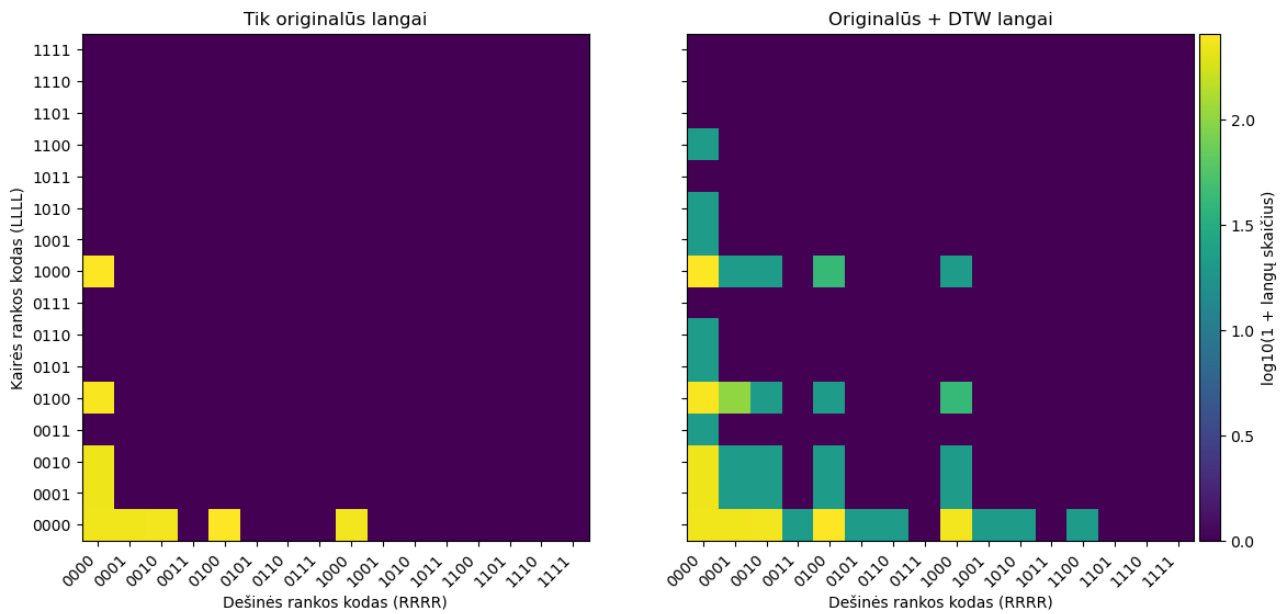
2	Kairės rankos pirštų sustingimas
3	Kairės rankos drebulys
4	Dešinės rankos sumažėjęs judesių diapazonas X ašyje
5	Dešinės rankos sumažėjęs judesių diapazonas Y ašyje
6	Dešinės rankos pirštų sustingimas
7	Dešinės rankos drebulys

Tyrimo metu buvo užregistruotos devynios sesijos. Viena sveika ir aštuonios simuliuojančios po vieną sutrikimą. Kiekviena sesija trunka dvi minutes ir turi apytiksliai 3600 kadrai. Dėl infrastruktūrinių ir etinių sumetimų dirbant su tikrais pacientais, sutrikimai buvo atliekami sveiko asmens sekant ekspertų gaires nusakančias sutrikimų specifika.



4.7 pav. Klasių pasiskirstymas

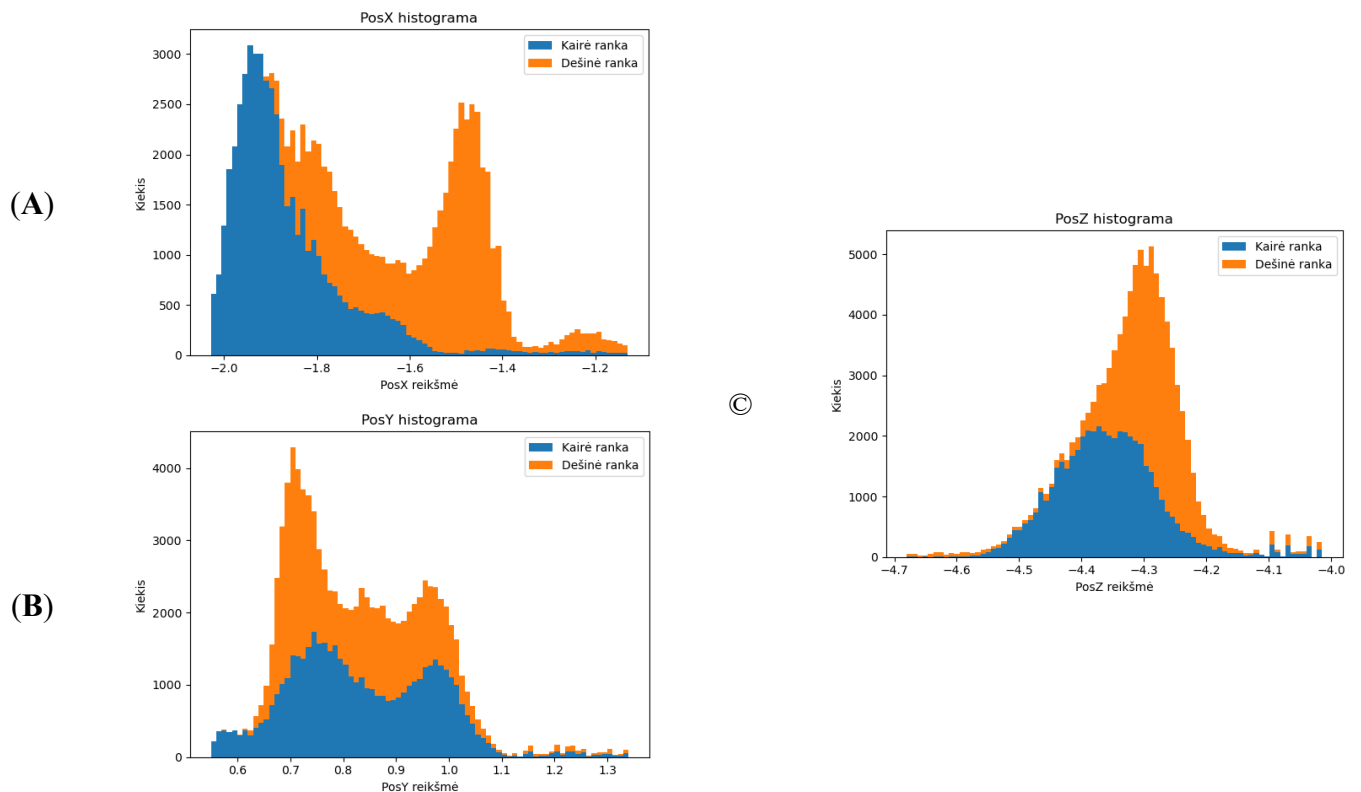
Užfiksuotų klasių pasiskirstymą galima matyti aukščiau (žr. 4.7 pav.). Čia labai aiškiai yra matomas disbalansas tarp sveikų sesijų ir sesijų su sutrikimu. Taip yra dėl to, kad kiekviena sesija izoliuoja konkretų sutrikimą vienoje rankoje. Natūraliai, kita ranka toliau lieka identifikuojama kaip sveika, o tai sudaro duomenų disbalansą į kurį reikia atsižvelgti.



4.8 pav. Mokymo aibės sutrikimo kombinacijų surinktų kadrų kiekis prieš ir po DTW duomenų sintezės

Šis reiškinys labai aiškiai yra matomas paveikslėlyje aukščiau (žr. 4.8 pav.) – čia yra atvaizduojamos visos abiejų rankų sutrikimų kombinacijos. Kadangi kiekviena sesija buvo arba tik sveika, arba simuliuojanti tik vieną sutrikimą, didelė dalis kombinacijų liko nepadengta.

Siekiant šią problemą išspręsti buvo pasitelktas DTW („Dynamic Time Warping“) algoritmas. Jis buvo naudojamas sulygiuoti sesijų sekas, kurios turi skirtingus sutrikimus, atsižvelgiant į trukmės ir tempo skirtumus. Po sulygiavimo, visos sekų kombinacijos buvo interpoliuojamos tarpusavyje, taip sukuriant naujus sintetinius duomenis, turinčius daugiau nei vieną sutrikimą.



4.9 pav. Kairės ir dešinės rankos duomenų pasiskirstymas (A) X ašyje, (B) Y ašyje ir (C) Z ašyje

Prieš pradėdant mokymą, taip pat yra svarbu įsitikinti duomenų kokybe. Trūkstamų ar sugadintų duomenų nebuvo surinkta, o paveikslėlyje aukščiau (žr. 4.9 pav.) yra pateikiamos pozicinių duomenų histogramos skirtingose ašyse. Čia galima matyti, kad visi pratimai buvo atliekami toje pačioje pozicijoje, nėra tam tikrų klasterių kuriuos būtų galima panaudoti klasių atskyrimui. 4.9 pav. (A) galima matyti du kauburius, tačiau tai yra rezultatas, kurio buvo galima tikėtis – kairė ir dešinė rankos X ašyje turi skirtingus judėjimo diapazonus.

Šiems duomenims taip pat buvo taikoma augmentacija. Čia buvo panaudotos atsitiktinės afininės transformacijos, kurių reikšmės yra $[0.9,1.1]$, duomenys papildomai buvo pasukami $\pm 5^\circ$ aplink vertikalią ašį. Čia taip pat buvo pridėdamas mažas Gausinis triukšmas $(0,0.01)$.

Papildomai, duomenų rinkinys buvo apdorotas naudojantis slankiojančio lango principu. Kiekviena originali sesija buvo perskirta į tris dalis – mokymo, validavimo ir testavimo. Tai buvo atliekama, siekiant užtikrinti, kad šių rinkinių nebūtų persidengimo. Toliau iš šių segmentų buvo generuojami 20 ir 25 sekundžių langai. Tas pats buvo atlikta ir su sintetiniais duomenimis, tačiau jie priskirti tik į mokymo aibę, siekiant užtikrinti, kad validavimas ir testavimas atitiktų realybę. Galutinį duomenų rinkinį sudaro 70 % mokymo langų, 15 % validavimo ir 15 % testavimo.

4.3. Dirbtinio intelekto modelių architektūra

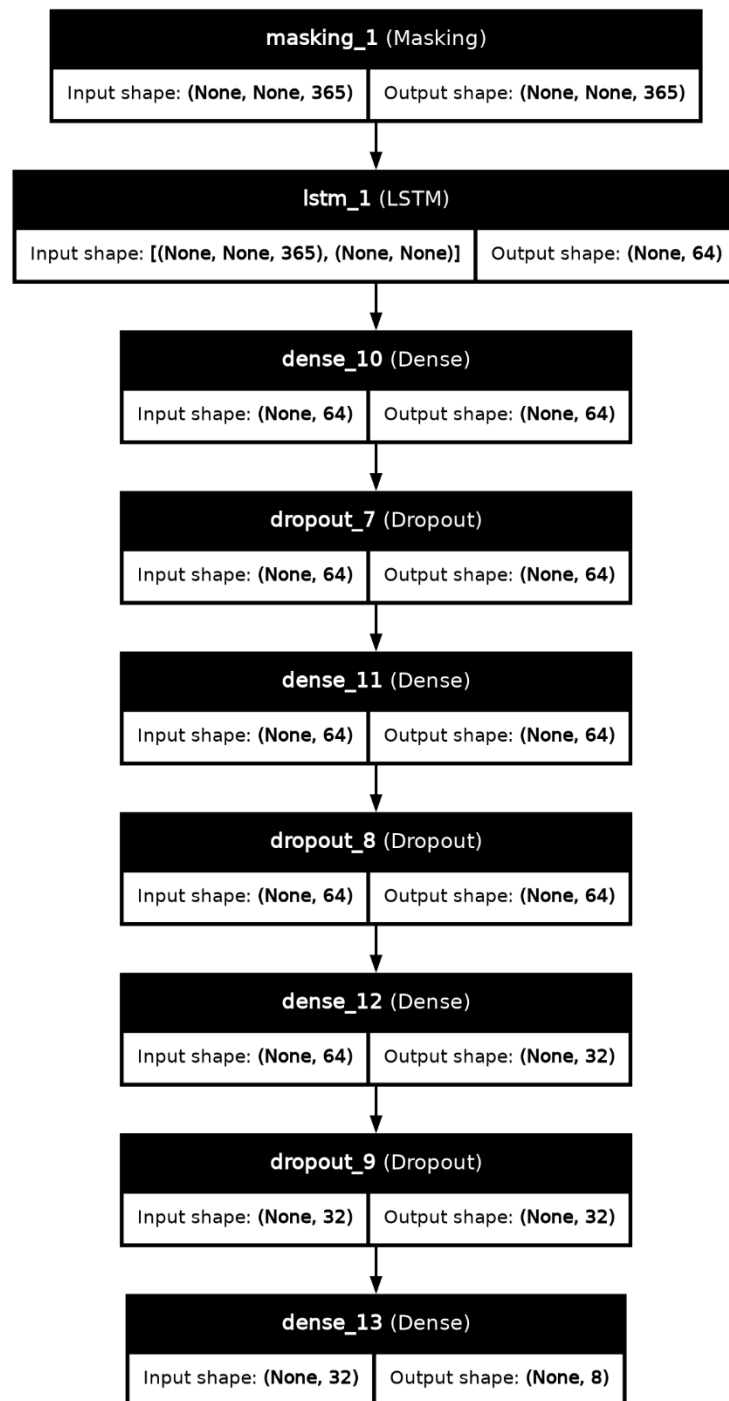
Surinkus duomenis gali būti pradėdamas modelių mokymo etapas. Šiame tyrime buvo dirbama su trimis modelių tipais – rekurentiniai ilgos trumposios atminties architektūra grįsti modeliai, erdviniai-laikiniai grafų konvoliuciniai tinklai (*angl. Spatial Temporal Graph Convolutional Networks*, „ST-GCN“) ir „XGBoost“. Tyrimo metu taip pat buvo sukurtas ansamblinis „LSTM-XGBoost“ modelis.

Čia yra svarbu apsibrėžti išvesties strategiją – yra galimos dvi. Vienu atveju mes galime gražinti vektorius, kuris naudoja „softmax“ funkciją ir nusako labiausiai tikėtiną sutrikimo klasę. Šis atvejis tinka tik tada, kai sesija turi tik vieną izoliuotą sutrikimą, taip kaip ir yra dabar surinkti duomenys. Tačiau šis variantas nėra tinkamas realiame gyvenime.

Geresnis sprendimas yra gražinti vektorius, kuris nusako kiekvieno sutrikimo tikimybę. Jei vertės perkopia tam tikrą reikšmę, mes galime teigti, kad užfiksuotoje sesijoje buvo atpažinti tam tikri sutrikimai. Šis metodas geriau atkartoja situacijas, kurios bus aptinkamos realybėje, kadangi pacientai turės daugiau nei vieną sutrikimą vienu metu. Tačiau tokį modelį yra sunkiau mokyti su tokiais duomenimis, kuriuos turime mes.

4.3.1. Ilgos trumposios atminties modeliai

Sesijos apdorojimo atveju „LSTM“ architektūra yra pranaši vienu veiksniu – ji gali nuosekliai apdoroti kintamo ilgio sekas. Tai yra naudinga dvejais atvejais. Visų pirma, šis modelis mums leidžia realiu laiku, po vieną kadra, apdoroti sesijos informaciją šiai dar nepasibaigus, t.y. kitaip nei medžiais grįstuose modeliuose, mums nereikia išvestinių kintamųjų. Tai padeda taupyti laiką ir skaičiavimo resursus. Visų antra, šis modelis turėtų gebėti atsižvelgti į informaciją, kuri gali įvykti trumpose sekose ir išvestiniuose duomenyse nesimatytų.



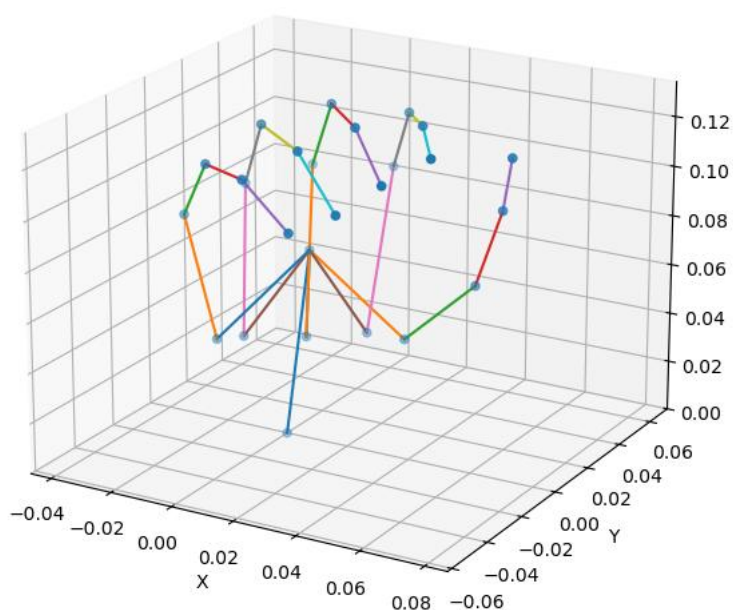
4.10 pav. „LSTM“ architektūra

Anksčiau išgauti langai yra nuosekliai paduodamos į „LSTM“ sluoksnį, kuris jas apdoroja bei apsimoko kuri informacija yra naudinga ir reikia atsiminti, o kurią galima atmesti. Duomenims praėjus šį sluoksnį yra gaunamas užkoduotas išvesties vektorius. Šis vektorius toliau yra paduodamas į neuroninį tinklą, kur, atitinkamai nuo užduoties specifikacijos, yra išgaunamas arba labiausiai tikėtinos klasės vektorius, kai bandoma klasifikuoti tik vieną sutrikimą, arba vėliavėlių tikimybių vektorius, kai įvertinti bandoma visus sutrikimus vienu metu. Šio modelio architektūra yra atvaizduota aukščiau (žr. 4.10 pav.)

4.3.2. Erdviniai-laikiniai grafų konvoliuciniai tinklai

Erdviniai-laikiniai grafų konvoliuciniai tinklai taip pat savo viduje naudoja rekurentinius tinklus siekiant apdoroti ilgas sekas. Bet tai, kas padaro juos specialius yra grafų sąvoka – ji duomenims suteikia struktūrą ir leidžia atlikti panašius konvoliucinius veiksmus, kurie būtų atliekami su dvimačiais taškiniais vaizdais. Vietoje to, kad būtų naudojamas slankiojantis langas, kuris atlieka veiksmus su kaimyniniais pikseliais, grafų sąvoka leidžia paimti grafo viršūnes, kaulus, kurių informacija yra saugoma įvesties duomenyse, ir atlikti veiksmus, t.y. išvesti vidurkį, su viršūnės kaimynais (žr. 4.11 pav.) ir kitais kadrtais. Šis metodas modeliui gali suteikti realią informaciją apie kaulų struktūrą, taip galimai palengvinant sprendimų priėmimą.

L hand — sample 0, frame t=50



4.11 pav. Kairės plaštakos grafo struktūra

Šie konvoliuciniu būdu apdoroti kadratai tada gali būti paduodami į rekurentinį sluoksnį, kuris apibendrina informaciją į vieną vektorius. Šis vektorius, lygiai taip pat, kaip ir „LSTM“ tinkle, yra apdorojamas neuroniniuose sluoksniuose ir iš jų gaunamas išvesties vektorius, kurio tiksli strategija gali keistis nuo panaudojimo būdo. Siekiant užtikrinti, kad modelių lyginimas būtų lygiavertis, rekurentiniai ir klasifikavimo sluoksniai yra identiški „LSTM“ modeliui (žr. 4.10).

4.3.3. „XGBoost“

„XGBoost“, tai mašininio mokymosi sistema pagrįsta sprendimų medžių auginimo (*angl. boosting*) algoritmu. Šio metodo esmė yra iteratyvus miško konstravimas, kai sukuriant kiekvieną naują medį yra mokomasi kompensuoti anksčiau padarytas klaidas. Tokiu būdu modelis gerina savo tikslumą ir geriau orientuojasi į išimtinius atvejus. „XGBoost“ nuo kitų klasikinių sprendimų išsiskiria dar tuo, kad šioje sistemoje yra realizuojami įvairūs optimizavimo sprendimai, kurie pagerina modelio tikslumą ir padeda išvengti persimokymo.

Šiame tyrime „XGBoost“ buvo naudojamas kaip alternatyvus sprendimas rekurentiniams neuroniniams tinklams. Kadangi šis modelis negali tiesiogiai apdoroti kadrų sekos, pradiniai kaulų duomenys buvo transformuoti į fiksuoto ilgio požymių vektorius. Čia kiekvienam kaulo kintamajam buvo apskaičiuotos agreguotos statistinės charakteristikos: vidurkis, standartinis nuokrypis, minimalios ir maksimalios reikšmės, mediana, pirmas ir trečias kvartilis, absoliutaus pokyčio vidurkis ir RMS.

Verta pastebėti, kad tuo atveju, kai ieškoma sutrikimų aibė, o ne vienas labiausiai tikėtinas sutrikimas, kiekvienam sutrikimui yra sukuriamas atskiras modelis, o tai skaičiavimo laiką sparčiai išaugina net aštuonis kartus.

4.3.4. Ansamblis

„LSTM“ ir „XGBoost“ modeliai turi atskirus privalumus. Pirmasis – tai rekurentinis tinklas, o tai reiškia, kad mums nereikia turėti visos sesijos informacijos siekiant pradėti ją apdoroti. Mes galime jau nuo sesijos pradžios pradėti nuosekliai krauti kadrus į modelį ir juos perleisti per „LSTM“ sluoksnį šių dar nepaduodant klasifikatoriaus sluoksniui. To pasėkoje, mes turime modelį, kuris yra pakankamai greitas veikti realiu laiku. Kita vertus, nors „XGBoost“ yra lėtesnis, kadangi jis turi visus duomenis paleisti per gausią aibę medžių, jis pasižymi aukštesniu tikslumu ir geba geriau atskirti sutrikimus.

Siekiant sukombinuoti šiuos du modelius, buvo sukurtas taisyklėmis grįstas ansamblinis modelis sutrikimų aibei klasifikuoti. Pradinė kiekvienos sesijos sutrikimų diagnozė iš pradžių yra gaunama iš „LSTM“ modelio, o toliau, atsižvelgiant į sudarytą taisyklių rinkinį, kiekvienos diagnozės prognozė gali būti modifikuojama „XGBoost“ modelio.

Pačios taisyklės buvo sudaromos kiekvienam sutrikimui atskirai, kadangi atpažinimo kokybė tarp skirtingų sutrikimų nėra pastovi. Čia buvo naudojamos tokios taisyklės, kaip „visada naudoti „LSTM“ rezultatus“, „naudoti „XGBoost“, jei „LSTM“ mažas užtikrintumas“, „naudoti „XGBoost“ jei „LSTM“ spėjimas yra lygus 0“ ir taip toliau. Taisyklės buvo išrenkamos naudojantis validavimo aibėje gautu F_1 įverčio pagerėjimu, tokiu būdu neliečiant testavimo aibės ir sprendimus paliekant nešališkus.

Toks ansamblinis modelis leidžia naudoti „LSTM“ kaip pagrindinį realaus laiko klasifikatorių, o „XGBoost“ – tik kaip papildomą korekcijos mechanizmą.

5. Viršutinės kūno dalies judėjimo sutrikimų klasifikavimo rezultatai

Šiame skyriuje bus pateikiami eksperimentai, kurių metu buvo bandoma surasti minimalią įmanomą kintamųjų aibę. Tai buvo atliekama dvejais būdais – godžiai atmetant kaulų poras ir ieškant tokios kaulų kombinacijos, kuri pasirodytų geriausiai, bei atmetant pozicinius arba pasisukimo įverčius.

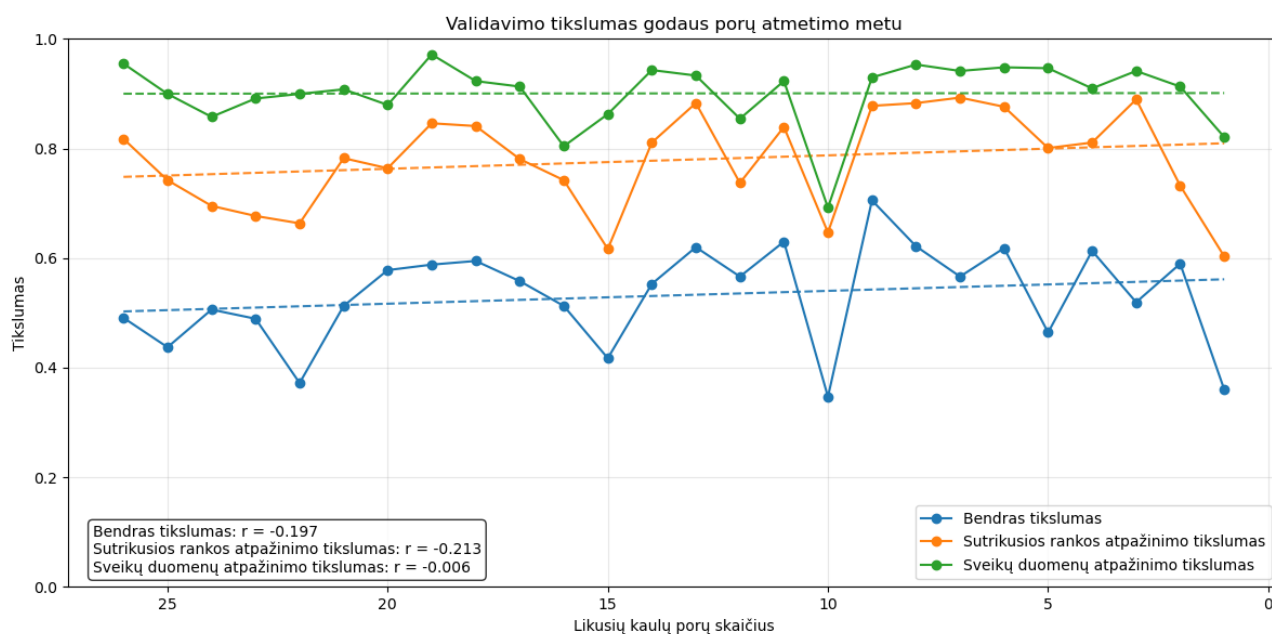
Čia taip pat buvo įvertinti trijų modelių, „LSTM“, „ST-GCN“ ir „XGBoost“, rezultatai dvejomis išvesties strategijomis – kai yra spėjama viena labiausiai tikėtina klasė ir kai yra spėjamos klasių vėliavėlių tikimybės.

5.1. Minimali kintamųjų aibė

Siekiant surasti minimalią kintamųjų aibę, buvo atlikti eksperimentai godžiai atmetant kaulus bei iš mokymo aibės atmetant pozicinius bei pasisukimo įverčius. Šių eksperimentų rezultatas leidžia atrinkti pačius naudingiausius kintamuosius, kuriuos reikia saugoti norint išlaikyti tikslumą bei optimizuoti duomenų rinkinį.

5.1.1. Godus kaulų atmetimas

Godaus kaulų atmetimo eksperimentas buvo vykdomas etapais. Kiekvienas etapas prasideda mokinant standartinį „LSTM“ modelį naudojantis visais pasiekiamais duomenimis. Šis gautas modelis tada yra naudojamas surasti tą kaulų porą, t.y. kairės ir dešinės plaštakos kaulus, kurių reikšmės sugadinus ir pavertus nuliais, modelis suprastėja mažiausiai. Šis kaulas tada yra pašalinamas iš duomenų rinkinio, pradedamas naujas etapas, o modelis mokinamas jau su modifikuotais duomenimis. Tai yra kartojama tol, kol lieka tik vienas kaulas.



5.1 pav. Validavimo tikslumas godaus porų atmetimo metu

Šio eksperimento rezultatus galima matyti 5.1 paveikslėlyje. Čia matome, kad koreliacija tarp modelio tikslumo ir kaulų skaičiaus yra maža ir neigiama ($r = -0,197$) – tai reiškia, kad modelis veikia sąlyginai geriau, kai jam yra pateikiama mažiau informacijos. Todėl galima daryti išvadą, kad didžioji duomenų dalis yra perteklinė turimiems sutrikimams nuspėti. Rankos diapazoną bei drebulį

įvertinti galima vos su vienu sekimo tašku, o sustingimui pakanka vos kelių taškų, dėl ko yra matomas ženklus modelio tikslumo kritimas likus mažiau nei trims kaulams.

5.1 lentelė. Svarbiausi „LSTM“ kaulai

Svarba	Kaulas
1	Nykščio delnakaulis
2	Didžiojo piršto vidurinis pirštakaulis
3	Delnas
4	Mažylio vidurinis pirštakaulis
5	Smiliaus galiukas
6	Bevardžio vidurinis pirštakaulis

Taip pat verta pastebėti, kad šeši patys svarbiausi kaulai (žr. 5.1 lentelė) priklauso delnui ir kiekvienam atskiram pirštui. Lygiai tokį patį rezultatą galima matyti pažvelgus į „XGBoost“ modelio rezultatus, kai jis buvo išmokytas naudojantis visais pasiekiamais kintamaisiais (žr. 5.2 lentelė) – pirmuose aštuoniuose svarbiausiuose kintamuosiuose mes matome visus dešinės plaštakos pirštus – vidurinį, nykštį, mažylį, bevardį ir smilių.

5.2 lentelė. „XGBoost“ pirmi dešimt svarbiausių kintamųjų

Svarba	Kintamasis
0,047015	Dešinio riešo RotZ absoliutaus skirtumo vidurkis
0,046124	Dešinio vidurinio piršto galiuko RotY pirmasis kvartilis
0,042043	Dešinio delno RotY absoliutaus skirtumo vidurkis
0,037616	Dešinio nykščio galiuko RotY trečiasis kvartilis
0,033326	Dešinio vidurinio piršto delnakaulario RotX pirmasis kvartilis
0,032092	Dešinio mažylio delnakaulario RotY absoliutaus skirtumo vidurkis
0,030630	Kairio smiliaus delnakaulario PosZ pirmasis kvartilis
0,030263	Dešinio bevardžio piršto distalinio pirštakaulario RotZ absoliutaus skirtumo vidurkis
0,030170	Dešinio vidurinio piršto tarpinio pirštakaulario RotZ pirmasis kvartilis
0,030102	Dešinio mažylio delnakaulario RotX mediana

Toks rezultatas sufleruoja, kad standartinį 26 kaulų modelį turėtų būti galima supaprastinti į daugiau nei keturis kartus paprastesnį modelį neprarandant tikslumo ir išlaikant galimybę atpažinti subtilesnius rankos judesius ir gestus.

5.1.2. Pozicinių ir pasisukimo reikšmių atmetimas

Naudojamą duomenų rinkinį sudaro 26 kaulai, kiekvienas su 7 kintamaisiais – trys nusakantys poziciją ir keturi pasisukimą. Atmetant vieną ar kitą reikšmę, duomenų rinkinį galima sumažinti per pusę. Dėl to buvo atliktas eksperimentas, mokinantis „LSTM“ ir „XGBoost“ modelius atpažinti sutrikimų klases tik su vienu iš šių reikšmių.

5.3 lentelė. Modelių tikslumas atmetant pozicijos ir pasisukimo imtis

Metodas	Atmesta imtis	Tikslumas	Pokytis, %
LSTM	Niekas	0,5135	0,00
	Pozicija	0,4969	-3,23
	Pasisukimas	0,2484	-51,63
XGBoost	Niekas	0,6244	0,00
	Pozicija	0,7120	14,03
	Pasisukimas	0,3419	-45,24

Šio eksperimento rezultatai yra pateikiami 5.3 lentelėje. Čia abejais atvejais matome, kad atmetus pasisukimo reikšmes modelio tikslumas sumažėja beveik per pusę. Tai reiškia, kad šis parametras yra ypatingai svarbus identifikuojant sutrikimus.

Pavyzdžiui, sužnybimo judesys, kuris yra naudojamas žiedo perkėlimui valdyti bei sustingimui identifikuoti, yra žymiai lengviau aptinkamas naudojantis pasisukimo, o ne poziciniais duomenimis. Taip yra todėl, nes šis judesys yra lengviau apibrėžiamas pirštų tarpusavio orientacija, o ne absoliučia jų padėtimi erdvėje. Panašiai, judesio diapazono sutrikimus taip pat yra lengviau nustatyti naudojant kampus ir neatsižvelgiant į paciento rankos ilgį ar jo globalią poziciją aplinkoje.

Taip pat yra verta pastebėti, kad atmetus pozicinius parametrus, „XGBoost“ tikslumas pagerėja net 14,03 %. Tai rodo, kad poziciniai duomenys šio modelio atveju yra pertekliniai ir įneša triukšmą, kuris trukdo įsisavinti svarbiausias savybes.

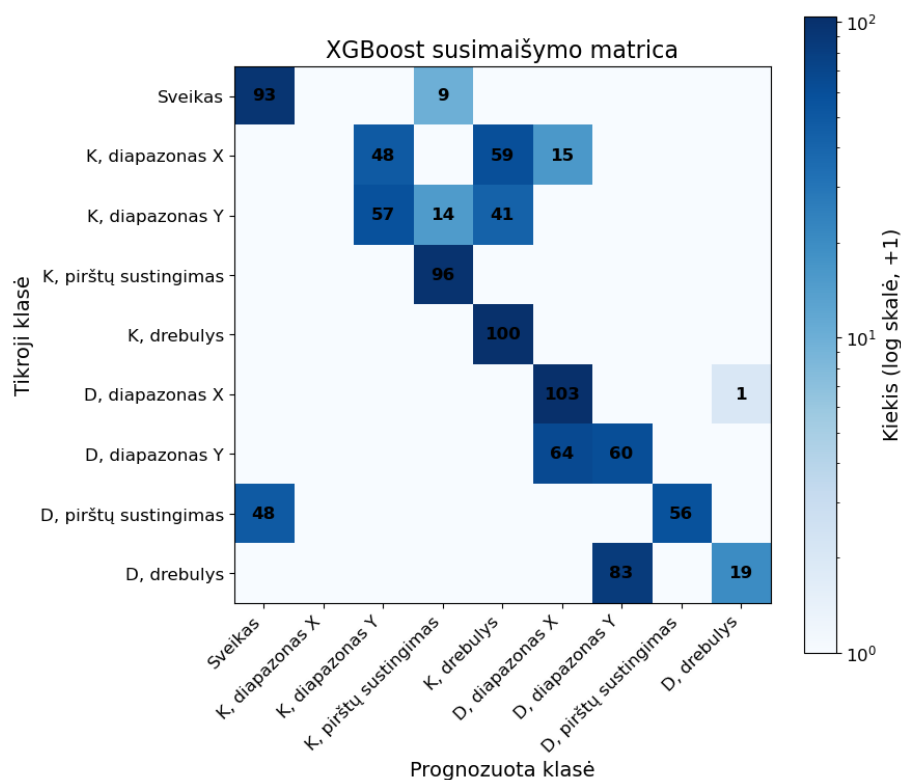
5.2. Labiausiai tikėtino sutrikimo identifikavimas

Siekiant identifikuoti labiausiai tikėtiną sutrikimą buvo naudojami „ST-GCN“, „ST-GIN“, „LSTM“ ir „XGBoost“ modeliai. Siekiant sukurti atskaitos tašką ir įvertinti modelių kokybę, papildomai buvo apskaičiuotas modelis, kuris atsitiktinai spėja sutrikimus, o jo įvertis yra vidutinis rezultatas po šimto epochų. Šio eksperimento rezultatai apibendrinami 5.4 lentelėje.

5.4 lentelė. Modelių rezultatai, identifikuojant labiausiai tikėtiną sutrikimą

Modelis	Diagnozės tikslumas	Rankos atskyrimo tikslumas	Sveiko asmens atpažinimo tikslumas	Macro F_1
Atsitiktinis	0,1107	0,4096	0,8068	0,1103
ST-GCN	0,3157	0,5663	0,8282	0,3157
ST-GIN	0,4224	0,8509	0,8592	0,4224
LSTM	0,5373	0,7412	0,8851	0,4940
XGBoost	0,6046	0,9255	0,9410	0,5591

Rezultatuose galima matyti, kad visi modeliai geba apsimokyti ir dalinai identifikuoti sutrikimus. Geriausiai šią užduotį atlieka „XGBoost“ (žr. 5.2 pav.), o po jo – paprastas „LSTM“ (žr. 5.4 pav.). Prasčiausiai čia pasirodo grafų modeliai („ST-GCN“, „ST-GIN“), o tai reiškia, kad plaštakos modelio struktūros formalizavimas modelyje papildomos vertės neduoda, o kaimyninių reikšmių agregavimas paslepia esminius kintamuosius, kurie yra naudingi modelių mokymui.



5.2 pav. „XGBoost“ susimaišymo matrica

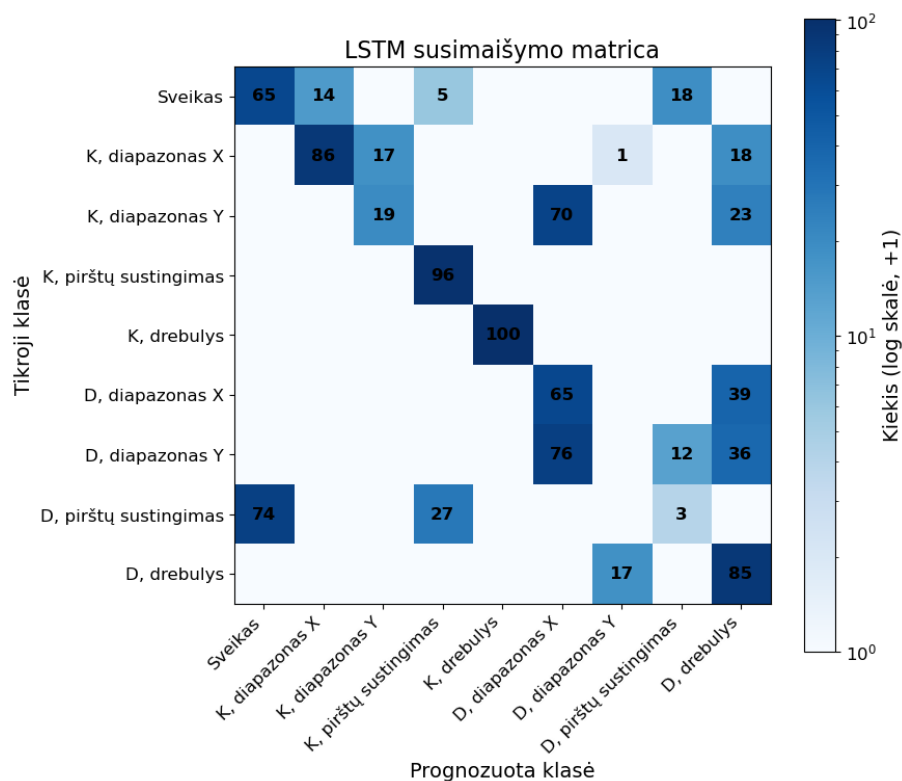
„XGBoost“ modelis pasiekė aukščiausią tikslumą kuris yra lygus 60,46 %. Jis puikiai atpažįsta kairės rankos sustingimą ($F_1 = 0,89$), dešinės rankos diapazono sutrikimą X ašyje ($F_1 = 0,72$) ir sveikus asmenis ($F_1 = 0,77$). Tai rodo, kad modelis geba efektyviai atskirti didelę dalį sutrikimų bei atpažinti sveikus įrašus.

Kita vertus, kairės rankos diapazono sutrikimas X ašyje yra visiškai nesuklasifikuojamas ($F_1 = 0,00$), nors analogiškas dešinės rankos įvertis yra vienas geresnių. Susimaišymo matricoje matome, kad šis sutrikimas dažniausiai yra priskiriamas Y ašies diapazono sutrikimui arba drebuliui. Šią tendenciją galima dalinai paaiškinti remiantis atliktų pratimų pobūdžiu (žr. 5.3 pav.).



5.3 pav. Atliekamo pratimo pavyzdys

Dalis žiedo perkėlimo pratimų yra asimetriniai, kairėje pusėje turi kauburius. Tie asmenys, kurie turi kairės rankos diapazono sutrikimus tiek X, tiek Y ašyje, šio kauburio negali lengvai įveikti, todėl modelis jiems pritaiko neteisingą diagnozę. Tuo tarpu, pratimų, kurie tokius kauburius turėtų dešinėje pusėje nėra. Galimai dėl šios priežasties modelis dešinės rankos diapazono sutrikimus dažnu atveju atskiria geriau.



5.4 pav. „LSTM“ susimaišymo matrica

„LSTM“ modelis taip pat pasirodė sąlyginai gerai ir pasiekė 53,73 % tikslumą. Jis ypatingai gerai atpažįsta kairės rankos drebulį ($F_1 = 1,00$) bei pirštų sustingimą ($F_1 = 0,86$), tačiau visiškai negeba atskirti dešinės rankos diapazono sutrikimų X ašyje ($F_1 = 0,00$) bei dešinės plaštakos pirštų sustingimo ($F_1 = 0,04$). Tai rodo, kad šis modelis buvo linkęs per daug prisitaikyti prie tam tikrų sutrikimų, taip prarandant gebėjimą atskirti kitus. Būtent dėl šios priežasties „LSTM“ vidutinis F1 įvertis yra mažesnis nei „XGBoost“.

Apibendrinant, nors „LSTM“ tam tikrus sutrikimus geba atpažinti nepriekaištingai, jo rezultatai nėra pastovūs, o tai sumažina jo tikslumą. Tuo tarpu, nors „XGBoost“ ir nepasiekia F_1 reikšmių, kurios yra artimos vienetui, jis yra žymiai stabilesnis ir patikimesnis didesniai klasių kiekiui, todėl šis modelis yra tinkamesnis labiausiai tikėtino sutrikimo identifikavimo uždaviniams spręsti.

5.3. Daugelio sutrikimų identifikavimas

Tokie atvejai, kai pacientas turi tik vieną sutrikimą ir tik vienoje rankoje yra paprasčiausiai nerealistiški. Siekiant sukurti tokį modelį, kuris būtų tinkamas naudojimui profesionalioje aplinkoje ir sveikatos sektoriuje, mes turime naudoti modelį, kuris identifikuoja daugelį sutrikimų (*angl. multi-label*).

Šioje tyrimo dalyje buvo analizuojami „ST-GIN“, „LSTM“ ir „XGBoost“ modeliai. Čia taip pat buvo sukurtas taisyklėmis grįstas ansamblinis modelis, kuris naudoja „LSTM“ kaip pagrindinį klasifikatorių ir „XGBoost“ kaip korekcijos modelį. Siekiant suteikti atskaitos tašką, čia taip pat bus pateikiamas svertinis atsitiktinis (*angl. Weighted-Random*) modelis, kurio rezultatai apskaičiuoti per šimtą epochų, o įverčių tikimybė atitinka klasių pasiskirstymą treniravimo aibėje. Šių eksperimentų rezultatai yra pateikiami 5.5 lentelėje

5.5 lentelė. Modelių rezultatai, identifikuojant daugelį sutrikimų

Modelis	Macro preciziškumas	Macro jautrumas	Macro F_1	Tiksli atitiktis, %
Atsitiktinis	0,1121	0,1314	0,1207	7,90
ST-GIN	0,3435	0,6578	0,4097	17,29
LSTM	0,4462	0,6090	0,4546	29,19
XGBoost	0,5732	0,7247	0,6100	52,38
Ansamblis	0,5965	0,7404	0,6128	45,86

Panašiai kaip ir praeito eksperimento metu, geriausiai pasirodė „XGBoost“ modelis, o sekantis – „LSTM“. „ST-GIN“ modelis iš šių trijų veikė prasčiausiai, o tai vėl užtikrina idėją, kad kaulų struktūros formalizavimas papildomos vertės sutrikimų atpažinimui neduoda. Atvirkščiai, kaimyninių reikšmių agregavimas rezultatus tik pablogina.

Kita vertus, visi šie modeliai žymiai pranoksta atsitiktinius spėjimus, o tai parodo, kad modeliai geba įsisavinti mokymo aibės informaciją. Aukštesnis F_1 įvertis reiškia, kad šie metodai yra pajėgūs atskirti sveikus įrašus nuo tų, kuriuose yra matomi sutrikimai bei šiuos sutrikimus teisingai identifiuoti.

Tikslios atitikties įvertis taip pat nusako kokia pacientų dalis yra pilnai ir teisingai diagnozuojamos – siekiant sukurti tikslų paciento įvertinimą, net pusei jų papildomo specialisto įsiterpimo paprasčiausiai nereikėtų.

Tyrimo metu sukurtas ansamblinis taisyklėmis grįstas modelis apjungia du geriausius, t.y. „LSTM“ ir „XGBoost“, modelius. Tokia kombinacija mums leidžia pasinaudoti abiejų modelių gerosiomis savybėmis ir užpildyti vieno ar kito modelio daromas spragas. Kad tai tikrai įvyksta įrodo išaugę preciziškumo, jautrumo ir F_1 įverčiai. Taisyklės, kurios buvo naudotos srauto paskirstymui pateikiamos 5.6 lentelėje.

5.6 lentelė. Ansamblinio modelio taisyklių rinkinys

Sutrikimas	Taisyklė	F_1 prieaugis validacijos duomenų rinkinyje	Nukreipiamų įrašų dalis, %
K, diapazonas X	„LSTM“ užtikrintumo riba mažesnė nei 0,2	0,1227	7,25
K, diapazonas Y	Visada	0,2209	100,00
K, pirštų sustingimas	„LSTM“ užtikrintumo riba mažesnė nei 0,2	0,2222	22,77
K, drebulys	Niekada	0,0000	0,00
D, diapazonas X	Visada	0,2343	100,00
D, diapazonas Y	Visada	0,5292	100,00
D, pirštų sustingimas	Visada	0,2952	100,00
D, drebulys	„LSTM“ užtikrintumo riba mažesnė nei 0,1	0,0883	25,98

Čia matome, kad mūsų du modeliai užima dvi skirtingas nišas – „LSTM“ prižiūri abiejų rankų drebulio sutrikimus bei kairės rankos diapazono X ašyje ir pirštų sutrikimo problemas. „XGBoost“ modeliui yra deleguojami visi likę sutrikimai.

Verta pastebėti, kad dalis sutrikimų (t.y. kairės rankos diapazonas Y ašyje, dešinės rankos diapazonas X ir Y ašyse bei dešinės plaštakos pirštų sustingimas) yra visada nukreipiami į „XGBoost“ modelį. Tai sufleruoja, kad „LSTM“ modelis yra linkęs juos klasifikuoti prasčiau nei „XGBoost“. Tokią pačią išvadą galima padaryti pažvelgus į 5.7 lentelę.

5.7 lentelė. F_1 įverčiai kiekvienam sutrikimui

Duomenų aibė		Modelis	K, diapazonas X	K, diapazonas Y	K, pirštų sustingimas	K, drebulys	D, diapazonas X	D, diapazonas Y	D, pirštų sustingimas	D, drebulys
Validavimo	LSTM	0,7843	0,6061	0,7513	1,0000	0,7559	0,3782	0,3307	0,6328	
	XGBoost	0,7485	0,8269	0,8767	1,0000	0,9903	0,9075	0,6260	0,6329	
	Ansamblis	0,9070	0,8269	0,9735	1,0000	0,9903	0,9075	0,6260	0,7216	
Testavimo	LSTM	0,6354	0,0986	0,8295	0,9756	0,3471	0,0088	0,3396	0,4024	
	XGBoost	0,7348	0,4076	0,8136	1,0000	0,0000	0,5849	0,7239	0,5970	
	Ansamblis	0,8037	0,4076	0,9143	0,9756	0,0000	0,5849	0,7239	0,4928	

Čia galima matyti, kad būtent minėtuose sutrikimuose „LSTM“ modelis labiausiai atsilieka nuo „XGBoost“. Skirtumas tarp jų dviejų F_1 įverčių blogiausiu atveju vertinant validavimo aibę siekia net 0,5293, o tai rodo stiprų „LSTM“ modelio neapsimokymą identifikuojant šį konkretų sutrikimą. Kita vertus, likusiais atvejais šis skirtumas nėra toks drastiškas, o kairės rankos diapazono X ašyje sutrikimo identifikavimas, nors ir neženkliai ($\Delta F_1 = -0,0358$), bet pranoksta „XGBoost“ pasiektą įvertį.

Vertinant testavimo aibės rezultatus galima matyti, kad tais atvejais, kai į „XGBoost“ yra siunčiama tik dalis įrašų (kairės rankos diapazonas ir kairės plaštakos pirštų sustingimas), ansamblis gauna geresnius rezultatus nei atskiri modeliai. Tai parodo, kad ansamblio naudojimas mums atneša papildomos naudos, kadangi skirtingi modeliai mokymosi metu išmoksta sutrikimus atpažinti pagal skirtingas savybes.

Vertinant modelius, kurie yra naudojami ansamblio viduje, taip pat yra svarbu atsižvelgti ir į jų vykdymo laiką. Ši eksperimentų informacija yra pateikiama 5.8 lentelėje. Skaičiavimai buvo vykdomi naudojant „NVIDIA H100 NVL“ vaizdo plokštę.

5.8 lentelė. Modelių vykdymo laikai apdorojant 600 kadru (20 sekundžių) seką

Modelis	Vieno kadro apdorojimas, ms	30 kadru per sekundę biudžeto sunaudojimas, %	Galutinis klasifikavimas, ms	Bendras skaičiavimo laikas, ms
LSTM	0,4645±0,2388	1,3934	0,7701±0,4350	391,8928±14,0695
XGBoost	0,0041±0,0018	0,0122	203,0524±14,5987	205,8470±14,6037
Ansamblis	0,4686±0,2389	1,4057	97,7758±4,5996	491,6930±14,8082

Žiūrint į šią problemą naiviai, atsižvelgiant tik į bendrą skaičiavimo laiką, galima teigti, kad „XGBoost“ modelis yra beveik dvigubai greitesnis nei „LSTM“. Tačiau tokia išvada ignoruoja modelių skirtumus ir jų esminę panaudojimo sritį – realaus laiko sutrikimų identifikavimą.

„LSTM“ – tai rekursinis modelis, kuris kadrus nuosekliai apdoroti gali dar jų generavimo metu. Čia, vieno kadro perėjimas per „LSTM“ sluoksnį ir išsaugant sluoksnio būseną užtrunka vidutiniškai 0,4645 ms, o tai laisvai telpa į 33 ms biudžetą, kai kadru generavimo dažnis yra 30 kadru per sekundę. Papildomai, „LSTM“ sluoksnio išvesties vektorius klasifikavimas neuroniniame sluoksnyje užtrunka vos 0,7701 ms. Tai reiškia, kad didžioji modelio skaičiavimo apkrova, kadru sekos užkodavimas į vektorių, yra paskirstoma duomenų generavimo metu.

Kita vertus, „XGBoost“ modelio kadru apdorojimas yra daug paprastesnis. Šio etapo metu mums yra aktualus tik statistinių išvestinių reikšmių generavimas, o tam pasiekti pakanka paprasčiausiai agreguoti gaunamų kintamųjų reikšmes. Dėl šios priežasties, vieno kadro apdorojimas trunka 0,0041 ms. Tačiau priešingai nei „LSTM“, didžioji modelio apkrova nėra paskirstoma laike – klasifikavimas gali būti vykdomas tik galutinio etapo metu, kai yra turimos visos išvestinės reikšmės. Tai reiškia, kad vietoje paprasto neuroninio klasifikatoriaus, čia yra naudojama didelė aibė sprendimų medžių, kurių vykdymo laikas yra ženkliai ilgesnis ir siekia 205,8470 ms. Ši trukmė perkopia 30-ies kadru per sekundę biudžetą, o tai reiškia, kad skaičiavimas realiu laiku nėra įmanomas, nebent rezultatas būtų atnaujinamas rečiau. Šiuos modelius apjungus į ansamblinį modelį, klasifikavimo laikas yra sumažinamas per pusę, kadangi „XGBoost“ yra naudojamas tik keturių iš aštuonių sutrikimų klasifikavimui.

Apibendrinant, „XGBoost“ modelis buvo geriausias iš visų bazinių modelių ir pasiekė tiksliausius įverčius, o „LSTM“ liko antroje vietoje. Grafais grįstas modelis pasiekė prastesnius rezultatus, o tai rodo, kad šis sprendimas nėra tinkamas būtent šioje problemoje. Papildomai, du geriausi modeliai buvo apjungti į vieną ansamblinį taisyklėmis grįstą modelį, kurio F_1 įvertis pranoko kiekvieno modelio atskirą įvertį. Šis modelis pasinaudoja tiek „XGBoost“ gebėjimu tiksliai įvertinti sutrikimus, tiek „LSTM“ realaus laiko skaičiavimo savybėmis.

Išvados

1. Tyrimo metu buvo atlikta egzistuojančių sprendimų analizė. Jos metu buvo peržvelgtos sistemos, kurios yra panašios į kuriamą sprendimą. Vienos jų terapijos procesą paverčia žaidimu, kuris pajvairina pacientų rutiną. Kitos pasitelkia dirbtinio intelekto įrankius, siekdamas palengvinti reabilitologo darbą. Tačiau nė vienas iš analizuotų sprendimų neapjungia šių dviejų sričių – virtualiosios realybės ir dirbtinio intelekto. Tai rodo, kad ši kombinacija, kuri turi potencialą papildyti terapijos eigą, nėra pakankamai iširta, todėl kuriamas įrankis gali būti laikomas inovatyviu.
2. Viršutinės kūno dalies judėjimo sutrikimų yra daug, o atsižvelgti į kiekvieną individualų atvejį nėra įmanoma. Kadangi ši sritis vis dar nėra pakankamai išplėta, standartizuotos sutrikimų aibės neegzistuoja. Dėl šios priežasties modelių mokymui buvo naudojamos supaprastintos, apibendrintos sutrikimų klasės. Tyrime nagrinėjamas modelių gebėjimas atpažinti judėjimo diapazono sutrikimus horizontalioje ir vertikalioje plokštumoje, rankos drebinimą bei plaštakos sugniaužimo sutrikimus. Šios klasės yra taikomos tiek kairei, tiek dešinei rankai. Verta pastebėti, kad siekiant supaprastinti duomenų rinkimą, į sutrikimo sunkumą nebuvo atsižvelgta. Tokį aspektą būtų tikslinga įtraukti į tolimesnius tyrimus, siekiant sukurti sistemą, kuri geriau atitinka realius panaudojimo atvejus.
3. Toliau, šio darbo metu buvo suprojektuota ir realizuota smulkiosios motorikos lavinimo virtualiojoje realybėje sistema, kuri buvo sudaryta iš trijų posistemų. Virtualiosios realybės posistemė buvo sukurta „Unity“ aplinkoje, kurioje pacientui yra pateikiami įvairūs reabilitacijos pratimai žaidimo formatu. Čia, be papildomos įrangos poreikio, yra renkami ir išsaugomi rankų judesiai, kurie vėliau yra panaudojami dirbtinio intelekto posistemės viduje mokant modelius. Svarbi buvo ir reabilitologo sąsajos posistemė – ji leidžia vienkartinu raktu užmegzti ryšį su virtualiosios realybės įrenginiu ir nuotoliniu būdu kontroliuoti paciento pratimus, koreguoti jo poziciją. Visa sistema gali būti toliau vystoma tiek tolimesniems tyrimams, tiek pritaikymui profesionalioje aplinkoje.
4. Kadangi duomenų rinkinių, kurie atitinka šią specifinę tyrimo sritį, nepavyko rasti, realizuota reabilitacijos sistema buvo panaudota duomenų rinkimui. Duomenys buvo surinkti su sveiku asmeniu, kuris atliko pratimus ir simulavo anksčiau nustatytus sutrikimus. Sutrikimų simulavimas buvo atliekamas remiantis ekspertų reabilitologų nustatytomis gairėmis. Toks sprendimas buvo priimtas atsižvelgus į infrastruktūrinius ir etinius klausimus, kylančius dirbant su pacientais. Nors ateityje į tyrimus būtų naudinga įtraukti ir realius pacientus, šio trūkumo poveikiui mažinti buvo panaudota DTW metodu grįsta duomenų sintezė. Jos dėka surinkti sutrikimai buvo kombinuojami poromis taip išplečiant galimų sutrikimų aibę ir suteikiant modeliams daugiau pavyzdžių mokymo metu. Be to, duomenų rinkimas naudojantis virtualiosios realybės akinių suteikiama rankų atpažinimo programavimo sąsaja taip pat buvo sėkmingas – jis buvo tikslesnis, pigesnis ir lengviau prieinamas nei alternatyvos.
5. Surinkus ir apdorojus duomenis buvo vykdomi mažesnės parametrų aibės paieškos eksperimentai – godus kaulų atmetimas mokant „LSTM“ modelį bei pozicinių ir pasisukimo parametrų atskyrimas. Pirmojo eksperimento metu buvo pastebėta, kad modelio tikslumas ir kaulų skaičius duomenų rinkinyje turi labai silpną neigiamą koreliaciją ($r = -0,197$). Dėl šios priežasties yra daroma išvada, kad 26 kaulų plaštakos reprezentacija yra perteklinė, o ją

galima sumažinti vos iki 6 svarbiausių kaulų, t.y. delno ir kiekvieno plaštakos piršto. Duomenų rinkinį taip pat galima optimizuoti atmetant pozicinius kaulų parametrus. Atlikus tokį pakeitimą, „XGBoost“ modelio tikslumas išaugo net 14,03 %. Tai rodo, kad pasisukimo parametrai yra ypač svarbūs aptinkant didžiąją sutrikimų dalį, o jų pašalinimas reikšmingai sumažina modelių tikslumą. Šiuos pakeitimus kombinuojant, duomenų rinkinys sumažėja nuo 365 stulpelių iki 25, o tai leidžia efektyviau rinkti ir saugoti reabilitacijos sesijų informaciją bei sukurti greitesnius ir tikslesnius modelius.

6. Toliau buvo vykdomas detalesnis sutrikimų klasifikavimo uždavinio tyrimas. Jo metu buvo analizuojami „LSTM“, „XGBoost“, taisyklėmis grįstas ansamblinis modelis bei grafais grįsti modeliai („ST-GCN“ ir „ST-GIN“). Pastarieji visų bandymų metu pasirodė prasčiausiai, todėl galima teigti, kad kaulų struktūros formalizavimas šiam uždaviniui nėra naudingas. „XGBoost“ modelis buvo tiksliausias modelis iš visų, jis lenkė „LSTM“ tiek identifikuojant labiausiai tikėtiną sutrikimą ($\bar{F}_1 = 0,5591$), tiek aptinkant sutrikimų aibę ($\bar{F}_1 = 0,6100$). Kita vertus, „LSTM“ modelis pasižymėjo geresne greitaveika, kadangi jo skaičiavimus galima paskirstyti laike. Vieno kadro apdorojimas ir būsenos saugojimas „LSTM“ sluoksnyje užtrunka 0,4645 ms, o šio sluoksniu išvesties klasifikavimas vos 0,7701 ms. Tuo tarpu sutrikimų aibės klasifikavimas su „XGBoost“ užtrunka 203,0524 ms. Šios dvi svarbios savybės buvo apjungtos tarpusavyje sukuriant ansamblinį modelį, kurio tikslumas pranoko „XGBoost“ ($\bar{F}_1 = 0,6128$), o klasifikavimo laikas buvo sumažintas iki 97,7758 ms.

Literatūros sąrašas

1. DAS, A. ir kt. Towards Intelligent Extended Reality in Stroke Rehabilitation. *Augmenting Neurological Disorder Prediction and Rehabilitation Using Artificial Intelligence*. 2022. p. 309–329, doi: 10.1016/B978-0-323-90037-9.00006-0.
2. WODU, C.O. ir kt. Exploring the Reasons Behind the Low Focus on Upper Limb Rehabilitation in the Early Stages After a Stroke: A Qualitative Study. *Journal of Hand Therapy*. 2024, doi: 10.1016/j.jht.2024.08.005.
3. LIU, Z. ir kt. Application of Immersive Virtual-Reality-Based Puzzle Games in Elderly Patients with Post-Stroke Cognitive Impairment: A Pilot Study. *Brain Sciences*. 2023. vol. 13, no. 1, p. 79, doi: 10.3390/brainsci13010079.
4. HADJIPANAYI, C. ir kt. Virtual Reality Exergames For Enhancing Engagement in Stroke Rehabilitation: A Narrative Review. *Heliyon*. 2024. vol. 10, no. 18, p. e37581, doi: 10.1016/j.heliyon.2024.e37581.
5. MU, P. ir kt. Application of Artificial Intelligence in Rehabilitation Assessment. *Journal of physics: Conference Series*. 2021. vol. 1802, no. 3, Article 032057, doi: 10.1088/1742-6596/1802/3/032057.
6. EL NAAMANI, K. ir kt. The Artificial Intelligence Revolution in Stroke Care: A Decade of Scientific Evidence in Review. *World neurosurgery*. 2024. vol. 184, p. 15–22, doi: 10.1016/j.wneu.2024.01.012.
7. SARWAT, H. ir kt. Design of a Data Glove for Assessment of Hand Performance Using Supervised Machine Learning. *Sensors*. 2021. vol. 21, no. 21, p. 6948, doi: 10.3390/s21216948.
8. CHEN, T. ir GUESTRIN, C. XGBoost: a Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. 2016. p. 785–794, doi: 10.1145/2939672.2939785.
9. VARGA, G. ir kt. Comparison of Data Classification Results in Serious Gaming for Rehabilitation of Rheumatoid Arthritis. *Studies in health technology and informatics*. 2023, vol 309, p. 63-67 doi: 10.3233/SHTI230740.
10. LATRECHE, A. ir kt. A New Home-Based Upper- and Lower-Limb Telerehabilitation Platform with Experimental Validation. *Arabian Journal for Science and Engineering*. 2023, doi: 10.1007/s13369-023-07720-0.
11. HOCHREITER, S. ir SCHMIDHUBER, J. Long short-term memory. *Neural Computation*. 1997. vol. 9, no. 8, p. 1735–1780, doi: 10.1162/neco.1997.9.8.1735.
12. VASWANI, A. ir kt. Attention is all you need. *Advances in neural information processing systems*. 2017. doi: 10.48550/arXiv.1706.03762.
13. POPEL, M. ir BOJAR, O. Training tips for the transformer model. *The Prague Bulletin of Mathematical Linguistics*. 2018. Vol 110, no. 1, p. 43-70, doi: 10.2478/pralin-2018-0002.

14. YAN, S. ir kt. Spatial temporal graph convolutional networks for skeleton-based action recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2018. vol. 32, no. 1, doi: 10.1609/aaai.v32i1.12328.
15. XU K., ir kt. How Powerful are Graph Neural Networks? *International Conference on Learning Representations (ICLR)*, 2019. doi: 10.48550/arXiv.1810.00826.
16. Unity Technologies. Unity Asset Store produktų paieška [interaktyvus]. *Unity Asset Store*. [žiūrėta: 2026 m. gegužės 11 d.] Prieiga per internetą: <https://assetstore.unity.com/search>.
17. Epic Games, Inc. FAB produktų paieška [interaktyvus]. *FAB*. [žiūrėta: 2026 m. gegužės 11 d.] Prieiga per internetą: <https://www.fab.com/search>.
18. The Godot Engine community. Godot Asset Library [interaktyvus]. *Godot Engine*. [žiūrėta: 2026 m. gegužės 11 d.] Prieiga per internetą: <https://godotengine.org/asset-library/asset>.
19. Steam žaidimų naudojamos technologijos [interaktyvus]. *SteamDB*. [žiūrėta: 2026 m. gegužės 11 d.] Prieiga per internetą: <https://steamdb.info/tech/>.
20. Unity Technologies. Hand data model: XR hands: 1.4 [interaktyvus]. *Unity*. [žiūrėta: 2026-01-19]. Prieiga per internetą: <https://docs.unity3d.com/Packages/com.unity.xr.hands@1.4/manual/hand-data/xr-hand-data-model.html>.