



Kauno technologijos universitetas

Informatikos fakultetas

Virtualiosios realybės rankų sekimo prognozavimo, naudojant rekurentinius neuroninius tinklus, tyrimas

Baigiamasis magistro studijų projektas

Gytis Dokšas

Projekto autorius

Prof. Dr. Tomas Blažauskas

Vadovas

Kaunas, 2025



Kauno technologijos universitetas

Informatikos fakultetas

Virtualiosios realybės rankų sekimo prognozavimo, naudojant rekurentinius neuroninius tinklus, tyrimas

Baigiamasis magistro studijų projektas

Dirbtinio intelekto informatika (6211BX007)

Gytis Dokšas

Projekto autorius

Prof. Dr. Tomas Blažauskas

Vadovas

Prof. Dr. Rytis Maskeliūnas

Recenzentas

Kaunas, 2025



Kauno technologijos universitetas

Informatikos fakultetas

Gytis Dokšas

Virtualiosios realybės rankų sekimo prognozavimo, naudojant rekurentinius neuroninius tinklus, 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.

Gytis Dokšas

Patvirtinta elektroniniu būdu

Dokšas, Gytis. Virtualiosios realybės rankų sekimo prognozavimo, naudojant rekurentinius neuroninius tinklus, tyrimas. Magistro studijų baigiamasis projektas / vadovas prof. dr. Tomas Blažauskas; Kauno technologijos universitetas, Informatikos fakultetas.

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

Reikšminiai žodžiai: virtuali realybė, rankų sekimas, rekurentiniai neuroniniai tinklai.

Kaunas, 2025. 117 p.

Santrauka

Virtualios realybės (VR) technologijos vis populiarėja – išleidžiama nauja techninė įranga, pritaikomi vis pažangesni algoritmai, kuriamas didelis kiekis VR technologijas naudojančių sistemų, plečiasi VR taikymas įvairiose srityse. Rankų sekimas yra viena svarbiausių technologijų virtualiosios realybės sistemose, siekiant sukurti realistišką naudotojo sąsają. „Meta Quest 2“ sistema palaiko rankų sekimą naudojant tik akinius be papildomų valdiklių, tačiau susiduriama su problemomis: rankų sekimas sutrinka uždengus dalį rankos, naudojant papildomus įrenginius ant rankų (pavyzdžiui, haptines pirštines) arba esant neoptimalioms apšvietimo sąlygoms. Šiais atvejais rankų sekimas nutrūksta. Darbo tikslas – pasiūlyti dirbtinio intelekto metodus, įgyvendinančius rankos pirštų sąnarių pozicijos ir pasukimo erdvėje prognozes naudojant virtualiosios realybės sekimo priemones ir ištirti jų efektyvumą. Pagrindinis dėmesys skiriamas trumpalaikiam prognozavimui, siekiant kompensuoti momentinį sekimo praradimą, taip padidinant VR programų tęstinumą ir tikroviškumą.

Sistemai kurti naudojami rekurentiniai neuroniniai tinklai (angl. *recurrent neural networks*, *RNN*): modifikuoti ilgalaikės trumpalaikės atminties (angl. *long short-term memory*, *LSTM*) ir vartais paremtų rekurentinių vienetų tinklai (angl. *gated recurrent unit*, *GRU*). Duomenų rinkinys sudaromas ir vizualizuojamas naudojant „Unity“ su „OpenXR“ paketu. Atliekamas papildomas tyrimas siekiant išsiaiškinti optimalų dvipusės komunikacijos metodą – komunikacija tarp dirbtinio intelekto metodų ir VR posistemių užtikrinama naudojant „pythonnet“ paketą. Penkiolikos paskutinių kadro pozicija, pasukimas, linijinis ir kampinis greitis yra naudojami ateities kadro prognozėms atlikti kai sutrinka rankų sekimas.

Eksperimentų rezultatai su optimalia parametų kombinacija pasiekia 1,11 cm vidutinę Euklido pozicijos prognozavimo paklaidą ir 8,04 laipsnių vidutinę kampinę paklaidą prognozuojant vieną kadrą į ateitį. Prognozuojant tik pasukimą gaunama 5,2 laipsnių vidutinę kampinę paklaidą. Daugelio kadro prognozavimo eksperimentai rodo, kad autoregresinis mokymas pagerina ilgalaikį prognozavimo tikslumą. Pasiekiamos vidutinės Euklido pozicijų paklaidos yra 1,47 cm, 1,62 cm, 3,36 cm ir 10,65 cm prognozuojant 1, 5, 10 ir 30 kadro į ateitį atitinkamai. Pasiekti rezultatai parodo, kad sistema esant trikdžiams geba pagerinti VR rankų sekimą be papildomos aparatinės įrangos nesulėtinant greitaveikos, leidžiant prognozes atlikti realiu laiku.

Dokšas, Gytis. Research on Virtual Reality Hand Tracking Prediction Using Recurrent Neural Networks. 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, hand tracking, recurrent neural networks.

Kaunas, 2025. 117 pages.

Summary

Virtual reality (VR) technologies are becoming increasingly popular, with new hardware releases, more advanced algorithms being used, increase in VR-enabled systems and a broader range of VR usage in various fields. Hand tracking is one of the most important technologies in VR systems when creating an immersive user interface. Meta Quest 2 virtual reality system supports hand tracking using only headsets with no additional controllers but encounters problems: hand tracking is impaired if part of hand is occluded during movements, additional devices (e.g. haptic gloves) are worn on hands or when lighting conditions are sub-optimal. These issues cause loss of tracking. The main aim of this research is to propose artificial intelligence (AI) methods for hand joint spatial data predictions using virtual reality systems when tracking is lost and evaluate their performance. The focus is on short-term predictions to compensate for common loss of tracking, thus increasing the continuity and immersion of VR applications that use hand tracking.

Recurrent neural networks (RNN) are used, including long short-term memory (LSTM) and gated recurrent unit (GRU) networks with modifications. The dataset is constructed and visualized using Unity with OpenXR package. Additional research is done to find performant bi-directional communication between AI methods and VR subsystems – pythonnet package is used. Hand joint position, rotation, linear and angular velocity of last fifteen frames is used to make predictions of next joint spatial data frame in case tracking is lost.

Provided experimentation suggests that optimal combination of parameters is able to achieve an average Euclidean position prediction error of 1.11 cm and an average angular error of 8.04 degrees when predicting a single frame into the future. Separate model suggests that rotation can be predicted with average angular error of 5.2 degrees. Multi-frame prediction experiments show that autoregressive training improves long-term prediction accuracy. Average Euclidean positions errors of 1.47 cm, 1.62 cm, 3.36 cm and 10.65 cm were achieved for 1, 5, 10 and 30 frame prediction into the future respectively. The results show that the system is capable of improving VR hand tracking without additional hardware or loss of performance in real time applications.

Turinys

Lentelių sąrašas	8
Paveikslų sąrašas	9
Santrumpų ir terminų sąrašas	11
Įvadas.....	12
Projekto naujumas ir aktualumas.....	12
Tikslas ir uždaviniai	12
Dokumento struktūra	12
1. Rankų sąnarių informacijos prognozių sprendimų ir dirbtinio intelekto metodų analizė	13
1.1. Technologijų analizė	13
1.1.1. Virtualiosios realybės sistemų analizė.....	13
1.1.2. Žaidimo kūrimo įrankių analizė	15
1.1.3. „Unity“ žaidimų kūrimo įrankio palaikomų įskiepių analizė.....	15
1.2. Egzistuojančių sprendimų analizė	16
1.2.1. Rankų sekimo taikymas ir įvesties būdai	16
1.2.2. Gestų klasifikavimas ir aptikimui taikomi dirbtinio intelekto metodai.....	20
1.2.3. Sąnarių pozicijos ir orientacijos prognozės	22
1.2.4. Sprendimų analizės išvados.....	24
2. Rankų sąnarių virtualiojoje realybėje prognozių sistemos projektavimas	26
2.1. Sistemos funkcijos.....	26
2.1.1. Bendras sistemos funkcionalumas.....	27
2.1.2. Duomenų rinkinio kūrimo posistemė	28
2.1.3. Rankų sekimo tobulinimo posistemė.....	33
2.1.4. Gestų aptikimo posistemė.....	36
2.2. Diegimo diagrama	40
2.3. Realizacijai keliami nefunkciniai reikalavimai	41
2.4. Techninė specifikacija	41
2.5. Dirbtinio intelekto algoritmai	43
2.6. Metrikos ir klaidos funkcijos.....	44
2.7. Testavimas	48
2.7.1. Testavimo planas	48
2.7.2. Testavimo kriterijai	48
3. Rankų sąnarių virtualiojoje realybėje prognozių sistemos įgyvendinimas	50
3.1. Duomenų surinkimo ir vizualizavimo posistemės įgyvendinimas	50
3.1.1. Komunikacija tarp <i>C#</i> ir <i>Python</i> posistemų	50
3.1.2. Duomenų surinkimas ir vizualizavimas	53
3.2. Rankų sekimo tobulinimo posistemės įgyvendinimas	57
3.2.1. Duomenų nuskaitymas ir transformavimas	57
3.2.2. Modelių architektūra	58
3.2.3. Papildomas funkcionalumas tikslumui didinti	63
3.2.4. Treniravimas.....	64
3.2.5. Prognozuotų pozicijų ir pasukimų erdvėje vizualizavimas naudojant „Unity“	67
4. Rankų sąnarių pozicijų ir pasukimo prognozių eksperimentai ir rezultatai.....	70

4.1. Eksperimentai keičiant naudojamą duomenų formatą	72
4.2. Eksperimentai keičiant architektūros parametrus	76
4.3. Eksperimentai keičiant įvesties ypatybes	83
4.4. Eksperimentai keičiant įvesties ir išvesties ypatybes	89
4.5. Eksperimentai keičiant kitus parametrus	91
4.6. Eksperimentų rezultatų palyginimas	97
4.7. Kelių kadrų prognozių eksperimentai	98
4.8. Eksperimentai naudojant papildytą duomenų rinkinį	106
5. Išvados	111
Literatūros sąrašas	113
Priedai.....	117
1 priedas. Dalyvavimas IVUS 2025 (angl. <i>30th International Conference Information Society and University Studies</i>) tarptautinėje konferencijoje.....	117
2 priedas. Straipsnis ALTA 2024 (angl. <i>Avanced Learning Technologies and Applications</i>) tarptautinės konferencijos leidinyje.....	117

Lentelių sąrašas

1.1 lentelė. VR sistemų, palaikančių rankų sekimą, sąrašas.....	13
1.2 lentelė. Žaidimų kūrimo įrankių palyginimas.....	15
1.3 lentelė. Palaikomų įskiepių palyginimas	16
2.1 lentelė. „Žvalgytis VR erdvėje“ specifikacija	27
2.2 lentelė. „Judinti rankas“ specifikacija	28
2.3 lentelė. „Papildyti sekimo tobulinimo rinkinį“ specifikacija.....	29
2.4 lentelė. „Papildyti gestų duomenų rinkinį“ specifikacija	30
2.5 lentelė. „Atlikti modelių treniravimą“ specifikacija.....	31
2.6 lentelė. „Keisti nustatymus“ specifikacija	34
2.7 lentelė. „Aptikti gestą“ specifikacija	37
2.8 lentelė. „Pridėti gestą“ specifikacija.....	38
2.9 lentelė. Minimali ir rekomenduojama techninė įranga.....	42
3.1 lentelė. Sekami rankos sąnariai	54
3.2 lentelė. Specifinių sąnarių svoriai.....	64
4.1 lentelė. Erdvės kitimo eksperimentai naudojant GRU architektūrą	72
4.2 lentelė. Erdvės kitimo eksperimentai kai naudojant LSTM architektūrą	74
4.3 lentelė. Architektūros dydžio kitimo eksperimentai naudojant LSTM architektūrą	76
4.4 lentelė. Architektūros dydžio kitimo eksperimentai naudojant GRU architektūrą	78
4.5 lentelė. Architektūros dydžio kitimo eksperimentai naudojant LSTM ir GRU architektūras su dėmesio mechanizmais	80
4.6 lentelė. 5 geriausios architektūros lyginant bendrą įvertį	82
4.7 lentelė. 5 geriausios architektūros lyginant pozicijos įvertį	82
4.8 lentelė. 5 geriausios architektūros lyginant orientacijos įvertį	82
4.9 lentelė. Įvesties ypatybių kitimo eksperimentai naudojant GRU (512, 2 sluoksnių) architektūrą	83
4.10 lentelė. Įvesties ypatybių kitimo eksperimentai naudojant LSTM (1024, 2 sluoksnių) architektūrą	85
4.11 lentelė. Įvesties ypatybių kitimo eksperimentai naudojant GRU (512, 1 sluoksnio) architektūrą	87
4.12 lentelė. Įvesties ir išvesties ypatybių kitimo eksperimentai naudojant LSTM (1024, 2 sluoksnių) architektūrą	89
4.13 lentelė. Įvesties ir išvesties ypatybių kitimo eksperimentai naudojant LSTM (1024, 2 sluoksnių) architektūrą su 256 duomenų kiekiu.....	91
4.14 lentelė. Įvesties ir išvesties ypatybių kitimo eksperimentai naudojant LSTM (1024, 2 sluoksnių) architektūrą su 512 duomenų kiekiu.....	93
4.15 lentelė. Įvesties ir išvesties ypatybių kitimo eksperimentai naudojant LSTM (1024, 2 sluoksnių) architektūrą su 1024 duomenų kiekiu.....	95
4.16 lentelė. Geriausios kombinacijos pagal bendrą įvertį.....	97
4.17 lentelė. Geriausios kombinacijos pagal pozicijos įvertį	97
4.18 lentelė. Geriausios kombinacijos pagal orientacijos įvertį	98
4.19 lentelė. Kelių kadrų prognozių eksperimentų rezultatai.....	99
4.20 lentelė. Kelių kadrų prognozių eksperimentų su papildytu duomenų rinkiniu rezultatai.....	106

Paveikslų sąrašas

1.1 pav. Sensoriais pagrįstos pirštinės pavyzdys [16]	18
1.2 pav. Haptinių pirštinių pavyzdžiai („SenseGlove Nova“, „Manus Prime X Haptic VR“, „HaptX“, „TeslaSuit“) [19].....	19
2.1 pav. Sistemos panaudojimo atvejų diagrama	26
2.2 pav. „Žvalgytis VR erdvėje“ veiklos diagrama	27
2.3 pav. „Judinti rankas“ veiklos diagrama	28
2.4 pav. Duomenų rinkinio kūrimo posistemės panaudojimo atvejų diagrama	29
2.5 pav. „Papildyti sekimo tobulinimo rinkinį“ veiklos diagrama	30
2.6 pav. „Papildyti gestų duomenų rinkinį“ veiklos diagrama	31
2.7 pav. „Atlikti modelių treniravimą“ veiklos diagrama	32
2.8 pav. „Parinkti modelį“ veiklos diagrama.....	33
2.9 pav. „Parinkti parametrus“ veiklos diagrama	33
2.10 pav. Rankų sekimo tobulinimo posistemės panaudojimo atvejų diagrama.....	34
2.11 pav. „Keisti nustatymus“ veiklos diagrama.....	35
2.12 pav. „Pasirinkti modelį“ veiklos diagrama	35
2.13 pav. „Pakeisti interpoliavimą“ veiklos diagrama	36
2.14 pav. Gestų aptikimo posistemės panaudojimo atvejų diagrama.....	36
2.15 pav. „Aptikti gestą“ veiklos diagrama	37
2.16 pav. „Keisti modelį“ veiklos diagrama.....	38
2.17 pav. „Pridėti gestą“ veiklos diagrama.....	39
2.18 pav. „Įvesti pavadinimą“ veiklos diagrama	40
2.19 pav. „Užfiksuoti gesto veiksmus“ veiklos diagrama	40
2.20 pav. Sistemos diegimo diagrama, kai naudojamas „Pythonnet“ paketas	41
2.21 pav. RNN architektūros pavyzdys [45]	43
2.22 pav. RNN ir įprastų neuroninių tinklų, kuriuose perdavimas vyksta į priekį palyginamasis paveikslėlis [45].....	44
2.23 pav. RNN ir LSTM palyginimas [46].....	44
3.1 pav. Komunikacijos greitaveikos bandymų schema.....	50
3.2 pav. UDP tinklo lizdų rezultatai	51
3.3 pav. TCP tinklo lizdų rezultatai	52
3.4 pav. gRPC rezultatai	52
3.5 pav. „Pythonnet“ rezultatai.....	53
3.6 pav. Įgyvendintas rankos sąnarių vaizdavimas virtualioje erdvėje	53
3.7 pav. „XR Hands“ statinių gestų aptikimo naudotojo sąsaja (identiška informacija pateikiama ir dešinei rankai).....	56
3.8 pav. GRU architektūra.....	60
3.9 pav. GRU su dėmesio mechanizmu.....	61
3.10 pav. LSTM architektūra.....	62
3.11 pav. LSTM su dėmesio mechanizmu	63
3.12 pav. Pozicijos prognozės metrikos, prognozuotų ir tikslinių reikšmių palyginimas (anglų k.)	65
3.13 pav. Klaidos funkcijų ir sąnarių atstumų kitimas per paskutines 25 epochas treniravimo metu.	66
3.14 pav. Sąnarių atstumų kitimas per paskutines 25 epochas treniravimo metu.	67
3.15 pav. Vizualizuota prognozuota ranka	68

3.16 pav. Vizualizuota sekama (pilka) ranka ir prognozuota (geltona) ranka, kai atliekamas judesys į kairę	69
4.1 pav. Erdvės kitimo eksperimentai naudojant GRU architektūrą	73
4.2 pav. Erdvės kitimo eksperimentai naudojant LSTM architektūrą	75
4.3 pav. Architektūros dydžio kitimo eksperimentai naudojant LSTM architektūrą	77
4.4 pav. Architektūros dydžio kitimo eksperimentai naudojant GRU architektūrą	79
4.5 pav. Architektūros dydžio kitimo eksperimentai naudojant LSTM ir GRU architektūras su dėmesio mechanizmais	81
4.6 pav. Įvesties ypatybių kitimo eksperimentai naudojant GRU (512, 2 sluoksnių) architektūrą ...	84
4.7 pav. Įvesties ypatybių kitimo eksperimentai naudojant LSTM (1024, 2 sluoksnių) architektūrą	86
4.8 pav. Įvesties ypatybių kitimo eksperimentai naudojant GRU (512, 1 sluoksnių) architektūrą ...	88
4.9 pav. Įvesties ir išvesties ypatybių kitimo eksperimentai naudojant LSTM (1024, 2 sluoksnių) architektūrą	90
4.10 pav. Įvesties ir išvesties ypatybių kitimo eksperimentai naudojant LSTM (1024, 2 sluoksnių) architektūrą su 256 duomenų kiekiu	92
4.11 pav. Įvesties ir išvesties ypatybių kitimo eksperimentai naudojant LSTM (1024, 2 sluoksnių) architektūrą su 512 duomenų kiekiu	94
4.12 pav. Įvesties ir išvesties ypatybių kitimo eksperimentai naudojant LSTM (1024, 2 sluoksnių) architektūrą su 1024 duomenų kiekiu	96
4.13 pav. Pozicijos metrikų rezultatai, kai naudojamas įprastas modelis	100
4.14 pav. Orientacijos metrikų rezultatai, kai naudojamas įprastas modelis	101
4.15 pav. Pozicijos metrikų rezultatai, kai naudojamas naujas modelis, treniruotas naudojant 5 kadrus	102
4.16 pav. Orientacijos metrikų rezultatai, kai naudojamas naujas modelis, treniruotas naudojant 5 kadrus	103
4.17 pav. Pozicijos metrikų rezultatai, kai naudojamas naujas modelis, treniruotas naudojant 10 kadru	104
4.18 pav. Orientacijos metrikų rezultatai, kai naudojamas naujas modelis, treniruotas naudojant 10 kadru	105
4.19 pav. Pozicijos metrikų rezultatai su papildytų duomenų rinkiniu, kai naudojamas įprastas modelis	107
4.20 pav. Orientacijos metrikų rezultatai su papildytų duomenų rinkiniu, kai naudojamas įprastas modelis	108
4.21 pav. Pozicijos metrikų rezultatai su papildytų duomenų rinkiniu, kai naudojamas autoregresinis modelis, treniruotas naudojant 5 kadrus	109
4.22 pav. Pasukimo metrikų rezultatai su papildytų duomenų rinkiniu, kai naudojamas autoregresinis modelis, treniruotas naudojant 5 kadrus	110

Santrumpų ir terminų sąrašas

Santrumpos:

CNN (angl. *convolutional neural network*) – konvoliucinis neuroninis tinklas;

DI – dirbtinis intelektas;

GRU (angl. *gated reccurent unit*) – vartais paremtų rekurentinių vienetų tinklai;

HCI (angl. *human-computer interaction*) – žmogaus ir kompiuterio sąveika;

LSTM (angl. *long short-term memory*) – ilagalaiškės trumpalaikės atminties tinklas;

MAE (angl. *mean absolute error*) – vidutinė absoliutinė paklaida;

MED (angl. *mean Euclidean distance*) – vidutinis Euklido atstumas;

RNN (angl. *recurrent neural network*) – rekurentinis neuroninis tinklas;

RMSE (angl. *root mean square error*) – kvadratinės vidurkio šaknies paklaida;

SDK (angl. *software development kit*) – programinės įrangos kūrimo rinkinys;

VR – virtuali realybė;

XR (angl. *extended reality*) – išplėstoji realybė.

Terminai:

Virtuali realybė (VR) – kompiuteriu sukurta trimatė aplinka, kuri imituoja realų arba fiktyvų pasaulį ir leidžia vartotojui sąveikauti su šia aplinka naudojant specialią įrangą.

Rekurentinis neuroninis tinklas (RNN) – neuroninio tinklo architektūra, naudojama sekų apdorojimui, kai reikalingas kontekstas iš ankstesnių įrašų, pvz., laiko eilutės duomenys.

LSTM ir **GRU** – specializuoti RNN modeliai, pritaikyti ilgesnių priklausomybių sekose išsaugojimui, plačiai naudojami laiko eilučių prognozėms ir judesių modeliavimui.

OpenXR – atviro standarto specifikacija („Unity“ atveju paketas), leidžianti apdoroti įvairių gamintojų VR įrenginių įvestis ir išvestis.

Pythonnet – biblioteka, leidžianti naudoti „Python“ programavimo kalba parašytą kodą „C#“ programavimo kalba kuriamose aplikacijose, užtikrinant komunikaciją tarp skirtingų programavimo kalbų komponentų.

Autoregresinis mokymas – modelio mokymo metodas, kai kiekvienas prognozuojamas žingsnis remiasi ankstesniais to paties modelio prognozuotais duomenimis.

Išplėstoji realybė – bendrinis terminas, apimantis virtualiąją realybę, papildytąją realybę ir mišriąją realybę.

Įvadas

Projekto naujumas ir aktualumas

Virtualios realybės (VR) technologijos vis populiarėja – išleidžiama nauja techninė įranga, pritaikomi vis pažangesni algoritmai, kuriamas didelis kiekis VR technologijas naudojančių sistemų, plečiasi VR taikymas įvairiose srityse. Tačiau, nepaisant didėjančio populiarumo ir progreso šioje srityje, pakankamai mažai dėmesio skiriama rankų sekimui ir gestų aptikimui, kai nenaudojami papildomi valdikliai – pirštinės ar papildomos kameros. „Meta Quest 2“ sistema palaiko rankų sekimą naudojant akinius be papildomų valdiklių, tačiau susiduriama su problemomis – rankų sekimas sutrinka uždengus dalį rankos, naudojant papildomus įrenginius ant rankų (pavyzdžiui, haptines pirštines) arba esant neoptimalioms apšvietimo sąlygoms. Siekiant išspręsti šias problemas atliekama jau egzistuojančių sprendimų analizė ir kuriama sistema, naudojanti dirbtinio intelekto algoritmus, kuri leidžia prognozuoti rankos pirštų sąnarių pozicijas ir pasukimą kai sekimas nutrūksta. Pagrindinis dėmesys skiriamas trumpalaikiam prognozavimui, siekiant kompensuoti momentinį sekimo praradimą, taip padidinant VR programų tęstinumą ir tikroviškumą.

Tikslas ir uždaviniai

Darbo tikslas – pasiūlyti dirbtinio intelekto metodus, įgyvendinančius rankos pirštų sąnarių pozicijos ir pasukimo erdvėje prognozes naudojant virtualiosios realybės sekimo priemones ir ištirti jų efektyvumą. Šiam tikslui pasiekti iškelti uždaviniai:

1. Išnagrinėti virtualiosios realybės įrankius, leidžiančius atlikti rankų sekimą.
2. Išanalizuoti dirbtinio intelekto metodus, skirtus sąnarių pozicijų ir pasukimų prognozėms atlikti.
3. Suprojektuoti rankos sąnarių prognozių prototipą.
4. Pasiūlyti ir įgyvendinti dirbtinio intelekto metodus sąnarių pozicijos ir pasukimo prognozavimui atlikti.
5. Įvertinti pasiūlytų algoritmų efektyvumą atliekant eksperimentinius tyrimus.

Dokumento struktūra

Darbas susideda iš penkių pagrindinių dalių. Pirmoje dalyje pateikiama technologijų analizė, sudaryta iš virtualios realybės sistemų apžvalgos ir žaidimų kūrimo įrankių („Unity“ ir „Unreal Engine“) palyginimo. Analizuojamas rankų sekimo taikymas, įvestis ir egzistuojantys sprendimai, susiję su judesių sekimo prognozėmis. Antroje dalyje aprašomi funkciniai ir nefunkciniai reikalavimai, parenkami dirbtinio intelekto (DI) metodai, pasirinktos metrikos, klaidos funkcijos ir aprašomas testavimas. Trečioje dalyje pateikiamas sistemos įgyvendinimas, sudarytas iš duomenų surinkimo ir vizualizavimo posistemės aprašymo ir naudotų dirbtinio intelekto algoritmų, skirtų rankų sekimo prognozėms atlikti, įgyvendinimo aprašų. Ketvirtoje dalyje pateikiami eksperimentiniai tyrimai ir jų rezultatai, įvertinamas modelio tikslumas prognozuojant rankų sąnarių padėtį ir orientaciją, lyginami skirtingai modeliai ir parenkami optimalūs modelių parametrai. Penktoje dalyje pateikiamos išvados ir tolimesni sprendimai.

1. Rankų sąnarių informacijos prognozių sprendimų ir dirbtinio intelekto metodų analizė

Šiame skyriuje pateikiama informacija apie VR sistemas, palaikančias rankų sekimą be papildomų įrenginių. Atliekama egzistuojančių sprendimų analizė – pateikiamos rankų sekimo taikymo sritys, gestų aptikimas, su sąnarių erdvinės informacijos prognozėmis susiję moksliniai darbai.

1.1. Technologijų analizė

Poskyryje analizuojamos ir aprašomos virtualiosios realybės sistemos, lyginami žaidimų kūrimo varikliai, įrankiai ir įskiepai.

1.1.1. Virtualiosios realybės sistemų analizė

Atlikta virtualiosios realybės sistemų, leidžiančių rankos sąnarių pozicijų sekimą be papildomų valdiklių rankose, analizė. Įrenginiai, palaikantys rankų sekimą naudojant virtualiosios realybės akinius pateikiami lentelėje (žr. 1.1 lentelė).

1.1 lentelė. VR sistemų, palaikančių rankų sekimą, sąrašas

Gamintojas	Sistema
HTC Vive	Focus 3
	Focus 3 Plus
	Flow
Varjo	XR-3
	VR-3
	VR-2 Pro
Ultraleap ¹	Leap Motion Controller
	Leap Motion Controller 2
	Gemini
Pimax ²	Artisan
	5k+
	8K
	XR
	8k+
	8KX
Meta	Quest 2
	Quest 3

„HTC Vive“ – virtualiosios ir maišytosios (angl. *mixed*) realybės įrenginių linija, sukurta HTC (angl. *High Tech Computer*) korporacijos, esančios Taivane, Kinijos Respublikoje. Šiai linijai priklauso didelė dalis populiarių virtualiosios realybės akinių sistemų serijų: „XR Elite“, „Flow“, „Pro“, „Focus“ [1]. Taip pat galima įsigyti VR aksesuarų – papildomų sekimo įrenginių. Šiuo metu rankų sekimą palaiko „Focus 3“ ir „Flow“ įrenginiai naudojant „VIVE Wave“ programinės įrangos kūrimo

¹ Leap Motion Controller ir Leap Motion Controller 2 yra papildomi įrenginiai, kuriuos galima naudoti su kitoms VR sistemomis siekiant įgyvendinti rankų sekimą be valdiklių.

² Reikalingas papildomas įrenginys, kuris jungiasi prie VR akinių.

rinkinį (angl. *software development kit*, *SDK*). Įgyvendintas virtualių rankų pateikimas ir pirštų suspaudimo gesto aptikimas [2].

„Varjo“ – suomių virtualiosios, papildytos ir maišytosios realybės akinių gamintojas. „XR-3“, „VR-3“ palaiko rankų sekimą pagal nutylėjimą naudojant „Varjo Base“ programinę įrangą. „VR-2 Pro“ akiniai rankų sekimą palaiko naudojant „Ultraleap“ SDK [3]. Šio gamintojo įrenginiai palaiko pirštų ir kumščio suspaudimo gestus, leidžia kurti sistemas naudojant „Unity“ ar „Unreal Engine“ žaidimų kūrimo variklius su „OpenXR“ įskiepio pagalba.

„Ultraleap“ – įmonė, kuri susikūrė, kai 2019 m. susijungė „Leap Motion“ ir „Ultrahaptics“. Tai įmonė, kurianti virtualiosios realybės įrangą ir kitus įrenginius, leidžiančius sąveikauti su virtualia erdve – papildomi jutikliai, leidžiantys sekti rankų poziciją erdvėje, haptiniai įrenginiai (leidžia „jausti“ liečiamus virtualius objektus). Šios įmonės misija – padaryti, kad skaitmeninis pasaulis būtų artimesnis žmogui [4]. Vieni populiariausių rankų sekimo įrenginių yra jų kuriami „Leap Motion Controller“ jutikliai. Tai papildomi įrenginiai, naudojančios infraraudonuosius spindulius rankų aptikimui ir sekimui. 2023 m. vasarą išleista antra versija turi didesnę raišką ir matymo kampą, 25% mažesnes energijos sąnaudas ir 30% mažesnę dydį lyginant su pirmos versijos platforma [5]. Antros versijos platforma taip pat leidžia sistemą naudoti ne tik stacionariai, bet ir dėvint kartu su įvairių kitų sistemų VR akiniais kaip papildomą sekimo įrenginį. Taip pat kuriami virtualiosios realybės akiniai su integruotu rankų sekimu – „Gemini“.

„Pimax“ – technologijų įmonė, kurianti virtualiosios realybės aparatinę įrangą. Ši įmonė pateikia išskirtinį sprendimą rankų sekimui – sukurtas įrenginys, kurį galima prijungti prie virtualiosios realybės akinių kaip papildomą modulį naudojant USB C tipo jungtį [6]. Programinės įrangos kūrimui naudojamas „Leap Motion“ SDK.

„Meta“ – JAV tarptautinis technologijų konglomeratas, įkurtas Menlo Parke, Kalifornijoje. „Meta“ pradėjo kurti virtualiosios realybės įrenginius ir programinę įrangą, kai susijungė su „Oculus“ įmone. Nuo tada virtuali realybė taip viena pagrindinių „Meta“ platformos iniciatyvų [7]. Virtualiosios realybės srityje „Meta“ labiausiai pasižymi „Meta Quest“ produktų linija – tai vieni populiariausių VR įrenginių. Remiantis „Steam“ platformos 2023 m. gruodžio naudotojų aparatinės įrangos duomenimis, 37,87% naudotojų turi „Oculus Quest 2“ sistemas, 18,17% naudotojų turi „Valve Index HMD“ įrenginius ir 9,84% naudoja „Oculus Rift S“ virtualiosios realybės sistemas [8]. „Meta Quest 2“ ir „Meta Quest 3“ akiniai palaiko rankų sekimą ir paprastų gestų aptikimą. Sekimas vykdomas naudojant akinių kameras, leidžiančias nustatyti rankų padėtį ir orientaciją bei pirštų konfigūraciją erdvėje. Kai rankos aptinkamos, kompiuterinio matymo algoritmai yra naudojami rankų judėjimui ir orientacijai sekti [9].

VR akiniuose esantys sensoriai ir kameros yra naudojami stebėjimo, aplinkos pozicijos ir orientacijos sekimui. Tačiau, dažnu atveju tiesioginė prieiga prie neapdorotų duomenų (pavyzdžiui, jutikliais užfiksuoto vaizdo) programinės įrangos kūrėjams yra draudžiama. Prieiga neleidžiama dėl privatumo politikų ir naudotojų saugumo užtikrinimo. Programinės įrangos kūrėjams pateikiami programinės įrangos kūrimo rinkiniai arba kita aplikacijų programavimo sąsaja (angl. *API*) leidžiantys pasiekti tik jau apdorotą informaciją.

1.1.2. Žaidimo kūrimo įrankių analizė

Kuriant rankų sąnarių prognozių sistemą naudojami žaidimų kūrimo įrankiai, sudaryti iš žaidimų variklio ir integruotos kūrimo aplinkos. Analizuojami ir lyginami „Unity“ ir „Unreal Engine“ įrankiai (žr. 1.2 lentelė).

1.2 lentelė. Žaidimų kūrimo įrankių palyginimas

Lyginimo kriterijai	„Unity“	„Unreal Engine“
Kaina	Nemokamas (reikalinga profesionali licencija jei pajamos viršija 200000 USD per metus ³)	Nemokamas (taikomas 5% mokestis nuo pajamų, kai produkto pajamos viršija 1000000 USD)
Pirminio kodo (<i>source code</i>) prieiga	Su nemokama licencija nėra, tačiau kai kurie moduliai yra atviro kodo	Pilna prieiga per „GitHub“ talpyklą
Programavimo kalba	C#	C++
Išteklių kiekis bibliotekoje	92000+	40000+
VR palaikymas	Palaikoma didžioji dalis VR įrenginių, platus „Oculus“, „Windows Mixed Reality“ įrenginių palaikymas, AR „Android“ įrenginiuose, HTML5 VR palaikymas. „OpenXR“ papildinio ir „Interaction SDK“ palaikymas	Palaikomi „Oculus“, „Windows Mixed Reality“, „HTC“, „Samsung Gear“ ir „Google VR“ įrenginiai
Dirbtinio intelekto palaikymas	„Unity Sentis“, DI ištekčiai bibliotekoje, „ML Agents“ rinkinys, „Unity Machine Learnign Agents“	„Beheavior Trees“, „Environment Query System“, „AI Perception“, „AI Debugging“

Kainos atžvilgiu abu įrankiai yra nemokami. Yra skirtumų tarp kainų planų, tačiau kuriant sistemą tai neturėtų kelti problemų. „Unreal Engine“ leidžia platesnę prieigą prie atviro kodo – tai gali palengvinti sistemų kūrimą. Tačiau „Unity“ šioje srityje daug neatsilieka – didelė dalis plačiai naudojamų modulių yra paskelbtos atviro kodo talpyklose. „Unreal Engine“ naudoja C++ programavimo kalbą, kuri gali būti greitesnė, tačiau gali kilti papildomų problemų dėl atminties valdymo ir trūkstamo šiukšlių surinkimo (angl. *garbage collection*). „Unity“ turi platesnį virtualiosios realybės įrenginių palaikymą, prieigą ir dokumentaciją prie populiariausių virtualiosios realybės įrenginių SDK. Taip pat „Unity“ turi platesnį dirbtinio intelekto sistemų pasirinkimą.

Atsižvelgus į palygintus kriterijus, „Unity“ įrankis yra pranašesnis, todėl nuspręsta naudoti šį įrankį ir žaidimų variklį, kombinacijoje su „OpenXR“ įskiepiu, kuris teikia prieigą prie rankų sekimo naudojant „Meta Quest“ įrenginius.

1.1.3. „Unity“ žaidimų kūrimo įrankio palaikomų įskiepių analizė

Lyginami „Unity“ žaidimų kūrimo įrankio palaikomi įskiepiai. Šiuo atveju analizuojami įskiepiai, susiję su virtualiosios realybės valdymu, vizualizavimu, VR sistemų palaikymu ir teikiamu funkcionalumu (žr. 1.3 lentelė).

³ Viršijus ribą gali būti taikomi papildomi mokesčiai už kiekvieną atsiųstą programinės įrangos kopiją.

1.3 lentelė. Palaikomų įskiepių palyginimas

Lyginimo kriterijai	„OpenXR“	„Oculus“	„Windows Mixed Reality“ (WMR)
Palaikomi įrenginiai	„Oculus“ (<i>Rift, Rift S, Quest, Go</i>), „HTC“ (<i>Vive, Vive Pro, Vive Focus</i>), „Windows Mixed Reality“ įrenginiai, <i>Valve Index</i> ir daugelis kitų	Tik Oculus įrenginiai (<i>Rift, Rift S, Quest, Go</i>)	„HP Reverb G2“, „Acer“ WMR įrenginiai, „ASUS“ WMR įrenginiai, „Dell Visor“, „Lenovo Explorer“, „Samsung HMD“ <i>Odyssey</i> ir kiti
Dokumentacija	Integruota „Unity“ dokumentacija, „Oculus“ / „Vive“ / „Khronos“ dokumentacija	„Oculus“ dokumentacija	„Microsoft“ dokumentacija

Įskiepiai lyginami pagal palaikomus įrenginius ir dokumentacijos kiekį. Didelis palaikomų įrenginių kiekis yra svarbus, nes leidžia kuriamą sistemą pritaikyti skirtingiems įrenginiams ir atlikti platesnį testavimą, geriau įvertinti kuriamos sistemos tikslumą. Plati dokumentacija leidžia greitai ir efektyviai kurti programinį kodą ir rasti galimus problemų sprendimus.

Sistemai kurti pasirinktas „OpenXR“ įskiepis, kadangi leidžia integraciją su dideliu kiekiu virtualiosios realybės sistemų ir turi plačiausią rinkinį dokumentacijos šaltinių. Šis įskiepis palaiko didžiąją dalį analizuotų virtualiosios realybės sistemų, kurios gali atlikti rankų sekimą be papildomų įrenginių (valdiklių). Iš analizuotų įskiepių „OpenXR“ turi plačiausią dokumentaciją. Dokumentacijoje pateikiama informacija apie galimą funkcionalumą, naudojamas klases ir tipus bei pateikiami pavyzdiniai kodo šablonai. Šią dokumentaciją galima rasti oficialioje „Unity“ platformos svetainėje arba virtualiosios realybės sistemų gamintojų (šiuo atveju „Oculus“, „Vive“ arba „Khronos“) svetainėse.

Nuspręsta naudoti „OpenXR“ 1.10.0 versiją, „XR Plugin Management“ 4.4.0 versiją ir „XR Hands“ 1.4.1 versijos paketų kombinaciją. „XR Plugin Management“ paketas suteikia galimybę lengvai valdyti XR (angl. *extended reality*) įskiepius, palengvina inicijavimą, nustatymų keitimą ir suteikia XR papildinių palaikymą. „XR Hands“ paketas apibrėžia aplikacijų programavimo sąsają, leidžiančią pasiekti rankų pirštų sąnarių sekimo duomenis iš įrenginių, palaikančių rankų sekimą. Projekto kūrimo pradžioje buvo naudojama senesnė „XR Hands“ versija, tačiau ji buvo atnaujinta, kadangi 1.4.1 versijoje pateikiamas papildoma posistemė, leidžianti atlikti statinių gestų aptikimą ir pridėjimą. Šios posistemės naudojimas leis papildomai įvertinti kuriamą gestų aptikimo sistemą.

1.2. Egzistuojančių sprendimų analizė

Šioje dalyje apžvelgiami rankų sekimo taikymo ir įvesties būdai, analizuojami egzistuojantys gestų klasifikavimo, aptikimo, sąnarių erdvinės informacijos prognozių sprendimai.

1.2.1. Rankų sekimo taikymas ir įvesties būdai

Rankų sekimas leidžia užfiksuoti žmogaus rankos sąnarius ir juos interpretuoti. Aptinkami rankų judesiai, rankos ir pirštų sąnarių pozicijos ir orientacijos. Rankų sekimo technologija taikoma įvairiose pramonės šakose, yra skirta gerinti žmogaus ir kompiuterio sąveiką ir automatizavimą.

Vienas iš labiausiai paplitusių rankų sekimo panaudojimų yra gestų atpažinimas – tai galimybę užfiksuoti tam tikrus rankos judesius ir interpretuoti juos kaip įvestį. Gestų aptikimas ir rankų sekimas taikomas ne tik pramogoms – kompiuteriniuose žaidimuose ar virtualiosios realybės aplikacijose, bet ir kitose srityse, kaip robotika, medicina ar pramonės automatizavimas. Sritys, kuriose taikomas gestų aptikimas ir rankų sekimas [10, 11]:

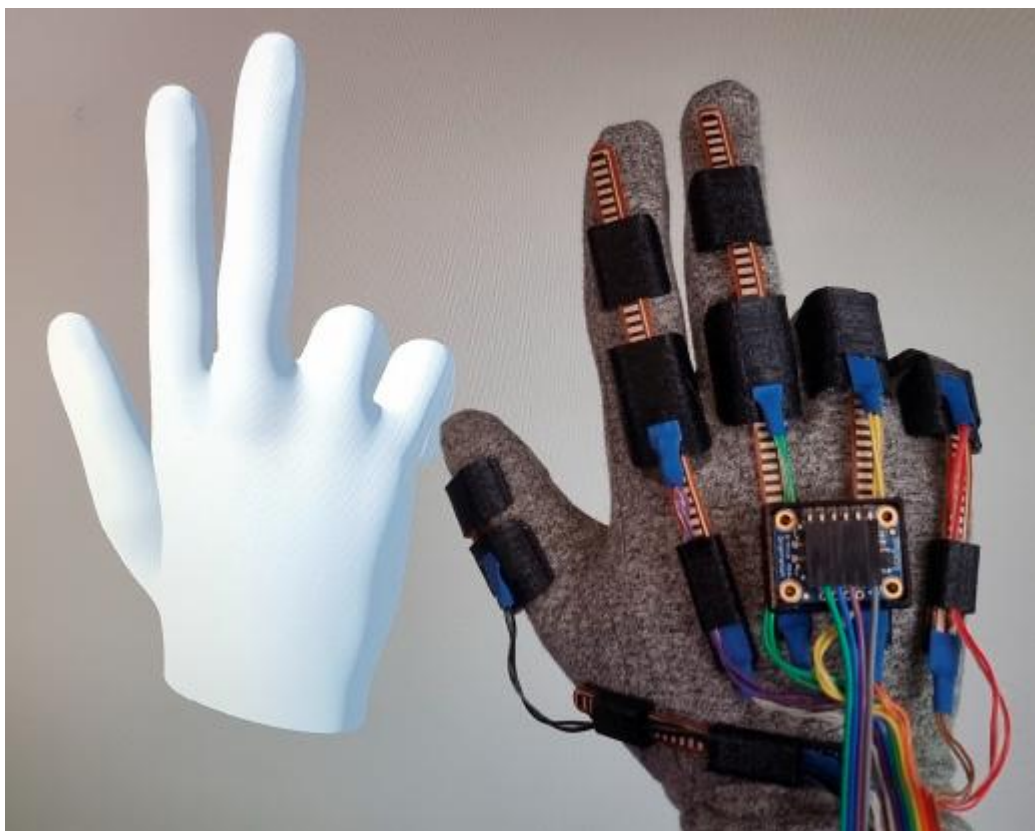
- Žmogaus ir kompiuterio sąveika (angl. *human-computer interaction, HCI*) – rankų sekimas leidžia sąveikauti su kompiuteriais ir skaitmeninėmis sąsajomis, nenaudojant tradicinių įvesties įrenginių, tokių kaip klaviatūros, kompiuterinės pelės arba kiti valdikliai. Naudotojai gali naršyti programose ir valdyti įvairias sistemas naudojant tik rankų judesius, todėl sąsajos tampa intuityvesnės ir labiau įtraukiančios.
- Žaidimai ir virtualioji realybė – naudojama siekiant padidinti įsitraukimą sąveikaujant su virtualiomis aplinkomis. Žaidėjai gali laisvai judinti rankas, į rankas imti virtualius objektus arba atlikti aptinkamus gestus.
- Robotika ir pramoninis automatizavimas – naudojama siekiant pagerinti robotų valdymą ir teleoperavimą, leidžia mašinoms atlikti realistiškesnius judesius. Taikoma gamyboje, kosmoso tyrinėjimo misijose, sąveikoje su pavojingomis aplinkomis nuotoliu, egzoskeletų ir pagalbinių robotų valdyme.
- Komunikacija ir gestų kalbos atpažinimas – kuriamos sistemos leidžia realiu laiku versti gestų kalbą į tekstą arba garsą. Tai pagerina prieinamumą klausos negalią turintiems asmenims.
- Automobilių pramonė – šiuolaikinėse transporto priemonėse vis dažniau naudojami gestais pagrįsti valdymo įrenginiai, leidžiantys valdyti garso sistemas, navigacijos sistemas ar atsilipti į skambučius, taip sumažinant dėmesio išblaškymą vairuojant ir padidinant saugumą.
- Sveikatos priežiūra ir rehabilitacija – taikoma fizioterapijoje ir rehabilitacijoje po traumų, leidžia tiksliai stebėti rankų judesius atliekant pratimus. Pacientai, sveikstantys po insulto ar kitų traumų gali naudotis VR pagrįstomis rehabilitacijos sistemomis, kurios realiu laiku teikia grįžtamąją ryšį apie atliekamus pratimus, siekiant pagerinti judrumą ir koordinaciją.
- Medicina – chirurgijoje rankų sekimas leidžia be kontakto naršyti medicininius vaizdus (pvz. magnetinio rezonanso ar kompiuterinės tomografijos tyrimus) fiziškai neliečiant įrenginių, užtikrinant papildomą sterilumą.
- Išmaniųjų namų automatizavimas – galimybė gestais valdyti integruotas sistemas. Galima valdyti šviesas, temperatūrą ar multimediją.
- Filmų kūrimas ir animacija – naudojama užfiksuojant judesius animacijos srityje, užtikrina realistišką rankos ar kitų kūno dalių judesių aptikimą. Naudojama kartu su viso kūno judesių fiksavimo sistemomis, siekiant padidinti tikroviškumą kuriant kompiuteriu sugeneruotą vaizdą [12].

Pagal įvesties tipą rankų sekimą ir gestų aptikimą galima skaidyti į dvi grupes: jutikliais (dėvimais įrenginiais) paremta įvestis ir kompiuterine rega pagrįsta įvestis.

1.2.1.1. Jutikliais pagrįstas rankų sekimas

Jutikliais paremtas rankų sekimas naudoja dėvimą įrangą, kuri tiesiogiai fiksuoja rankos judesius ir sąnarių orientaciją naudojant integruotus jutiklius. Tokio tipo įrenginiuose įprastai naudojami inerciniai matavimo vienetai, elektromagnetiniai jutikliai arba jėgai jautrūs rezistoriai, leidžiantys aptikti kampinį greitį, pagreitį ir sąnarių poziciją bei orientaciją realiu laiku.

Prie jutikliais pagrįstos įvesties galima priskirti pirštines su jutikliais ir specializuotus virtualiosios realybės sistemų valdiklius (haptines pirštines). Pirštinėmis paremtas rankų sekimas yra ganėtinai tikslus, kadangi jutikliai yra arti kiekvieno sąnario. Tokio tipo pirštinės naudojamos gestų kalbos vertimui realiu laiku [13], rehabilitacijoje [14] ir robotų pagalbininkų valdyme [15].



1.1 pav. Sensoriais pagrįstos pirštinės pavyzdys [16]

Panašaus tipo pirštinės kaip įvestis naudojamos ir virtualiosios realybės sistemose. Šios pirštinės turi daviklius, leidžiančius sekti pirštų ir rankų pozicijas ir orientaciją. Taip pat dažnu atveju suteikiamos haptinės galimybės – papildomi jutikliai ir įrenginiai gali simuliuoti pasipriešinimą suspaudžiant pirštus, suteikti prisilietimo pojūčius sąveikaujant su objektais virtualiojoje aplinkoje. Remiantis apžvalga, populiariausi šių pirštinių gamintojai yra „Meta“, „Manus“, „VRgluv“, „Haptx“, pastebima, kad bendra haptinių pirštinių rinkos dalis 2023-2030 metų laikotarpiu turėtų augti [17]. Tokio tipo pirštinės tampa vis populiareesnės, taikomos naujos technologijos, išgaunamas didesnis sekimo tikslumas. Šie įrenginiai naudojami tyrimų, simuliacinio treniravimo ir programinės įrangos kūrimo srityse. Tai naudingas įrankis automobilių pramonės, gamybos, kasybos, medicinos srityse kaip treniravimo įrankis, leidžiantis mokytis specialistus be galimų aplinkos pavojų [18]. Keletas tokių pirštinių pavyzdžių pateikiama toliau (žr. 1.2 pav.).



1.2 pav. Haptinių pirštinių pavyzdžiai („SenseGlove Nova“, „Manus Prime X Haptic VR“, „HaptX“, „TeslaSuit“) [19]

Dėvimais jutikliais paremtas rankos judesių sekimas yra tikslus, greitas ir nenutrunkantis dėl prastų apšvietimo sąlygų ar uždengimų. Tačiau, šis įvesties tipas taip pat turi minusų – didelė kaina (svyruoja tarp 250-6000 USD), įrenginiai gali būti nepatogūs, reikalauja perkalibravimo dažno naudojimo atveju. Šio tipo sekimas yra nepraktiškas sistemoms, reikalaujančioms natūralios, nekrintančios į akis sąveikos, nes vartotojai turi dėvėti papildomas pirštines ar valdiklius ant rankų.

1.2.1.2. Kompiuterine rega pagrįstas rankų sekimas

Kompiuterine rega pagrįstas rankų sekimas nereikalauja papildomos dėvimos techninės įrangos, nes rankos sąnarių padėtis ir orientacija įvertinama naudojant kameras ir kompiuterinės regos algoritmus. Šis sekimas dažniausiai įgyvendinamas naudojant paprastas RŽM (raudona, žalia, mėlyna) kameras, gylio jutiklius arba infraraudonųjų spindulių sistemas. RŽM kameromis paremtame rankų sekime judesiai nustatomi išskiriant spalvos, formos ir tekstūros požymius. Gylio jutikliai leidžia padidinti sekimo tikslumą pridedant papildomą trimatę informaciją.

Kompiuterinė rega plačiai naudojama rankų sekime ir gestų aptikime, kadangi dauguma atveju nėra reikalinga brangi papildoma įranga, žmogaus ir kompiuterio sąsaja yra bekontaktė. Tačiau šis būdas turi problemų: aplinkos apšvietimas, fone esantys objektai gali apsunkinti rankų ir pirštų sąnarių aptikimą [11]. Su šiomis problemomis susiduria ir rankų sekimas naudojant virtualiosios realybės akinius. Rankų sekimą naudojant kompiuterinę regą galima išskaidyti į atskirus tipus: aptikimas pagal spalvą, remiantis išvaizda (aptinkamos tam tikros ypatybės kadre), judesio aptikimas, skeletu paremtas aptikimas, gyliu paremtas, trimačiu rankos modeliu paremtas, dirbtiniu intelektu paremtas ir šių tipų kombinacijos.

Aptikimas pagal spalvą gali būti vykdomas naudojant papildomus skirtingų spalvų taškus ant rankų – galima naudoti pirštines su skirtingai nuspalvintais regionais. Išgaunant spalvą iš vaizdo kameros kadro galima paprastai ir pigiai (skaičiavimų prasme) rasti poziciją kadre, tačiau šis metodas nėra pakankamai tikslus. Kitas būdas aptikti rankas pagal spalvą yra naudoti odos spalvą. Tokiu atveju

nereikia dėvėti papildomų žymeklių, tačiau gali kilti problemų kai fono spalva yra panaši į odos spalvą, dėl prasto apšvietimo sąlygų arba gali būti sunku atskirti odos tekstūrą. Daugumoje mokslinių darbų, kuriuose naudojamas šis metodas pastebimos problemos: sunku aptikti odą pagal spalvą, kai fonas yra margas, duomenys yra iškreipiami kai objektai fone yra panašios į odą spalvos, reikia papildomų metodų pirštų galų ar sąnarių aptikimui, metodai sutrinka kai matomos kelios rankos ar žmonių veidai tame pačiame kadre [20, 21, 22].

Aptikimas pagal išvaizdą (angl. *appearance-based recognition*) – metodas kai iš vaizdo kadro išgaunamos tam tikros ypatybės, leidžiančios aptikti objektus kadre. Naudojant šį metodą nevykdoma atskira segmentacija, naudojamas visas kadras, bandoma išgauti ypatybes, algoritmą galima vykdyti realiu laiku. Kadangi nesiremiam spalva, išvengiama problemų dėl galimos skirtingos odos spalvos. Šiuos metodus galima naudoti kaip pagalbinius – išgauti rankos regioną naudojant spalvą, tikrinti ar tai ranka bandant aptikti pirštų, sąnarių kiekį. Taip pat galima aptikti ypatybes filmuojant tuščią kadrą be rankos. Taip išgaunamos fono ypatybės. Bandant aptikti ranką galima atlikti invertuotą veiksmą – bandyti aptikti ne pačią ranką, o pašalinti foną iš turimo kadro. Taip gaunamas likęs regionas, kuriame galimai yra ranka.

Aptikimas pagal judesį – metodas, kai naudojami keli kadrai, einantys iš eilės vaizdo medžiagoje. Leidžia atlikti tikslesnį sekimą, kadangi pašalinamas triukšmas dėl daugelio kadro panaudojimo. Metodo problemos: sutrinka aptikimas kai fonas yra dinamiškas (juda), padidėjusi delsa (kadangi naudojami keli paskutiniai kadrai), sunku aptikti stacionarius gestus.

Skeletu paremtas aptikimas – tai toks aptikimas, kai sekimui naudojama jau žinoma skeleto informacija, pasinaudojant žmogaus anatomija. Naudojama sąnarių pozicija, orientacija, kampas tarp sąnarių, atstumas, sąnario greitis, lyginant su kitu sąnariu.

Aptikimas naudojant gylio informaciją – kompiuterinės regos metodas, kai naudojant papildomą įrenginį išgaunama aplinkos gylio informacija. Įprastos kameros dažniausiai neturi tinkamų sensorių šiai informacijai išgauti iš aplinkos, todėl reikalinga papildoma įranga. Vieni populiariausių įrenginių – „Leap Motion Controllers“. Šis metodas išvengia problemų dėl apšvietimo, rankos spalvos ar fono spalvos.

Dirbtiniu intelekto algoritmais paremtas aptikimas – naudojami neuroniniai tinklai siekiant užtikrinti didesnę sekimo tikslumą kombinuojant su kitais metodais. Ši technika, naudojama daugelyje šiuolaikinių sistemų. Pagrindiniai sunkumai taikant šį metodą: tinkamo algoritmo pasirinkimas, duomenų rinkinio sudarymas, padidėjęs skaičiavimų skaičius, sulėtėjusi greitis.

1.2.2. Gestų klasifikavimas ir aptikimui taikomi dirbtinio intelekto metodai

Gestų aptikimas ir klasifikavimas – vienas iš pagrindinių rankų sekimo pritaikymo būdų. Gestų aptikimas yra svarbus, kadangi tam tikros judesių kombinacijos gali būti naudojamos įvairiose sistemose kaip įvestis. Gestų aptikimas yra susijęs su rankų sekimu, kadangi norint prognozuoti gestus svarbu tiksliai nustatyti pirštų sąnarių pozicijas ir pasukimą. Atliekama gestų aptikimo ir klasifikavimo metodų analizė bandant išsiaiškinti tinkamiausius metodus ir daugiau sužinoti apie taikomą papildomą duomenų apdorojimą.

Remiantis Yango Li et al. darbu, gestų klasifikavimą galima skaidyti į keturias grupes [23]: klasifikavimas pagal erdvinius ir laiko modelius (angl. *spatiotemporal patterns*), klasifikavimas pagal

gestų semantiką, klasifikavimas pagal sąveiką ir klasifikavimas pagal sąveikos apimtį. Pagal erdvės ir laiko veikimo būseną gestai skirstomi į du tipus: statiniai ir dinaminiai. Statiniai gestai apima fiksuotą erdvinę piršto, delno ar rankos būseną tam tikru laiko momentu, vaizduojančią vieną gestą. Dinaminis gestas apima piršto, delno ar rankos laikysenos pokyčius per tam tikrą laiką, t. y. įtraukiama laiko dimensija.

Dinaminiai gestai gali susidaryti iš statinių gestų kombinacijos – mojavimo gestas yra sudarytas iš statinio išplėsto delno gesto, kai ranka judinama į šonus. Remiantis elgesio ir kognityvine psichologija pagal Pavlocvic et al. [24], gestai pagal semantiką gali būti tyčiniai ir netyčiniai. Netyčiniai gestai neperteikia informacijos. Taip pat pagal semantiką gestus galima skirstyti į manipuliuojančius ir komunikuojančius gestus – priklausomai nuo to, ar gestas turi tam tikros informacijos, kuri gali būti panaudojama komunikuojant. Informaciją komunikuojantys gestai dažnai vykdomi nepastebint, įvyksta kalbant tarp asmenų. Pagal sąveiką gestai skirstomi gestus susiejant su įvestimi. Pagal sąveikos būdą gestai gali būti skirstomi į potėpio (angl. *stroke*) ir bekontakčius mostus ore.

Gestų aptikimas vykdomas naudojant įvairius dirbtinio intelekto algoritmus ir metodus. Algoritmo pasirinkimas priklauso nuo konkrečių reikalavimų – gestų tipo (statinis ar dinaminis), įvesties duomenų (gylis informacija, RŽM informacija, tiesioginės sąnarių koordinatės iš kitų sensorių) ir norimo pasiekti tikslumo. Neuroniniai tinklai gestų aptikimo srityje yra ypač populiarūs, nes gali efektyviai aptikti gestus, ypač kai naudojama kompiuterinė rega ar naudojami nuoseklūs duomenys. Naudojami kelių tipų neuroniniai tinklai: konvoliuciniai neuroniniai tinklai (angl. *convolutional neural networks, CNN*), rekurentiniai neurotiniai tinklai (RNN), ilgalaikės trumpalaikės atminties tinklai (LSTM), vartais paremtų rekurentinių vienetų tinklai (GRU). Neuroniniai tinklai ir jų variacijos tinka norint apdoroti vaizdo medžiagos duomenų kadrus ar gylis informacijos kadrus, leidžia nustatyti ir išmokyti reikšmingas erdvines ypatybes. Taip pat gestų klasifikavimui tinka paslėpti Markovo modeliai (angl. *hidden Markov model, HMM*), sprendimų medžiai ir miškai, atraminių vektorių klasifikatoriai (angl. *support vector machines, SVM*). Siekiant didesnio tikslumo galima naudoti ansamblio metodus – kelių skirtingų dirbtinio intelekto metodų kombinaciją, skiriant svorinius koeficientus kiekvienam modeliui.

Vienas iš dažnai naudojamų modelių objektų aptikimui yra „YOLOv3“ (arba kitos, naujesnės variacijos). Šis metodas objekto aptikimą traktuoja kaip regresijos uždavinį, numatydamas klasių tikimybes ir ribojančių langų (angl. *bounding-box*) poslinkius paveikslėlių pereinant naudojant vieną į priekį nukreiptą (angl. *feed-forward*) konvoliucinį neuroninį tinklą [25]. Naudojant šį metodą išvengiama atskiros regionų pasiūlymų generavimo ir pakartotinio mėginių paėmimo (angl. *resampling*) – visi metodai apjungiami į vieną tinklą. Tai užtikrina greitą algoritmo veikimą, leidžiantį atlikti objektų aptikimą realiu laiku. Moksliniame straipsnyje apie realaus laiko rankų gestų aptikimo, paremto giluminiu mokymusi naudojant „YOLOv3“ modelį [20] teigiama, kad šis modelis taip pat tinka gestų aptikimui kai pateikiami kadrai su antraštėmis. Lyginant su „VGG16“ [26], „SGD“ [27], „SSD“ [28], naudojamas „YoloV3“ modelis turi didžiausią tikslumą (angl. *precision* ir *accuracy*), atšaukimą (angl. *recall*) ir F-1 įvertį. Pateiktas metodas gali būti toliau patobulintas aptinkant keletą gestų vienu metu, kadangi aprašyti bandymai pateikiami aptinkant tik vieną gestą vienu metu.

Osamo Mazharo et al. straipsnyje apie giliojo mokymosi karkasus, skirtus statiniams ir dinaminiais gestams aptikti [29] naudojama kompiuterinė rega su RŽM kamera be gylis informacijos, kartu su pasiūlytu „StaDNet“ konvoliuciniu neuroniniu tinklu ir LSTM modeliu. Siekiant sumažinti netikslumą, kylantį dėl skalės trūkumo naudojant įprastą RŽM kamerą, papildomai naudojamas

„Kinect V2“ įrenginys. Tai leidžia patobulinti gylio nustatymą, kadangi modelio gylio aptikimas yra išreniruotas pagal bendrą tiesą (angl. *ground truth*), gautą iš įrašų sekų. Tai leidžia išvengti problemų, kai netinkamai aptinkami gestai ar kūno dalys dėl skirtingų aptinkamų galūnių dydžio. Kadangi darbe orientuojamasi į žmogaus ir robotų sąveiką ir pritaikymą pramonėje, galimybė naudoti įprastas, pigias kameras be gylio informacijos yra ypač svarbu. Šiame straipsnyje apžvelgiamos dviejų tipų įvestys: kai naudojama papildoma dėvima įranga (jutikliai) ir bekontakčiai metodai. Padaryta išvada, kad papildomos dėvimos sistemos yra nepatogios, brangios, reikalauja papildomų laiko resursų pritaikymui [29, 30], todėl plačiau apžvelgiama tik kompiuterinė rega. Taip pat darbe palyginami įprasti konvoliuciniai neuroniniai tinklai, trimačiai konvoliuciniai neuroniniai tinklai ir LSTM modeliai, išbandyti moksliniuose darbuose apie žmogaus veiksmų aptikimą ir vaizdo įrašų klasifikavimą [31, 32]. Pastebėta, kad sujungtų kadrų architektūros tikslumo atžvilgiu veikia panašiai kaip ir vieną kadrą naudojančiuose modeliuose. Aprašyta atliekama ypatybių inžinerija (angl. *feature engineering*) – naudojami ne tik tiesiogiai išgaunamos sąnarių koordinatės, bet ir išvestinė informacija iš 8 viršutinės kūno dalies sąnarių: greitis, pagreitis [29]. Naudojama konvoliucinio neuroninio tinklo ir LSTM modelio kombinacija su išmetimo (angl. *dropout*) sluoksniais ir „SoftMax“ aktyvacijos funkcija. Kai atliekama statinių gestų klasifikacija, naudojamas tik konvoliucinis neuroninis tinklas su aktyvacijos funkcija kairei ir dešinei rankai. Jei gestas yra dinaminis, naudojama abiejų rankų kombinacija ir LSTM. Naudojant „Praxis“ gestų duomenų rinkinį ir lyginant su kitu panašiu gestų aptikimo įgyvendinimu [33] (kai naudojamas panašus CNN-LSTM modelių tinklas) gaunamas 23% didesnis tikslumas („StaDNet“ tikslumas 99,6%, kito panašaus tinklo tik 76.61%). Naudojamas „OpenPose“ [34] skeleto išgavimo įrankis. Atsižvelgiama į anksčiau aprašytas problemas kai naudojama įprasta kamera – reikalingas tinkamas apšvietimas, minima fono spalvų įtaka.

Noorkholiso Luthfilo Hakimo et al. straipsnyje apie dinaminių rankos gestų aptikimą [35] aprašomas RŽM ir gylio kameromis surinkto duomenų rinkinio naudojimas, trimačio konvoliucinio neuronų tinklo (3DCNN) ir LSTM modelių naudojimas gestų aptikimui atlikti. Klasifikavimo proceso pabaigoje naudojama baigtinių būsenų mašina (angl. *finite state machine, FSM*). Tai leidžia padidinti aptikimo tikslumą nuo 89% iki 91%. Moksliniame straipsnyje aprašomi metodai leidžia atlikti gestų aptikimą realiu laiku – tai yra reikalinga, kadangi darbe siekiama įgyvendinti išmanių televizorių valdymą gyvai naudojant gestus. Sistema leidžia aptikti 24 gestus – 13 statinių ir 11 dinaminių. Duomenys yra apdorojami – iš RŽM kameros duomenų pašalinami regionai, kurie yra per toli remiantis gylio kamera. 3DCNN gali išgauti trumpalaikes laiko ypatybes iš duomenų sekų, tačiau duomenys yra trumpalaikiai, kyla problemų įsimeriant ilgesnes sekas – vien 3DCNN modelio neužtenka. Kadangi dinaminiai gestai užtrunka 32-50 kadrų, naudojamas papildomas LSTM tinklas. FSM naudojimas leidžia išsaugoti kontekstą apie paskutinius naudotus veiksmus ir dabartinę būseną. Naudojant minėtų modelių kombinaciją gaunamas 97,6% tikslumas. Tačiau atliekant testavimą realiu laiku tikslumas svyruoja apie 90% – sistemą testuojantys asmenys buvo apmokomi tik 5 minutes, jų teigimu nebuvo užtikrinta, kokie gestai yra tinkami sistemos naudojimui.

1.2.3. Sąnarių pozicijos ir orientacijos prognozės

Sąnarių trajektorijų ir erdvinės informacijos prognozavimas yra svarbus siekiant užtikrinti didesnę tikslumą ar įgyvendinti tikroviškesnį judesių nustatymą. Nors moksliniuose darbuose dažnu atveju sprendžiamos kitos užduotys: aptinkami viso kūno sąnariai, prognozuojama tik pozicija, naudojamos

įvairios įvestys (kameros ar kiti jutikliai), atliekama analizė, siekiant gauti išvalgų, kurias galima pritaikyti kuriant rankos pirštų sąnarių prognozių sistemą, kai sekimas sutrinka.

Tango et al. darbe aprašomi rekurentiniai dėmesio tinklai, leidžiantys pagerinti žmogaus sąnarių padėties prognozavimą dinamiškai sutelkiant dėmesį į svarbiausias erdvinės ir laikinės judesių sekų savybes [36]. Šiuo tyrimu siekta padidinti pozos pozicijos prognozės tikslumą integruojant savaiminio dėmesio mechanizmus į RNN pagrįstą sistemą. Lyginant su tradiciniais RNN modeliais, kurie apdoroja įvesties sekas tolygiai, RAN priskiria didesnę svorį svarbiausiems kadrams, todėl modeliai gali geriau užfiksuoti judesio priklausomybes bėgant laikui. Šis metodas leido pasiekti geresnių rezultatų nustatant žmogaus galūnių sąnarių pozicijas lyginant su kitais darbais, tačiau metodas buvo taikomas viso kūno sekimui, o ne detaliai rankos pirštų sąnarių pozicijų ir pasukimų nustatymui. Darbe pateikiami realaus laiko sekimo reikalavimai (30 kadrų per sekundę), tačiau šios greitaveikos nepakanka atliekant prognozes VR erdvėse.

Zhang et al. darbe pristatomas fizikiniais dėsniais paremtas judesio prognozės modelis, integruojantis *Eulerio-Lagrango* lygtis su rekurentinėmis neuroninių tinklų architektūromis [37]. Tyrime daugiausia dėmesio skiriama siekiant patobulinti ilgalaikį judesių prognozavimo tikslumą prijungiant fizikiniais dėsniais paremtus apribojimus, siekiant užtikrinti realistiškas žmogaus judesių dinamikas ir trajektorijas. Šis metodas pagerino stabilumą prognozuojant trajektorijas, ypač kai atliekami sudėtingi judesiai arba galūnės judinamos ypač greitai. Tyrime uždavinys sprendžiamas atliekant viso kūno sąnarių sekimą, kuriose nedideli netikslumai neturi reikšmingos įtakos prognozuojamų judesių tikroviškumui. Dėl papildomų skaičiavimų padidėja prognozavimo laikas, todėl šis metodas mažiau tinkamas VR prognozių realiu laiku atvejui.

Liu et al. atliktame tyrime nagrinėjamas hibridinis metodas, sujungiant rekurentinius neuroninius tinklus su UKF (angl. *Unscented Kalman Filter*) filtru judesių prognozėms atlikti [38]. UKF naudojamas patikslinti RNN tinklu generuojamas prognozes, pridėdamas nuoseklų būsenų įvertinimą, kuris leidžia sumažinti triukšmą judesiuose ir išlygina judesio trajektorijas. Šis metodas pagerino rezultatus, kai sekami judesiai yra nepastovūs. Tačiau, darbe aprašomos prognozės atliekamos tik viršutinės kūno dalies sąnarių prognozėms, netaikant realaus laiko greitaveikos apribojimų.

Cui et al. pasiūlė grafais pagrįstą rekurentinį modelį trimačiams žmogaus judesiams prognozuoti, naudojant visiškai sujungtus erdvės ir laiko grafus, siekiant modeliuoti priklausomybes, esančias tarp sąnarių [39]. Šis metodas veikia geriau nei tradiciniai RNN modeliai. Darbe pateikiama, kad modelis tinka prognozuojant visas kūno sąnarių pozų prognozes naudojant sporto ir fizinio aktyvumo duomenų rinkinius. Darbe neatliekamos orientacijos prognozės, prognozuojamos tik pozicijos – modelis netinka norint atlikti ir vizualizuoti prognozes virtualioje erdvėje. Taip pat grafais pagrįsti modeliais reikalauja daug didesnio skaičiavimų kiekio, sulėtėja greitaveika.

Gamate et al. pristatė hibridinį klasikinės ir regresinės kinematikos modelį, skirtą nepertraukiamam 3D rankos trajektorijos prognozavimui balistiniams judesiams virtualiosios realybės aplinkose [40]. Darbe aprašomas metodas, naudojantis kinematikos lygtis su regresijos pagrindu veikiančių koeficientų interpoliavimu, kai nenaudojami giliojo mokymosi metodai. Naudojama *OptiTrack* judesių fiksavimo sistema (naudojanti 8 kameras ir 7 žymeklius ant eksperimentų dalyvių kūnų) kartu su *Oculus Quest* virtualiosios realybės akiniais ir *Oculus Touch* valdikliais. Dalyviams pateiktos 4 virtualiosios realybės veiklos: specializuota užduotis, reikalaujanti į tam tikrą poziciją nukreipti ranką ir trys populiarius žaidimai. Nors darbe naudojama VR įranga, pagrindinis duomenų rinkinių

sudarymas vyksta naudojant *OptiTrack* sistemą, o *Oculus* įrenginiai naudojami tik kaip virtualiosios realybės sąsaja ir įrankis, skirtas patikrinti modelio generalizavimą specializuotoje užduotyje. Darbe aprašomas pasiektas 0,8 cm, 0,85 cm ir 3,15 cm RMSE (angl. *root mean square error*), 0,28 cm, 0,33 cm ir 1,97 cm MAE (angl. *mean absolute error*) atliekant prognozes 100 ms, 200 ms ir 300 ms į ateitį atitinkamai. Tačiau, darbe prognozuojamos tik rankų sąnarių pozicijos, o ne rankos sąnarių pozicijos ir orientaciją. Taip pat prognozės nėra atliekamos realiu laiku, metrikų įverčiai išgaunami atskirai, nenaudojant virtualiosios realybės sistemų fone – neišbandyta greitaveika, nėra realaus laiko prognozių apribojimų. Naudojama daug brangesnė ir tikslesnė (paklaida mažesnė nei 0.2 mm) judesių sekimo įranga, lyginant su VR sistemomis.

1.2.4. Sprendimų analizės išvados

Atlikta rankų sekimo analizė parodė, kad rankų sekimo technologija yra plačiai taikoma įvairiose srityse. Ant rankų dėvimų jutiklių (pvz. haptinių pirštinių) įvestis yra tiksli, tačiau, lyginant su kitais metodais, brangi. Kompiuterine rega pagrįstos sistemos susiduria su problemomis dėl apšvietimo arba dėl fone esančių objektų. Pastebimos galimybės kurti pigesnę sprendimą, kuriame nėra naudojami dėvimi jutikliai ir išvengiama problemų dėl apšvietimo naudojant pirštų erdvinės informacijos prognozes realiu laiku.

Atlikus gestų aptikimo ir klasifikavimo analizę pastebėta, kad statinių gestų aptikimui naudojami konvoliuciniai neuroniniai tinklai arba jų variacijos. Dinaminiams gestams aptikti ir klasifikuoti naudojami rekurentiniai neuroniniai tinklai (RNN) ir ilgalaikės trumpalaikės atminties tinklai (LSTM). Rekurentiniai neuroniniai tinklai gali geriau užfiksuoti sekas, tai leidžia atlikti tikslesnes prognozes kai judesys kinta laike. Analizuotuose darbuose pabrėžiama tikslaus duomenų rinkinio sudarymo svarba – siekiant pagerinti prognozių kokybę naudojami papildomi įrenginiai sudarant duomenų rinkinius, pvz. kameros, galinčios užfiksuoti gylio informaciją, naudojami papildomi jutikliai ant rankų arba keletas įrenginių vienu metu. Susiduriama su problemomis dėl įvesties tikslumo, kurį galima padidinti kuriant rankos pirštų sąnarių prognozių sistemą.

Atliekant sąnarių pozicijos ir orientacijos prognozių atlikimo sprendimų analizę pastebėta, kad daugelyje darbų naudojamas kompiuterine rega pagrįstas rankų sekimas, kombinuojamas su dirbtiniu intelektu paremtais algoritmais ir architektūromis, tokiomis kaip rekurentiniai neuroniniai tinklai. Analizuotuose darbuose sprendžiama problema skiriasi nuo darbe aprašomos ir sprendžiamos problemos – dažnu atveju bandoma prognozuoti tik žmogaus rankų (ar kitų galūnių) sąnarių pozicijas, o ne rankos pirštų sąnarių pozicijas ir orientaciją. Pagrindiniai skirtumai yra tai, kad kituose darbuose nėra didelio prioriteto greitaveikai ir dideliame tikslume. Greitaveikos atžvilgiu reikalavimų analizuotose darbuose nėra (arba yra minimalūs). Šiame darbe greitaveika yra ypač svarbi, kadangi sąnarių pozicijos nutrūkus sekimui turi būti aptinkamos tokiu dažniu, kaip virtualiosios realybės sistemos kuriamas vaizdas (72-120 kadrų per sekundę dažniu). 120 kadrų per sekundę dažnis atitinka 8,33 ms tarp prognozių atlikimo. „Meta“ Quest 2 įrenginio rankų sekimas įprastai vyksta 30 kadrų per sekundę dažniu. Remiantis „Meta“ [41] informacija apie greito sekimo režimą rankų sekimą, galimas 60 kadrų per sekundę dažnis. Tačiau, naudojant „XR Hands“ paketo įrankius pastebėta, kad sekimas atnaujinamas dažniau, kadangi atliekamos papildomos galimų pozicijų ir orientacijų prognozės realiu laiku. VR aparatinės įrangų kūrimo įmonės neatskleidžia prognozių modelių įgyvendinimo detalių. Skirtumas lyginant tikslumą yra tai, kad net ir su didesne paklaida bendra kūno forma išlieka panaši – pvz. 2 cm skirtumas tarp peties ir alkūnės sąnarių vizualiai nėra lengvai pastebimas. Tačiau lyginant tikslumą, kai sekamos rankos pirštų pozicijos 2 cm skirtumas tarp realios

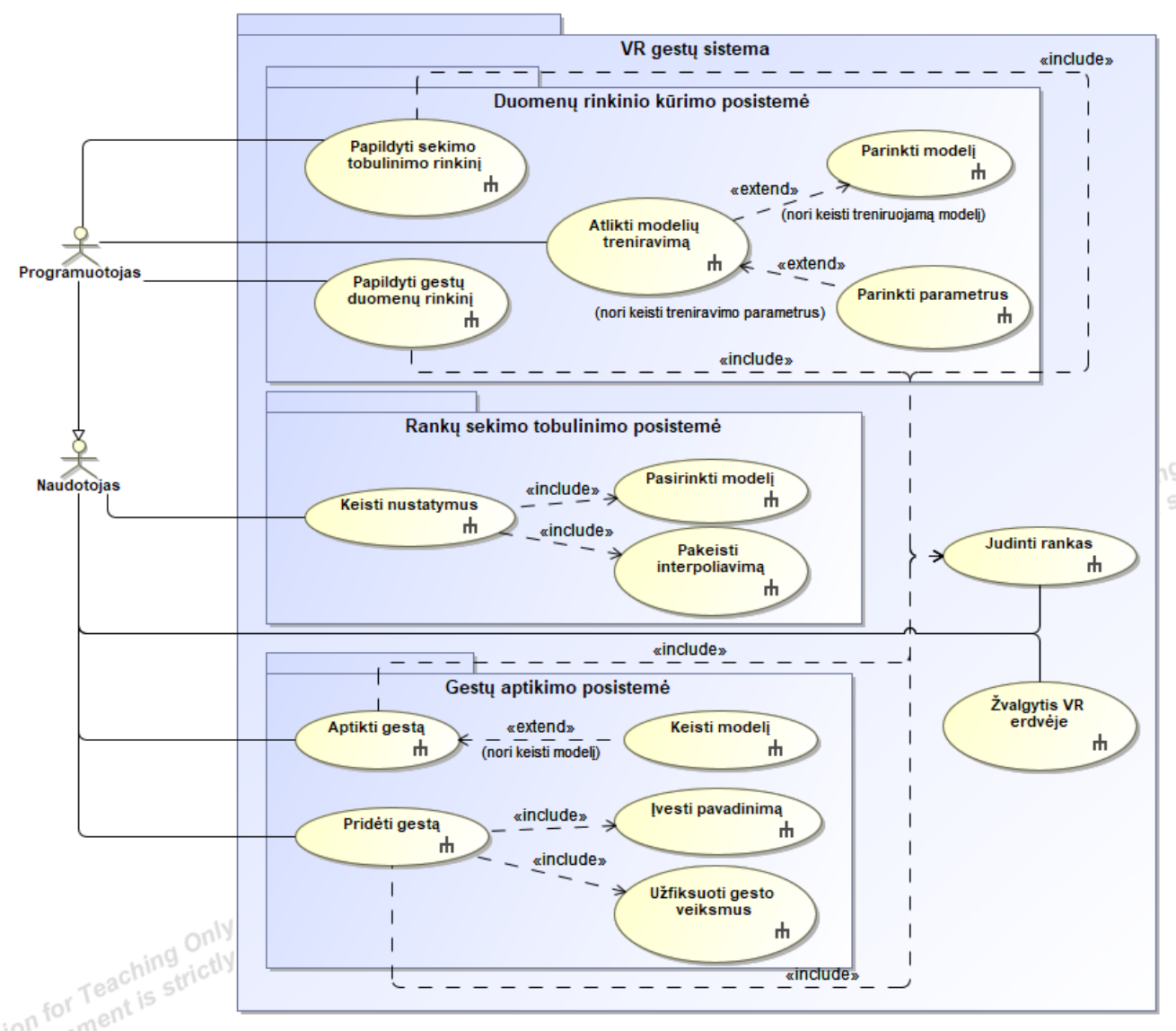
reikšmės ir prognozuotos reikšmės bus stipriai pastebimas. Atliekamas tyrimas leidžia išspręsti šias problemas.

2. Rankų sąnarių virtualiojoje realybėje prognozių sistemos projektavimas

Skyriuje pristatomas planuojamas sistemos funkcionalumas naudojant UML (angl. *unified modeling language*) diagramas. Funkciniai reikalavimai apibrėžti naudojant panaudojimo atvejų diagramas, veiklos diagramas ir lenteles. Apibrėžiami nefunkciniai reikalavimai. Pateikiama diegimo diagrama ir techninė specifikacija. Pateikiami planuojami įgyvendinti dirbtinio intelekto algoritmai ir sprendžiamai problemai pritaikytos metrikos. Apibrėžiamas testavimo planas ir kriterijai.

2.1. Sistemos funkcijos

Sistemos funkciniai reikalavimai pateikiami naudojant UML panaudojimo atvejų diagramą (žr. 2.1 pav.).



2.1 pav. Sistemos panaudojimo atvejų diagrama

Pagal funkcionalumą sistemą galima skaidyti į tris posistemas:

- duomenų rinkinio kūrimo posistemė – posistemė, atsakinga už veiksmus, susijusius su duomenų išsaugojimu, papildomų gestų modelio treniravimui pridėjimu ir modelių treniravimu;

- rankų sekimo tobulinimo posistemė – posistemė, leidžianti keisti nustatymus, susijusius su rankų sekimu;
- gestų aptikimo posistemė – posistemė, leidžianti naudotojui pridėti naujus gestus ir aptikti jau esamus gestus.

Sistemoje naudojami du aktoriai: naudotojas (šiuo atveju tai žmogus, kuris tik naudojasi įgyvendinta tikslumo tobulinimo ir gestų aptikimo sistema), programuotojas (žmogus, kuris savo kuriamoje programinėje įrangoje taiko tikslumo tobulinimo ir gestų aptikimo sistemą).

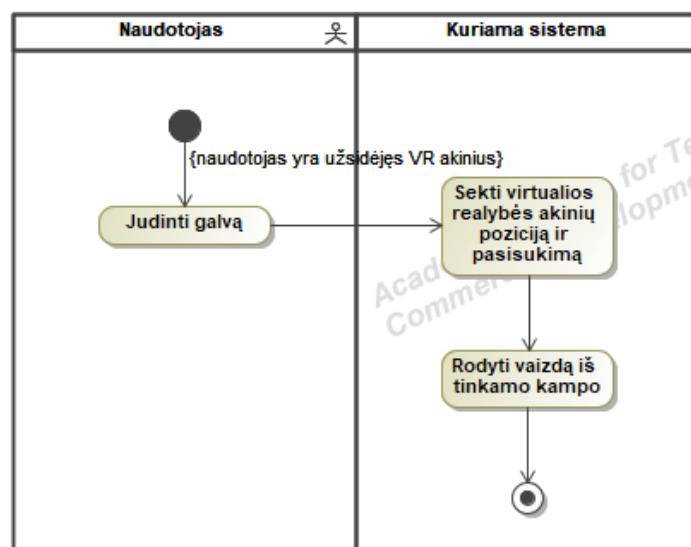
2.1.1. Bendras sistemos funkcionalumas

Sistemoje galima atlikti bendrus veiksmus – naudotojas gali žvalgytis VR erdvėje ir judinti rankas, kurios yra aptinkamos ir vizualizuojamos virtualioje erdvėje.

„Žvalgytis VR erdvėje“ panaudojimo atvejo specifikacija ir veiklos diagrama pateiktos toliau (žr. 2.1 lentelė, 2.2 pav.). Naudotojas gali judinti akinius ir žvalgytis virtualioje erdvėje.

2.1 lentelė. „Žvalgytis VR erdvėje“ specifikacija

Panaudojimo atvejis. Žvalgytis VR erdvėje		
Tikslas. Leisti naudotojui žvalgytis virtualioje erdvėje		
Aprašymas. Naudotojas turi turėti galimybę žvalgytis erdvėje		
Prieš sąlyga		Naudotojas yra užsidėjęs VR akinius
Aktoriai		Naudotojas
Susiję panaudojimo atvejai	Apimami PA	-
	Išplečiantys PA	-
	Specializuojami PA	-
Po sąlyga		Judinant akinius kinta matomas vaizdas

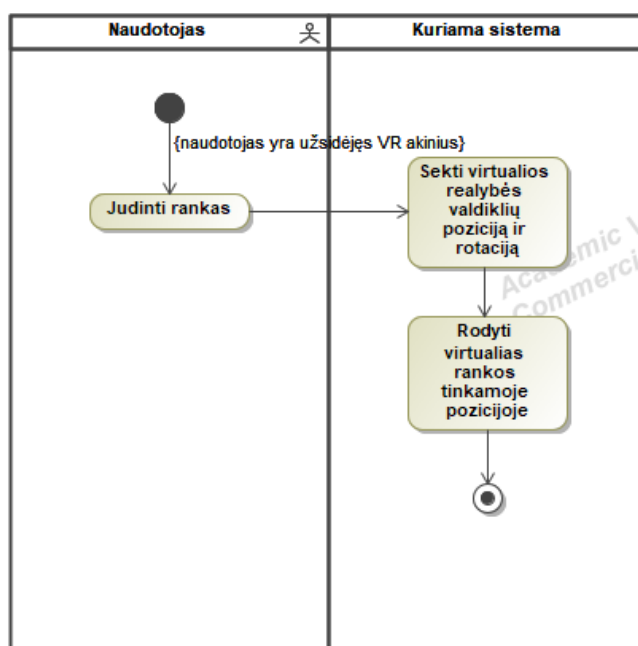


2.2 pav. „Žvalgytis VR erdvėje“ veiklos diagrama

„Judinti rankas“ panaudojimo atvejo specifikacija ir veiklos diagrama pateiktos toliau (žr. 2.2 lentelė, 2.3 pav.). Naudotojas gali judinti rankas ir matyti jas virtualioje erdvėje.

2.2 lentelė. „Judinti rankas“ specifikacija

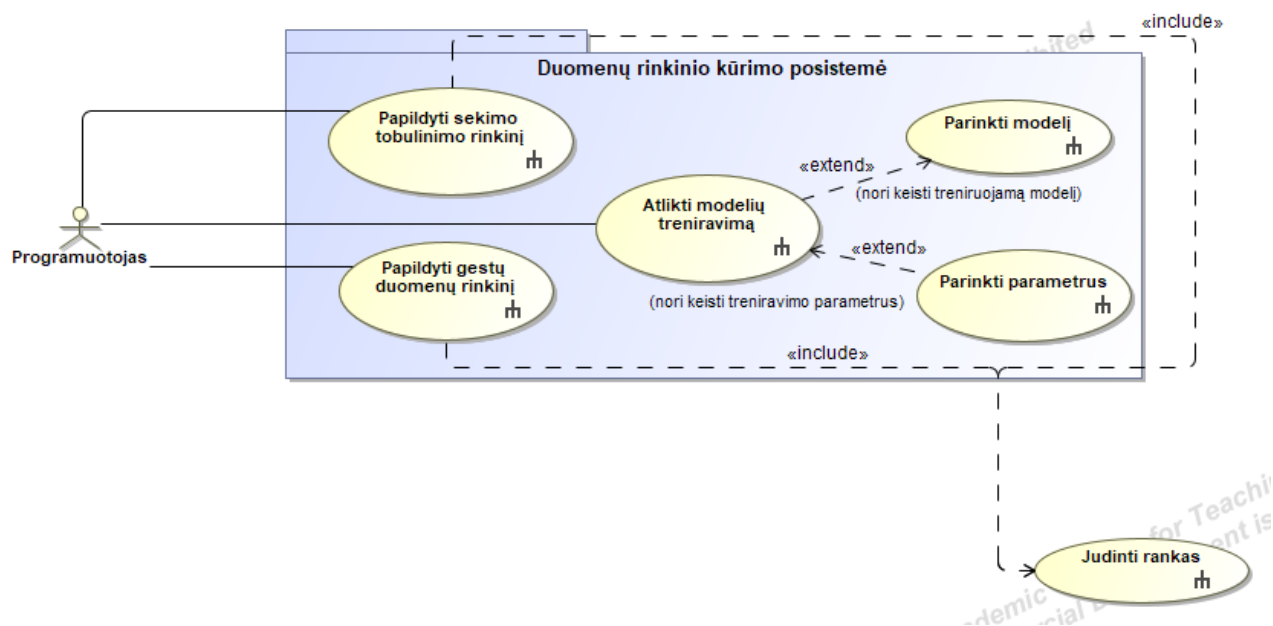
Panaudojimo atvejis. Judinti rankas		
Tikslas. Leisti naudotojui judinti rankas virtualioje erdvėje		
Aprašymas. Naudotojas turi turėti galimybę judinti rankas virtualioje erdvėje ir matyti virtualių rankų poziciją ir pasukimą		
Prieš sąlyga		Naudotojas yra užsidėjęs VR akinius
Aktoriai		Naudotojas
Susiję panaudojimo atvejai	Apimami PA	-
	Išplečiantys PA	-
	Specializuojami PA	-
Po sąlyga		Judinant rankas kinta regima virtualių rankų pozicija ir pasukimas



2.3 pav. „Judinti rankas“ veiklos diagrama

2.1.2. Duomenų rinkinio kūrimo posistemė

Ši posistemė yra atsakinga už veiksmus, susijusius su duomenų išsaugojimu, papildomų gestų modelio treniravimui pridėjimu ir modelių treniravimu. Posistemę sudaro 5 programuotojo veiksmai: sekimo tobulinimo rinkinio papildymas, gestų duomenų rinkinio papildymas ir modelių treniravimas, kurio metu papildomai galima pasirinkti treniruojamą modelį ir parametrus (žr. 2.4 pav.)

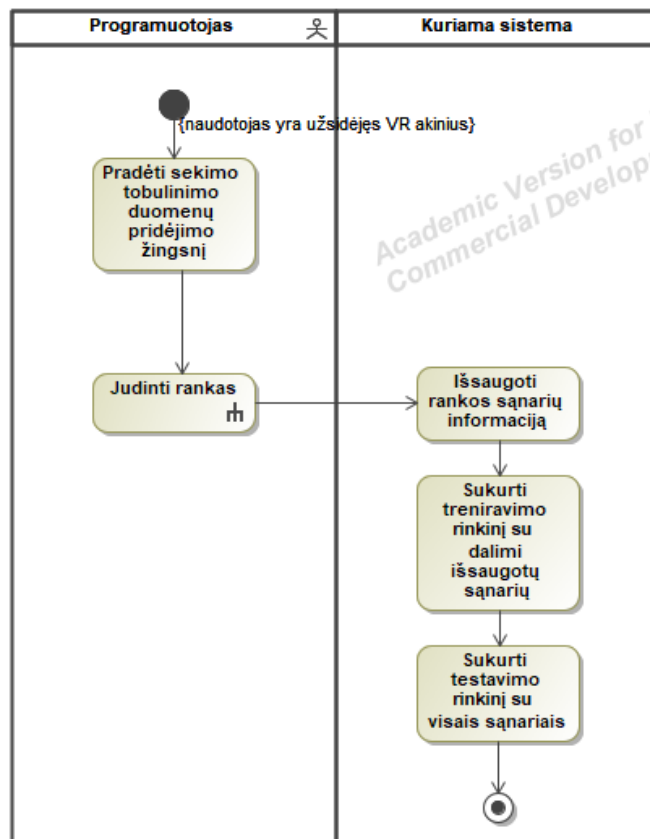


2.4 pav. Duomenų rinkinio kūrimo posistemės panaudojimo atvejų diagrama

„Papildyti sekimo tobulinimo rinkinį“ panaudojimo atvejo specifikacija ir veiklos diagrama pateiktos toliau (žr. 2.3 lentelė, 2.5 pav.). Programuotojas gali papildyti rankų sekimo tikslumo tobulinimo duomenų rinkinį. Automatiškai ištrinama dalis duomenų (treniravimo rinkiniui) ir sudaromas testavimo rinkinys iš visų sąnarių informacijos.

2.3 lentelė. „Papildyti sekimo tobulinimo rinkinį“ specifikacija

Panaudojimo atvejis. Papildyti sekimo tobulinimo rinkinį		
Tikslas. Leisti programuotojui pridėti duomenų į sekimo tobulinimo rinkinį		
Aprašymas. Programuotojas turi turėti galimybę pridėti papildomų duomenų į sekimo tobulinimo rinkinį, siekiant didinti modelių tikslumą.		
Prieš sąlyga		Naudotojas yra užsidėjęs VR akinius
Aktoriai		Programuotojas
Susiję panaudojimo atvejai	Apimami PA	Judinti rankas
	Išplečiantys PA	-
	Specializuojami PA	-
Po sąlyga		Kinta sekimo tobulinimo duomenų rinkinys

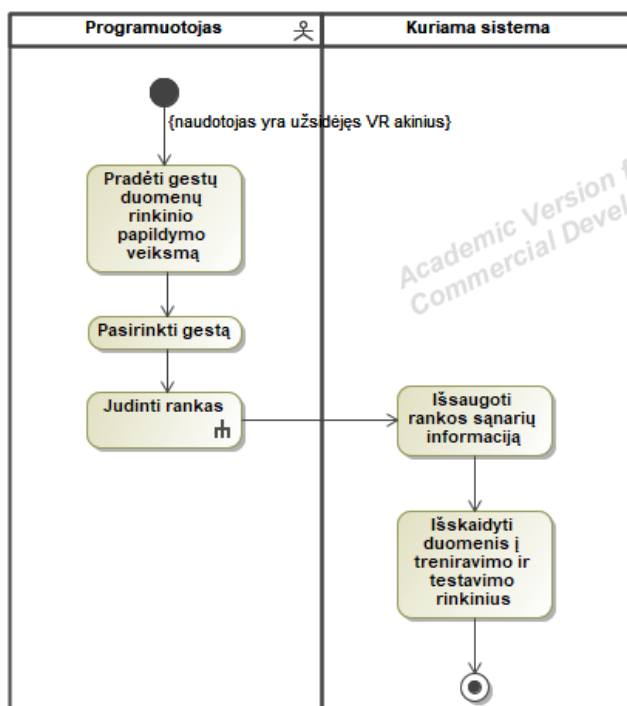


2.5 pav. „Papildyti sekimo tobulinimo rinkinį“ veiklos diagrama

„Papildyti gestų duomenų rinkinį“ panaudojimo atvejo specifikacija ir veiklos diagrama pateiktos toliau (žr. 2.4 lentelė, 2.6 pav.). Programuotojas gali papildyti esamų gestų duomenų rinkinį. Duomenų rinkinys skaidomas į testavimo ir treniravimo rinkinius.

2.4 lentelė. „Papildyti gestų duomenų rinkinį“ specifikacija

Panaudojimo atvejis. Papildyti gestų duomenų rinkinį		
Tikslas. Leisti programuotojui papildyti turimą gestų aptikimo duomenų rinkinį		
Aprašymas. Programuotojas turi galėti papildyti jau egzistuojančių gestų duomenų rinkinį. Su papildomais duomenimis užtikrinimas gestų aptikimo tikslumas.		
Prieš sąlyga		Naudotojas yra užsidėjęs VR akinius
Aktoriai		Programuotojas
Susiję panaudojimo atvejai	Apimami PA	Judinti rankas
	Išplečiantys PA	-
	Specializuojami PA	-
Po sąlyga		Papildomas gestų duomenų rinkinys

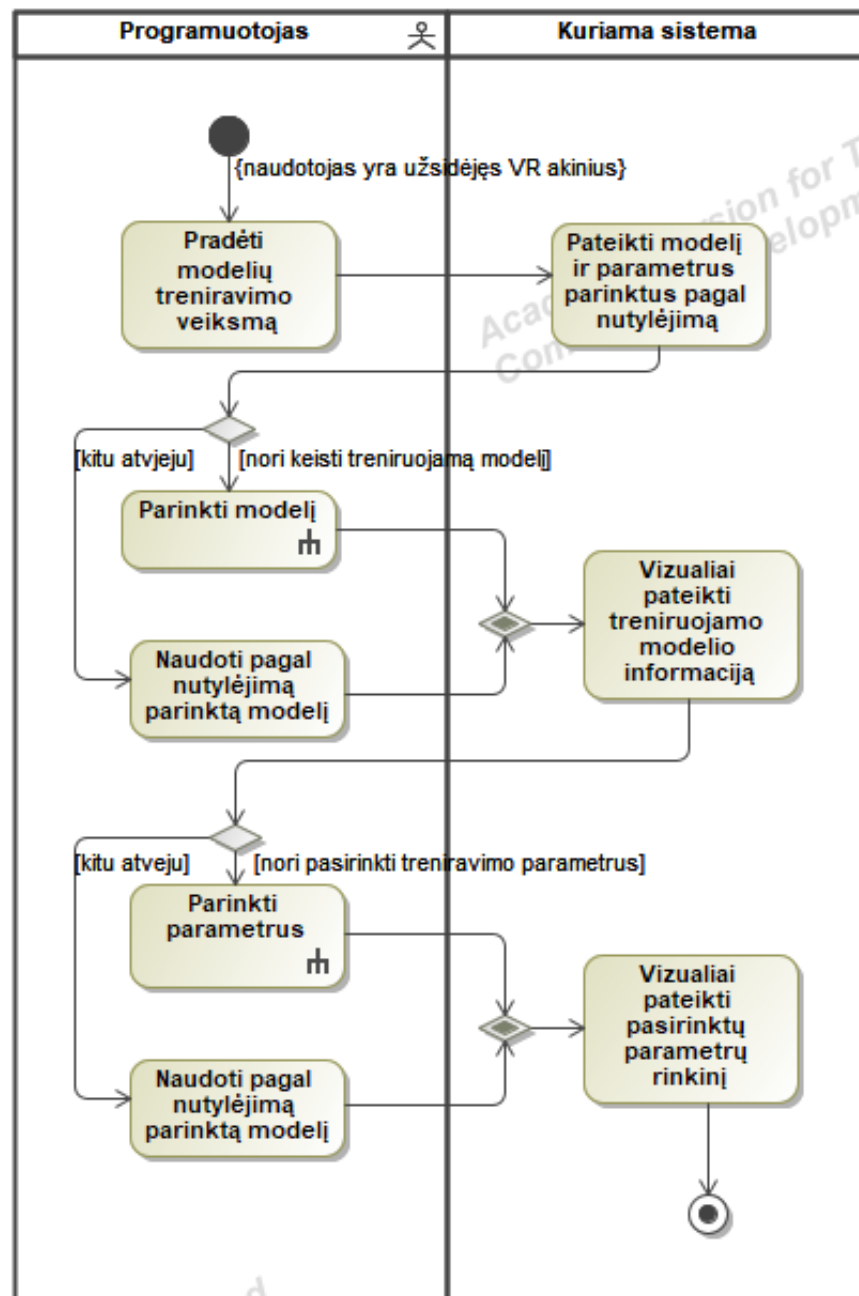


2.6 pav. „Papildyti gestų duomenų rinkinį“ veiklos diagrama

„Atlikti modelių treniravimą“ panaudojimo atvejo specifikacija ir veiklos diagrama pateiktos toliau (žr. 2.5 lentelė, 2.7 pav.). Programuotojas gali atlikti pakartotinį modelių treniravimą parinkdamas modelį ir parametrus. Papildomas treniravimas gali padidinti tikslumą.

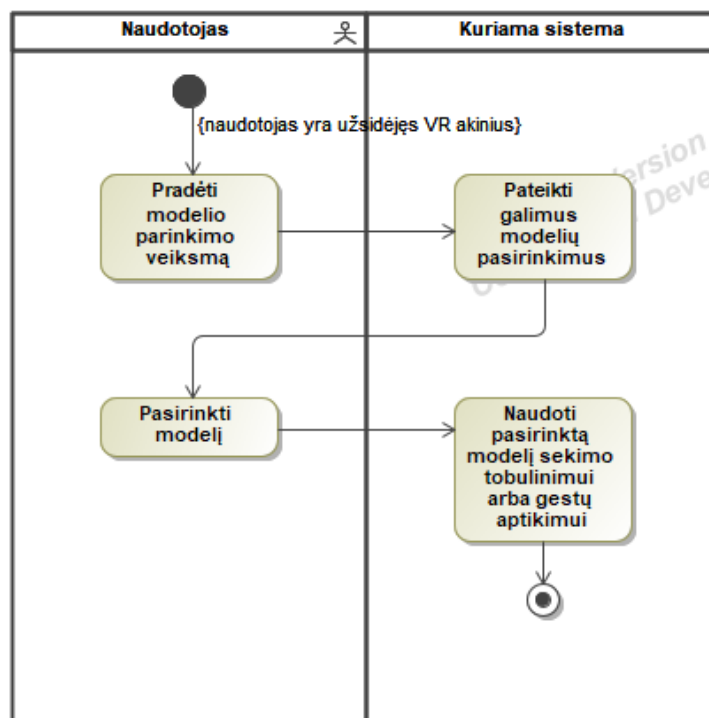
2.5 lentelė. „Atlikti modelių treniravimą“ specifikacija

Panaudojimo atvejis. Atlikti modelių treniravimą		
Tikslas. Leisti programuotojui atlikti modelių treniravimą naudojant išsaugotus duomenų rinkinius		
Aprašymas. Programuotojas turi galėti pasirinkti modelį ir treniravimo parametrus.		
Prieš sąlyga		Naudotojas yra užsidėjęs VR akinius
Aktoriai		Programuotojas
Susiję panaudojimo atvejai	Apimami PA	-
	Išplečiantys PA	Pasirinkti modelį Parinkti parametrus
	Specializuojami PA	-
Po sąlyga		Kinta modelių parametrai (svoriai)

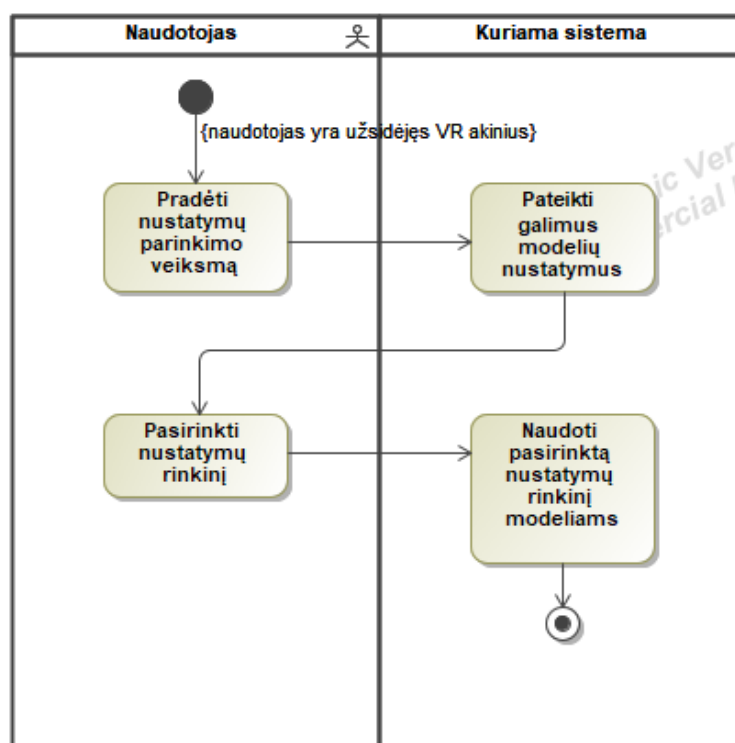


2.7 pav. „Atlikti modelių treniravimą“ veiklos diagrama

„Pasirinkti modelį“ ir „Parinkti parametrus“ panaudojimo atvejų veiklos diagrama pateiktos toliau (žr. 2.8 pav., 2.9 pav.). Programuotojas gali pasirinkti modelius ir parametrus arba naudoti nustatytus pagal nutylėjimą.



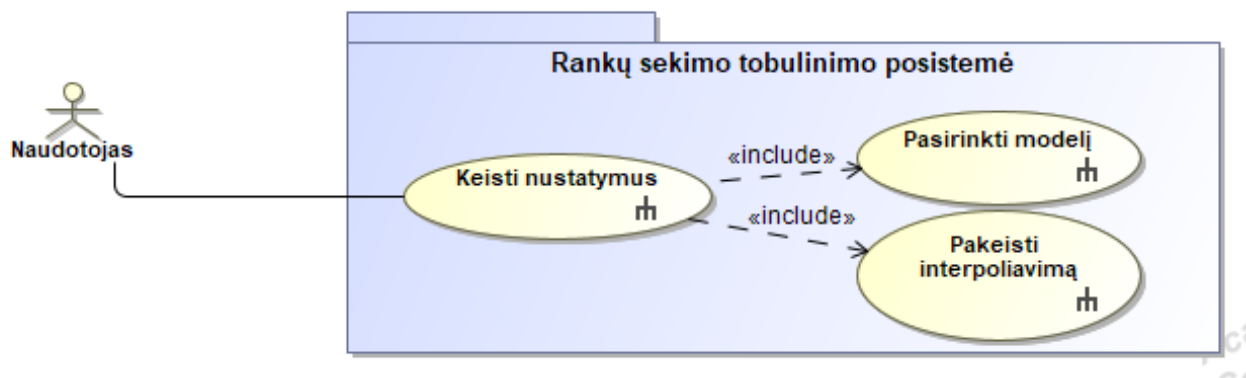
2.8 pav. „Parinkti modelį“ veiklos diagrama



2.9 pav. „Parinkti parametrus“ veiklos diagrama

2.1.3. Rankų sekimo tobulinimo posistemė

Ši posistemė leidžia naudotojui keisti nustatymus, susijusius su rankų sekimu – pasirinkti rankų sekimo tobulinimo modelį ir parinkti papildomus interpoliavimo nustatymus. Posistemė sudaryta iš trijų veiksmų (žr. 2.10 pav.).

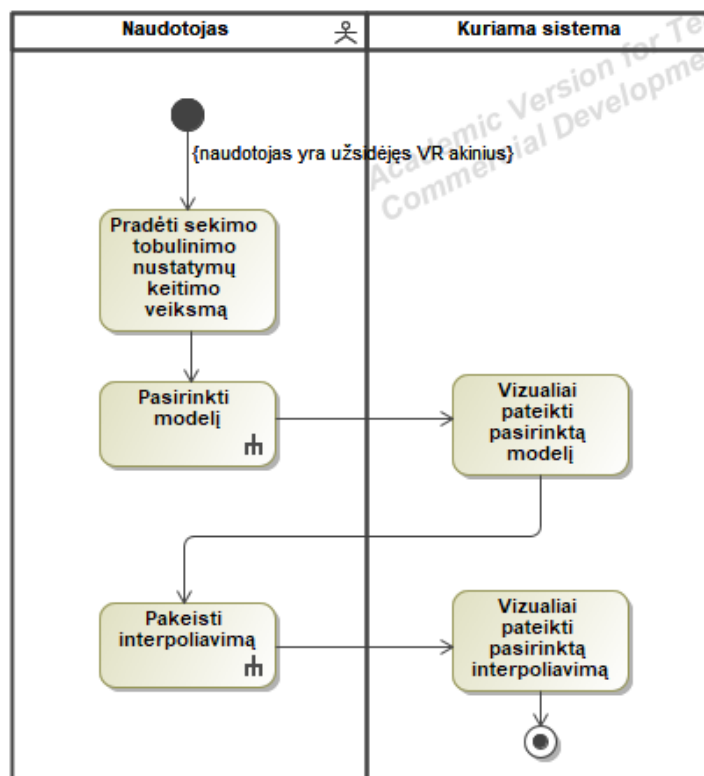


2.10 pav. Rankų sekimo tobulinimo posistemės panaudojimo atvejų diagrama

„Keisti nustatymus“ panaudojimo atvejo specifikacija ir veiklos diagrama pateiktos toliau (žr. 2.6 lentelė, 2.11 pav.). Naudotojas gali keisti rankų sekimo tobulinimo sistemos nustatymus, pasirinkdamas modelį ir interpoliavimo parinktį.

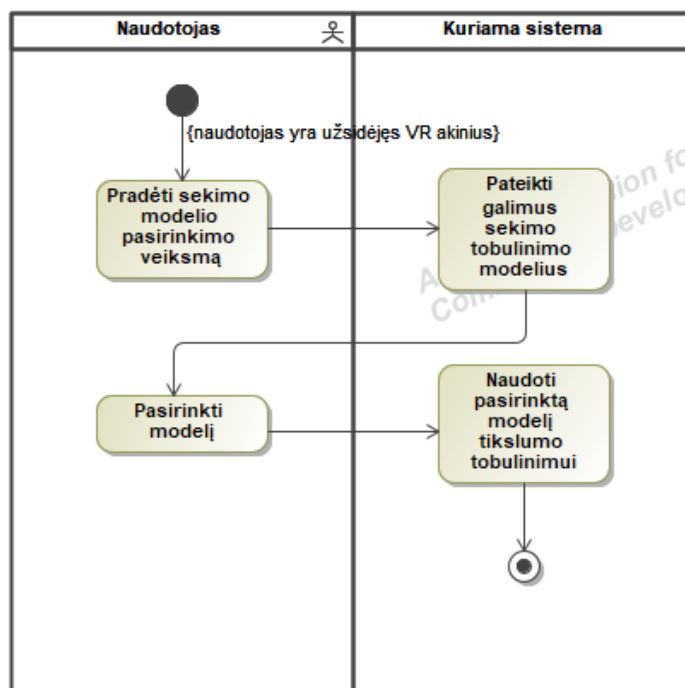
2.6 lentelė. „Keisti nustatymus“ specifikacija

Panaudojimo atvejis. Keisti nustatymus		
Tikslas. Leisti naudotojui keisti rankų sekimo tobulinimo sistemos naudojamą modelį ir nustatymus		
Aprašymas.		
Prieš sąlyga		Naudotojas yra užsidėjęs VR akinius
Aktoriai		Naudotojas
Susiję panaudojimo atvejai	Apimami PA	Pasirinkti modelį Pakeisti interpoliavimą
	Išplečiantys PA	-
	Specializuojami PA	-
Po sąlyga		

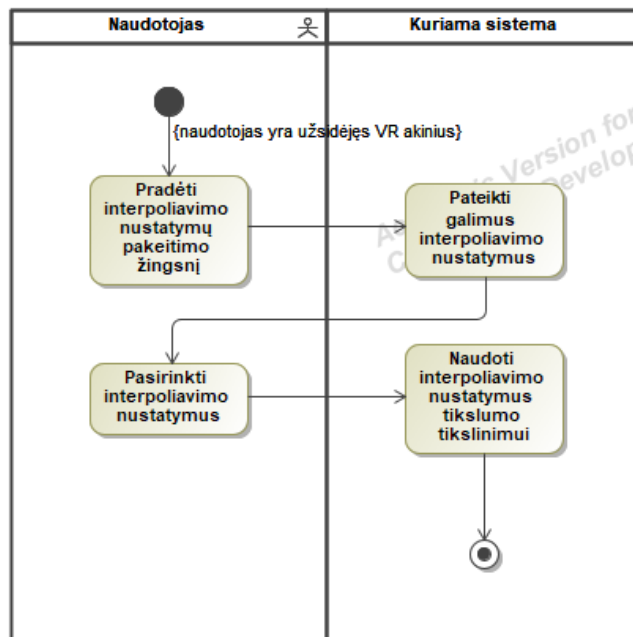


2.11 pav. „Keisti nustatymus“ veiklos diagrama

„Pasirinkti modelį“ ir „Pakeisti interpoliavimą“ panaudojimo atvejų veiklos diagrama pateiktos toliau (žr. 2.12 pav., 2.13 pav.). Programuotojas gali pasirinkti modelius ir parametrus arba naudoti nustatytus pagal nutylėjimą.



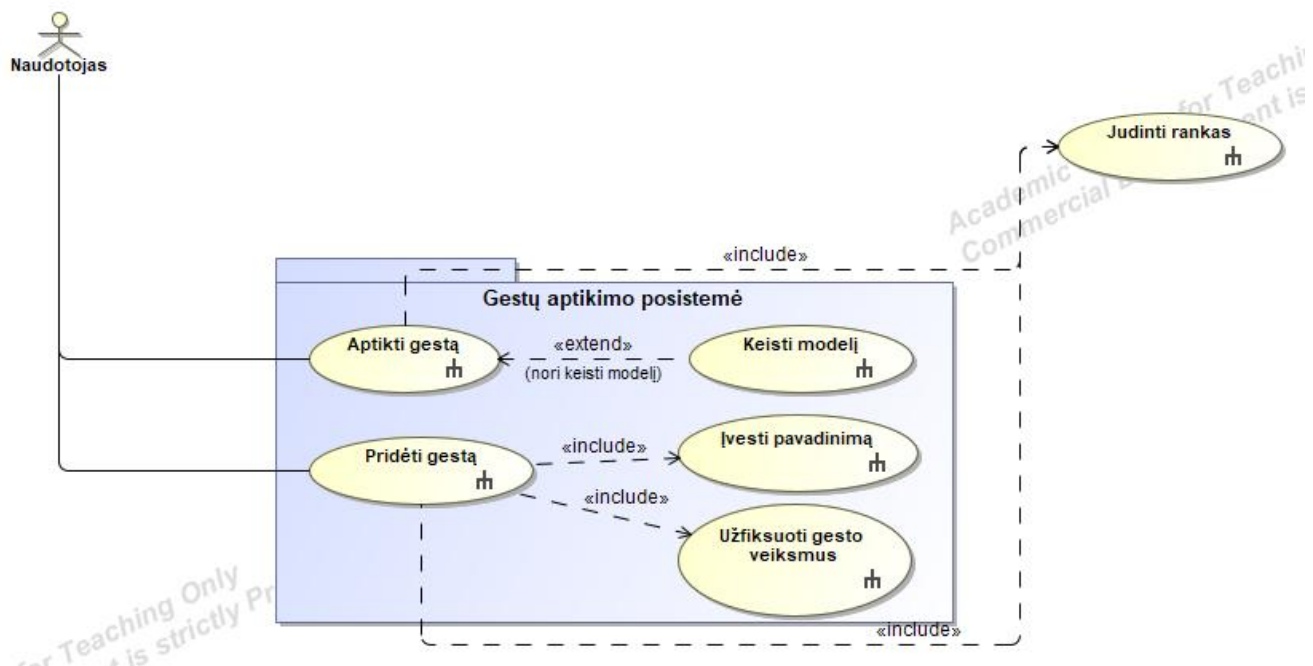
2.12 pav. „Pasirinkti modelį“ veiklos diagrama



2.13 pav. „Pakeisti interpoliavimą“ veiklos diagrama

2.1.4. Gestų aptikimo posistemė

Gestų aptikimo posistemė atsakinga už galimybę naudotojui pridėti naujus gestus ir aptikti jau esamus gestus. Gestų pridėjimui reikia įvesti pavadinimą ir užfiksuoti gesto veiksmus. Posistemė sudaryta iš penkių veiksmų: gesto aptikimo, gesto aptikimo modelio keitimo, gesto pridėjimo, sudaryto iš pavadinimo įvedimo ir gesto užfiksavimo (žr. 2.14 pav.).

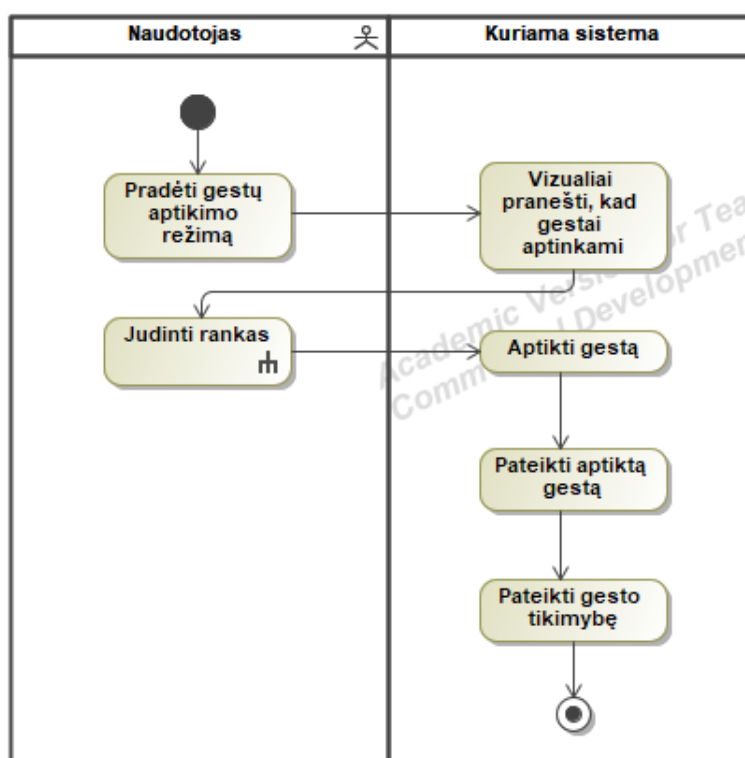


2.14 pav. Gestų aptikimo posistemės panaudojimo atvejų diagrama

„Aptikti gestą“ panaudojimo atvejo specifikacija ir veiklos diagrama pateiktos toliau (žr. 2.7 lentelė, 2.15 pav.). Naudotojas gali aptikti gestus, pateiktus sistemos gestų duomenų rinkiniuose.

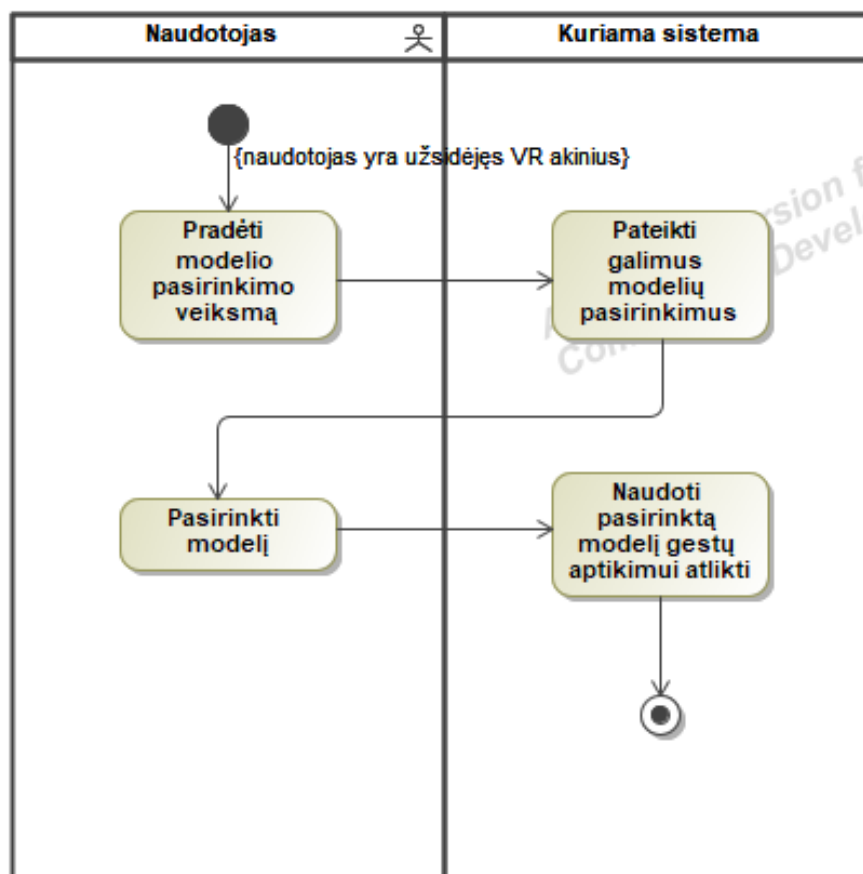
2.7 lentelė. „Aptikti gestą“ specifikacija

Panaudojimo atvejis. Aptikti gestą		
Tikslas. Sistema turi leisti naudotojui užfiksuoti jau esamus gestus		
Aprašymas. Naudotojas turi galėti aptikti sistemoje išsaugotus gestus. Naudotojui pateikiama informacija apie aptinkamą gestą: pavadinimas ir gesto aptikimo tikimybė		
Prieš sąlyga		Naudotojas yra užsidėjęs VR akinius
Aktoriai		Naudotojas
Susiję panaudojimo atvejai	Apimami PA	-
	Išplečiantys PA	Keisti modelį
	Specializuojami PA	-
Po sąlyga		Aptinkamas naudotojo atliekamas statinis arba dinaminis gestas



2.15 pav. „Aptikti gestą“ veiklos diagrama

„Keisti modelį“ panaudojimo atvejo veiklos diagrama pateikta toliau (žr. 2.16 pav.). Programuotojas keisti gestų aptikimui naudojamą modelį.

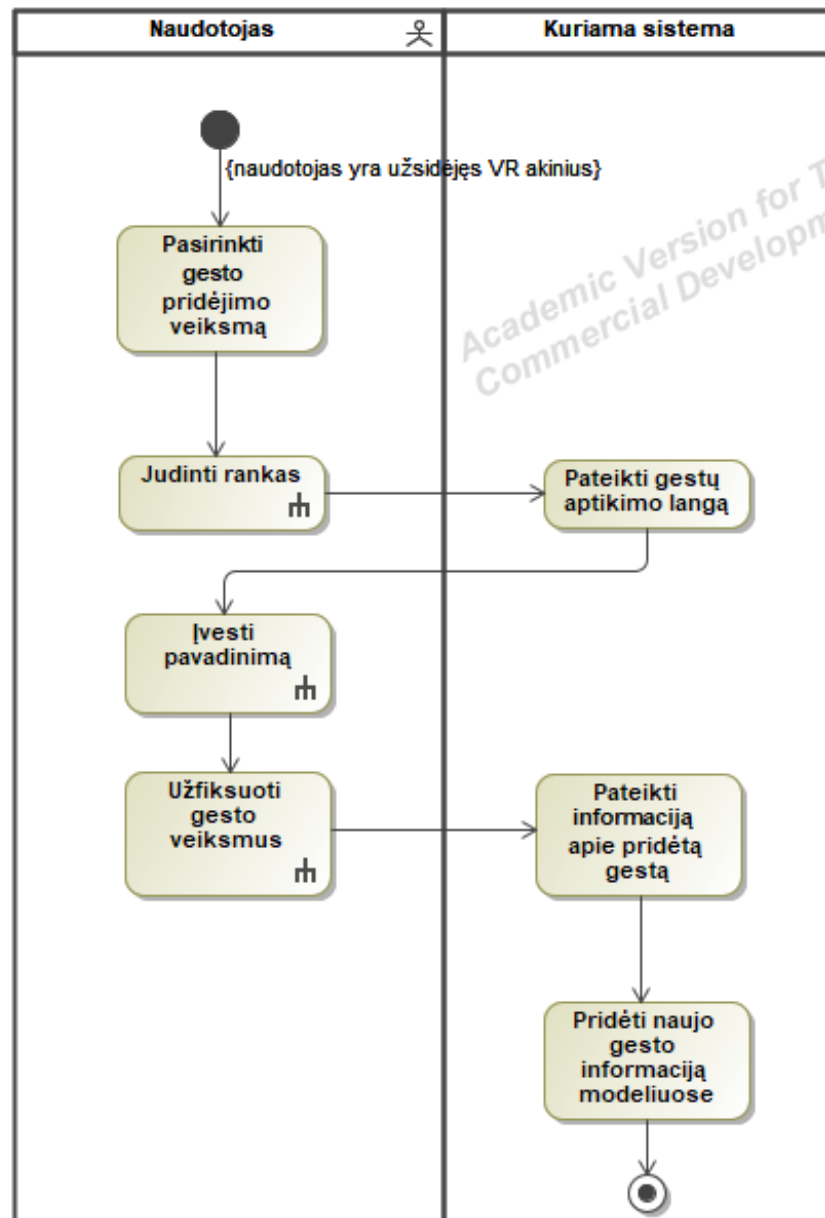


2.16 pav. „Keisti modelį“ veiklos diagrama

„Pridėti gestą“ panaudojimo atvejo specifikacija ir veiklos diagrama pateiktos toliau (žr. 2.8 lentelė, 2.17 pav.). Naudotojas gali pridėti naują gestą, įvesti pavadinimą ir užfiksuoti judesius.

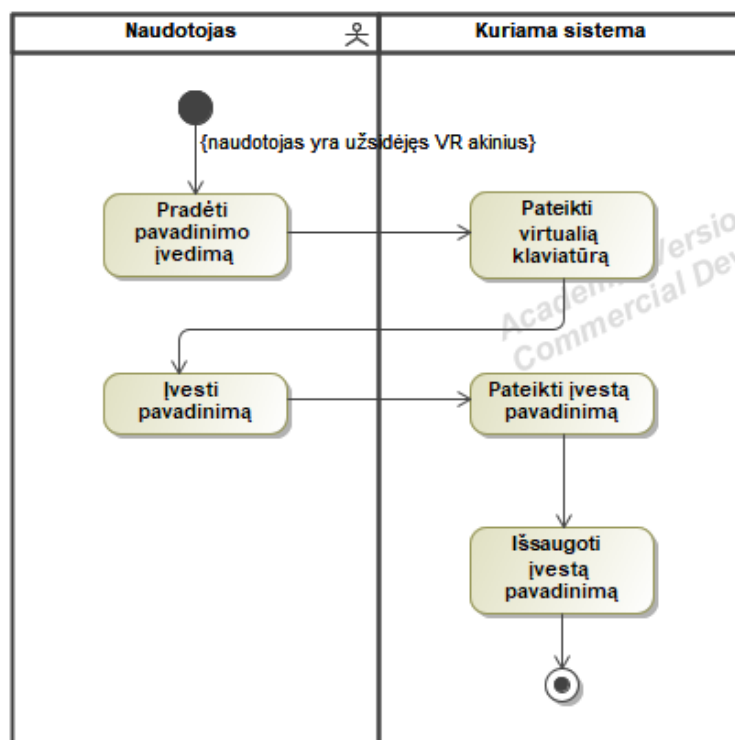
2.8 lentelė. „Pridėti gestą“ specifikacija

Panaudojimo atvejis. Pridėti gestą		
Tikslas. Leisti naudotojui pridėti naujus gestus į sistemą		
Aprašymas. Naudotojas turi turėti galimybę įvesti naują gestą, sudarytą iš pavadinimo ir atliekamo veiksmo. Išsaugotas gestas pridedamas prie sistemos		
Prieš sąlyga		Naudotojas yra užsidėjęs VR akinius
Aktoriai		Naudotojas
Susiję panaudojimo atvejai	Apimami PA	-
	Išplečiantys PA	-
	Specializuojami PA	-
Po sąlyga		Išsaugojamas naujas gestas, kuris yra pateikiamas į gestų aptikimo modelį

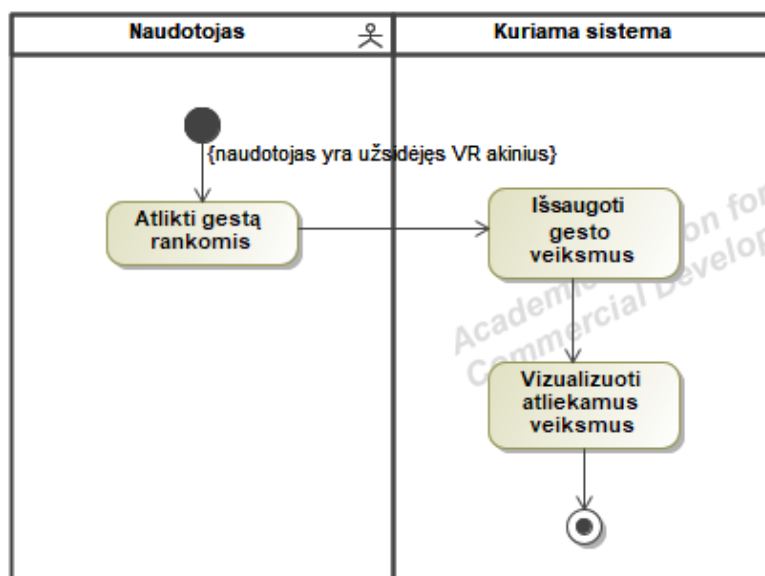


2.17 pav. „Pridėti gestą“ veiklos diagrama

„Įvesti pavadinimą“ ir „Užfiksuoti gesto veiksmus“ panaudojimo atvejų veiklos diagrama pateiktos toliau (žr. 2.18 pav., 2.19 pav.). Naudotojas gali pridėti naujo gesto pavadinimą virtualia klaviatūra ir įrašyti naujo gesto veiksmus rankomis.



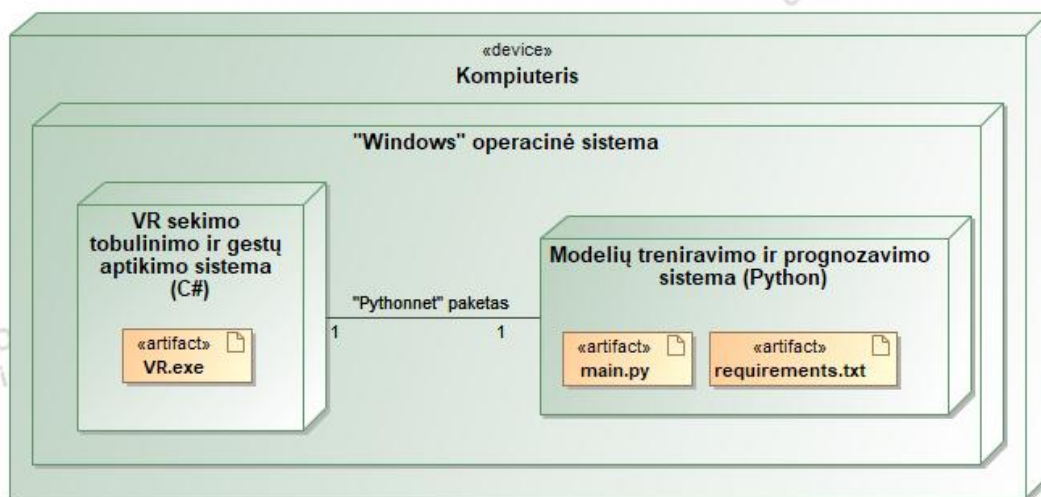
2.18 pav. „Įvesti pavadinimą“ veiklos diagrama



2.19 pav. „Užfiksuoti gesto veiksmus“ veiklos diagrama

2.2. Diegimo diagrama

Toliau pateikiama sistemos diegimo diagrama kai komunikacijai naudojami „Pythonnet“ paketas (platesnė informacija apie kilusią komunikacijos problemą pateikiama realizacijos skyriuje) (žr. 2.20 pav.). Šis paketas leidžia lokaliai siųsti duomenis tarp *Unity* (C#) ir modelių prognozavimo sistemos (*Python*) užtikrinant greitaveiką ir abipusę komunikaciją.



2.20 pav. Sistemos diegimo diagrama, kai naudojamas „Pythonnet“ paketas

2.3. Realizacijai keliami nefunkciniai reikalavimai

Sistemos nefunkciniai reikalavimai pateikiami toliau:

Našumo:

- minimalus vaizdo kadrų dažnis, naudojant „Meta Quest 2“ akinius, turi būti ne mažesnis nei 60 kadrų per sekundę;
- vidutinis vaizdo kadrų dažnis, naudojant „Meta Quest 2“ akinius, turi būti ne mažesnis nei 90 kadrų per sekundę;
- dirbtinio intelekto algoritmai, naudojami rankų sekimo tikslumo padidinimui gerinti, turi veikti greičiau nei 60 kartų per sekundę (16,66 ms arba mažiau tarp pateikimo ir išvesties) dažniu.

Duomenų formato:

- surinkti duomenys turi būti saugomi „MessagePack“ formatu;
- modelių svoriai ir architektūra turi būti saugomi PTH formatu.

Kultūriniai-politiniai:

- sistemos sąsaja kuriama anglų kalba.

Saugumo:

- papildomi naudotojų duomenys (įvesties informacija treniravimui) turi būti saugomi tik lokaliai.

Veikimo sąlygų:

- sistema turi veikti naudojant „Meta Quest 2“ virtualiosios realybės akinius;
- sistema turi veikti „Windows“ operacinę sistemą palaikančiuose įrenginiuose.

2.4. Techninė specifikacija

Sistemos naudojimui reikalingas kompiuteris, palaikantis „Windows“ operacinę sistemą. Techninei specifikacijai apibrėžti pateikiama lentelė, kurioje apibrėžiama minimali ir rekomenduojama techninė įranga (remiantis „Meta“ duomenimis [42], žr. 2.9 lentelė).

2.9 lentelė. Minimali ir rekomenduojama techninė įranga

Komponentas	Minimali techninė įranga	Rekomenduojama techninė įranga
Pagrindinis procesorius	Intel i5-4590 / AMD Ryzen 5 1500X arba geresnis	Intel i7 / AMD Ryzen 7 arba geresnis
Grafinis procesorius	NVIDIA: Titan X, GeForce GTX 970, GeForce GTX 1070, GeForce GTX 1080, GeForce GTX 1650 Super, GeForce GTX 1660, GeForce GTX 1660 TI, GeForce RTX 20-serija (išskyrus 2050), GeForce RTX 30 serija ⁴ , GeForce RTX 40 serija AMD: 400 serija, 500 serija, 5000 serija, 6000 serija ⁵ , Vega serija	Nvidia RTX 20 ⁶ serija / AMD Radeon RX 6000 serija arba geresnis
Operatyvioji atmintis	8 GB RAM arba daugiau	16 GB DDR4 RAM arba daugiau
Operacinė sistema	„Windows 10“, „Windows 11“	„Windows 10“, „Windows 11“
USB prievadai	Vienas USB 3.0 prievadas	Vienas USB C 3.0 prievadas

Dirbtinio intelekto modelių treniravimui ir testavimui bendri techninės įrangos reikalavimai neapibrėžiami.

Atsižvelgiant į informaciją, pateiktą „PyTorch“ svetainėje [43], reikalaujama įrašytos „Python“ 3.8 arba aukštesnės versijos, įrašyto „PyTorch“ paketo (su *CUDA* (angl. *Compute Unified Device Architecture*) versija jei naudojama vaizdo plokštė, palaikanti šią lygiagrečiojo skaičiavimo platformą, arba versiją, skirtą procesoriui), „Windows 7“ arba naujesnės versijos, paketų valdiklio („Anaconda“, „pip“, arba „LibTorch“).

Remiantis „Puget Systems“ rekomendacijomis [44] galima apibendrinti rekomenduojamą techninę įrangą:

- Pagrindinis procesorius – nors bendru atveju vaizdo plokštės naudojimas yra greitesnis ir efektyvesnis, galingas pagrindinis procesorius yra svarbus, kadangi duomenų analizė ir tvarkymas dažniausiai vykdomas naudojant centrinį procesorių. Taip pat gali būti atveju, kai vaizdo plokštės pasiekia tam tikrus limitus, pavyzdžiui integruotos atminties (angl. *video random access memory*, *VRAM*) trūkumą. Rekomenduojami *Intel Xeon W* arba *AMD Threadripper Pro* procesoriai, kadangi užtikrina didelį patikimumą, ir palaiko galimybę naudoti keletą vaizdo plokščių vienu metu.
- Grafinis procesorius – rekomenduojama naudoti *NVIDIA* vaizdo plokštės, kadangi jos yra populiariausios, turi plačiausią palaikymą. Verta naudoti vaizdo plokštės, kurios palaiko pusinį tikslumą (FP16), lyginant su pilnu (FP32). Nors šis standartas turi mažesnę tikslumą, dažnu atveju dirbtinio intelekto algoritmams ir gilesiems neuroniniams tinklams tai pakankamas tikslumas. Rekomenduojami grafiniai procesoriai – *NVIDIA GeForce RTX 3080/3080Ti/3090/4080/4090* (bendram naudojimui tinkamos vaizdo plokštės) arba *NVIDIA RTX A5000/A6000/A100* (profesionalios specializuotos vaizdo plokštės su dideliu kiekiu *VRAM*).

⁴ Nerekomenduojama naudoti *NVIDIA 3050* ir *3050ti* nešiojamo kompiuterio versijų.

⁵ Nerekomenduojama naudoti *RX 6500* vaizdo plokštės.

⁶ Išskyrus *NVIDIA GeForce RTX 2050*.

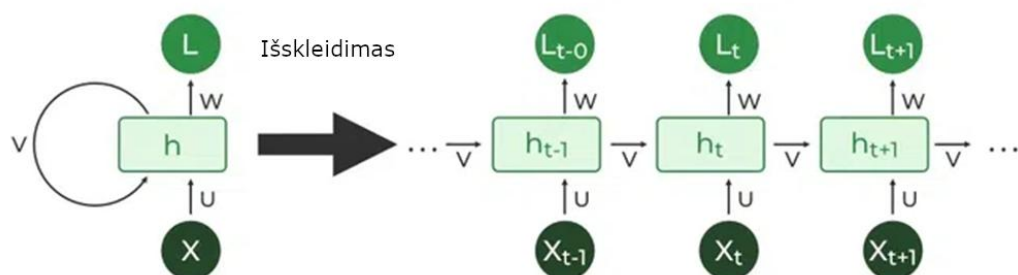
- Operatyvioji atmintis – rekomenduojama turėti dvigubai daugiau RAM atminties, nei naudojama VRAM. Jei naudojamos dvi *NVIDIA GeForce RTX 3090* vaizdo plokštės (48 GB VRAM) rekomenduojama naudoti konfigūraciją su 96 GB RAM.

Sistema pritaikyta virtualiosios realybės sistemoms, kurios palaiko „OpenXR“ formatą. Darbo metu atliekami bandymai naudojant „Meta Quest 2“ akinius naudojant „Meta Quest Link“ programinę įrangą. Dirbtinio intelekto algoritmai treniruojami ir vykdo prognozes naudojant kompiuterį su 13 kartos *Intel Core i7-13620H* centriniu procesoriumi (iki 4,9 GHz), 16 GB DDR5 5200 MHz RAM atmintimi, *NVIDIA GeForce RTX 4060* grafikos procesoriumi, naudojant „Windows 10“ operacinę sistemą.

2.5. Dirbtinio intelekto algoritmai

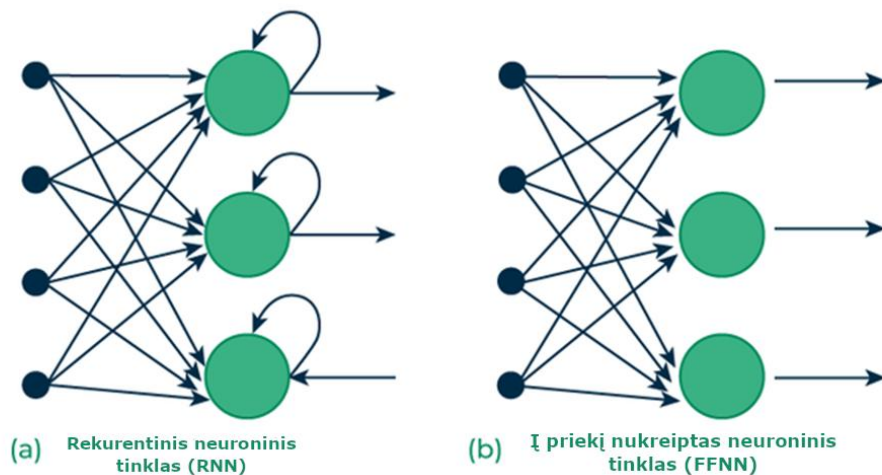
Rankų sąnarių sekimo tikslumui didinti planuojama surinkti duomenų rinkinius, kuriuose aptinkami visi galimi sąnariai. Sudaromi treniravimo, testavimo ir validavimo rinkiniai. Treniravimo rinkiniui skiriama 60% duomenų, testavimo ir validavimo rinkimiams skiriama po 20%. Iš treniravimo duomenų rinkinio pašalinami tam tikri sąnariai, bandoma atlikti prognozę apie pašalinto sąnario poziciją ir pasukimą, turint informaciją apie visus kitus sąnarius.

Projekto atlikimui naudojami modifikuoti rekurentiniai neuroniniai tinklai ir ilgalaikės trumpalaikės atminties modeliai, įgyvendinant papildomus metodus siekiant padidinti tikslumą. Bendra architektūra ir palyginimas pateikiama toliau (žr. 2.21 pav., 2.22 pav., 2.23 pav.). Naudojamos nedidelės architektūros, siekiant gauti modelius, atitinkančius išsikeltus greitaveikos reikalavimus. Jei LSTM architektūros neužtikrins pakankamo išvados laiko, bus naudojama GRU, kadangi jų architektūros yra mažesnės ir dažnu atveju greitesnės.



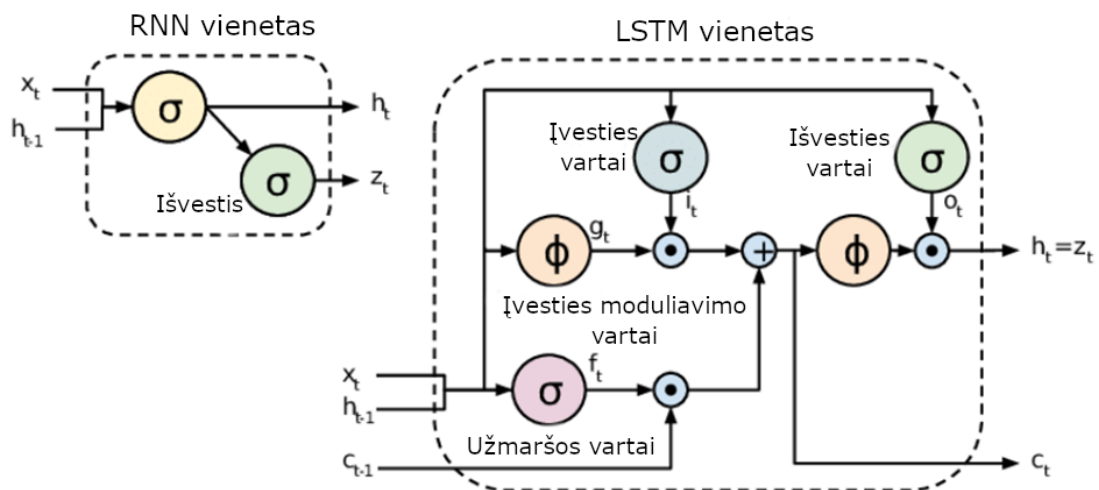
2.21 pav. RNN architektūros pavyzdys [45]

Čia h_t – paslėpta tinklo būsena tam tikru laiko momentu, x_t yra įvestis tam tikru laiko momentu, L_t – išvestis tam tikru laiko momentu. Prieš pateikiant išvestį paslėpta būsena atnaujinama rekursyviai, naudojant ankstesnę būseną h_{t-1} ir įvestį x_{t-1} .



2.22 pav. RNN ir įprastų neuroninių tinklų, kuriuose perdavimas vyksta į priekį palyginamasis paveikslėlis [45]

Esminis skirtumas yra tai, kad rekurentiniame neuroniniame tinkle prieš pateikiant išvestį paslėpta būsena atnaujinama rekursiškai.



2.23 pav. RNN ir LSTM palyginimas [46]

RNN ir LSTM struktūros palyginime pastebimas skirtingas vartų kiekis. Čia x_t – įvestis t laiko momentu, h_{t-1} – ankstesnio laiko momento paslėpta būsena, σ – aktyvacijos funkcija, z_t – gaunama išvestis. LSTM papildomai turi atminties būseną c_t ir keletą papildomų vartų. Užmaršos vartai f_t nustato, kokią informaciją reikia pamiršti iš atminties būsenos, įėjimo vartai i_t nustato, kokią naują informaciją reikia pridėti prie atminties būsenos, įėjimo moduliavimo vartai g_t moduliuoja naują informaciją, o_t valdo, kuri atminties būsenos dalis bus išvedama kaip h_t .

2.6. Metrikos ir klaidos funkcijos

Gauti rezultatai bus lyginami pagal du pagrindinius kriterijus – greitaveiką ir prognozių kokybę. Greitaveika įvertinama pagal vaizdo kadų per sekundę dažnį ir išvados laiką (angl. *inference time*) milisekundėmis. Prognozių kokybė įvertinama naudojant klaidos funkcijas treniravimo metu ir naudojant metrikas pozicijos ir pasukimo prognozėms. Problemai spręsti apibrėžtos specifinės metrikos. Pozicijos prognozėms įvertinti pateikiama šaknies iš vidutinės kvadratinės paklaidos (angl. *RMSE*, *root mean square error*), vidutinė absoliutinė paklaida (angl. *MAE*, *mean absolute error*),

vidutinio Euklido atstumo (angl. *MED*, *mean Euclidean distance*), maksimalaus Euklido atstumo (angl. trumpinys *MaxED*) antrojo maksimalaus Euklido atstumo, vidutinio absoliutinio laikinio skirtumo metrikos. Pateikiamos ir specifinių sąnarių (riešo ir smiliaus piršto galo) vidutinio Euklido atstumo metrikos. Pasukimo prognozėms įvertinti naudojama kvaterniono kampinio skirtumo ir geodezinės paklaidos. Toliau pateikiamos metrikų formulės.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (P_i - T_i)^2}; \quad (1)$$

čia N – duomenų rinkinio kiekis, P_i – prognozuojama pozicija, T_i – tikrinė pozicija. RMSE yra jautrus didelėms paklaidoms, yra lengvai interpretuojamas dėl sutampančių vienetų (visos metrikos su vienetais pateikiamos metrais).

$$MAE = \frac{1}{N} \sum_{i=1}^N |P_i - T_i|; \quad (2)$$

čia N – duomenų rinkinio kiekis, P_i – prognozuojama pozicija, T_i – tikrinė pozicija. MAE yra atsparesnis išimtims, lengvai interpretuojamas dėl sutampančių vienetų.

$$MED = \frac{1}{N} \sum_{i=1}^N ||P_i - T_i||; \quad (3)$$

čia N – duomenų rinkinio kiekis, P_i – prognozuojama pozicija, T_i – tikrinė pozicija, $||P_i - T_i||$ Euklido vektoriaus norma. MED parodo geometrinę paklaidą trimatėje erdvėje, lengvai interpretuojamas įvertinant poziciją.

$$MaxED = \max_{i \in [1, N]} ||P_i - T_i||; \quad (4)$$

čia N – duomenų rinkinio kiekis, P_i – prognozuojama pozicija, T_i – tikrinė pozicija, $||P_i - T_i||$ Euklido vektoriaus norma. Parodo prasčiausio pozicijos prognozės atvejį. Atliekant eksperimentus pastebėta, kad ši metrika dažnu atveju aptinka kraštinius atvejus (angl. *edge cases*). Geresnei interpretacijai pridedama antroji maksimalaus Euklido atstumo metrika.

$$Vid. \ abs. \ laik. \ skirtumas = \frac{1}{N(f-1)} \sum_{m=1}^M \sum_{t=1}^{f-1} \sum_{j=1}^J ||p_{m,t+1,j} - p_{m,t,j}||; \quad (5)$$

čia M – trajektorijų skaičius, f – prognozuojamų laiko žingsnių skaičius, J – sąnarių kiekis, $p_{m,t,j}$ – pozicija m , laiko žingsnyje t , sąnariui j . Ši metrika įvertina pozicijos pasikeitimą tarp gretimų laiko žingsnių. Leidžia įvertinti tolygumą, mažesnė reikšmė atitinka sklandesnį, be staigių skirtumų judesį.

$$Kvaterniono \ vidutinis \ kampinis \ skirtumas = \frac{1}{N} \sum_{i=1}^N \cos^{-1} \left(\frac{R_i \cdot Q_i}{||R_i|| \cdot ||Q_i||} \right); \quad (6)$$

čia R_i – pasukimo kvaterniono prognozė, Q_i – pasukimo kvaterniono tikrinė reikšmė. Parodo kampinį skirtumą tarp prognozės ir tikrinio kvaterniono. Šią metriką galima lengviau interpretuoti nei kvaterniono geodezinę paklaidą.

$$Kvaterniono\ geodezinė\ paklaida = \frac{1}{N} \sum_{i=1}^N 2 \cdot \cos^{-1}(|\langle Q_i, R_i \rangle|); \quad (7)$$

čia $\langle Q_i, R_i \rangle$ – skaliarinė sandauga tarp kvaterniono prognozės ir tikrinės reikšmės. Dažnai naudojama metrika pasukimo tikslumui įvertinti.

Klaidos funkcija – tai matematinė funkcija, kuri matuoja skirtumą tarp modelio prognozuojamos reikšmės ir realios reikšmės pagal pateiktą kriterijų. Darbe planuojama įgyvendinti metodą, leidžiantį apskaičiuoti bendra klaidos funkciją parenkant skirtingas metrikas pagal prognozuojamas ypatybes. Kituose moksliniuose darbuose sąnarių erdvinės informacijos prognozių tikslumui įvertinti naudojama vidutinio kvadrato paklaida arba vidutinė absoliutinė paklaida, tačiau sprendžiant darbe aprašomą problemą nuspręsta naudoti specifines ir lengviau interpretuojamas klaidos funkcijas.

Pagrindinės prognozuojamos ypatybės yra sąnarių pozicija ir pasukimas erdvėje. Pozicija aprašoma kaip trimatis vektorius erdvėje (x, y ir z koordinatės). Pozicijai sukurtos klaidos funkcijos:

1. Vidutinio Euklido atstumo klaida – skaičiuoja vidutinį Euklido atstumą tarp prognozuotos sąnario pozicijos ir realios (tikslinės) sąnario pozicijos. Mažėjanti klaidos funkcija užtikrina mažą atstumų skirtumą tarp prognozuojamos ir realios pozicijos.
2. Sąnario atstumo pirštuose klaida – skaičiuoja skirtumą tarp prognozuotos sąnario pozicijos ir realios sąnario pozicijos naudojant rankos sąnarių jungčių hierarchiją. Užtikrina mažą atstumų skirtumą tarp tam tikrų sąnarių kiekviename piršte.

Sakykime, kad $P \in \mathbb{R}^{B \times N \times F}$ – prognozuojamos pozicijos, o $T \in \mathbb{R}^{B \times N \times F}$ – tikslinės pozicijos, čia B – duomenų rinkinio dydis (angl. *batch size*), N – sąnarių kiekis, F – sąnario koordinatės (x, y, z). Tada kiekvienam sąnariui j iš duomenų rinkinio Euklido atstumas tarp prognozuotos pozicijos p_{ij} ir tikslinės pozicijos t_{ij} skaičiuojamas pagal formulę:

$$d_{ij} = ||p_{ij} - t_{ij}|| = \sqrt{\sum_{k=1}^F (p_{ijk} - t_{ijk})^2}; \quad (8)$$

čia p_{ijk} – prognozuojama k -oji sąnario j koordinatė duomenų rinkinyje i , t_{ijk} – tikslinė k -oji sąnario j koordinatė duomenų rinkinyje i . Gauname atstumų matricą D :

$$D \in \mathbb{R}^{B \times N}; \quad (9)$$

čia kiekvienas d_{ij} žymi sąnario j atstumą duomenų rinkinyje i . Tada vidutinio Euklido atstumo paklaida bus skaičiuojama pagal formulę:

$$Vidutinė\ Euklido\ klaida = \frac{1}{B \times N} \sum_{i=1}^B \sum_{j=1}^N d_{ij}; \quad (10)$$

čia B – duomenų rinkinio dydis (*batch size*), N – sąnarių kiekis, d_{ij} – Euklido atstumas sąnariui j duomenų rinkinyje i .

Sąnario atstumo pirštuose paklaida apskaičiuojama panašiu principu – šiuo atveju bandoma minimizuoti atstumus tarp kiekvieno piršto sąnarių, o ne bendrą atstumą tarp visų sąnarių. Skaičiuojant atstumus atrenkami specifiniai sąnariai pagal sąnarių hierarchiją. Matuojamas atstumas tarp:

- Smiliaus delnakaulio ir smiliaus pamatinio pirštakaulio; smiliaus pamatinio pirštakaulio ir smiliaus vidurinio pirštakaulio; smiliaus pamatinio pirštakaulio ir smiliaus galinio pirštakaulio; smiliaus galinio pirštakaulio ir smiliaus galo.
- Didžiojo delnakaulio ir didžiojo pamatinio pirštakaulio; didžiojo pamatinio pirštakaulio ir didžiojo vidurinio pirštakaulio; didžiojo pamatinio pirštakaulio ir didžiojo galinio pirštakaulio; didžiojo galinio pirštakaulio ir didžiojo galo.
- Bevardžio delnakaulio ir bevardžio pamatinio pirštakaulio; bevardžio pamatinio pirštakaulio ir bevardžio vidurinio pirštakaulio; bevardžio pamatinio pirštakaulio ir bevardžio galinio pirštakaulio; bevardžio galinio pirštakaulio ir bevardžio galo.
- Mažylio delnakaulio ir mažylio pamatinio pirštakaulio; mažylio pamatinio pirštakaulio ir mažylio vidurinio pirštakaulio; mažylio pamatinio pirštakaulio ir mažylio galinio pirštakaulio; mažylio galinio pirštakaulio ir mažylio galo.
- Nykščio delnakaulio ir nykščio galinio pirštakaulio; nykščio galinio pirštakaulio ir nykščio galo.

Pasukimas erdvėje aprašomas naudojant kvaternionus – keturmačius vektorius, skirtus pasukimui vaizduoti. Pasukimui sukurtos klaidos funkcijos:

1. Kvaterniono geodezinis nuostolis – skaičiuoja vidutini geodezinį atstumą tarp prognozuoto ir realaus (tikslinio) kvaterniono.
2. Kvaterniono kampinis nuostolis – skaičiuoja vidutinį kampo skirtumą tarp prognozuoto ir realaus (tikslinio) kvaterniono. Ši metrika yra lengviau interpretuojama.

Sakykime, kad turime kvaternionus $q_{prog} = (q_x, q_y, q_z, q_w)$ – prognozuotas kvaternionas ir $q_{tiksl} = (t_x, t_y, t_z, t_w)$ – tikslinis kvaternionas. Kvaternionai yra normalizuojami taip, kad jų ilgis būtų lygus 1:

$$q'_{prog} = \frac{q_{prog}}{||q_{prog}||}; \quad q'_{tiksl} = \frac{q_{tiksl}}{||q_{tiksl}||} \quad (11)$$

$$||q|| = \sqrt{q_x^2 + q_y^2 + q_z^2 + q_w^2} \quad (12)$$

Skaičiuojama skaliarinė sandauga pagal formulę:

$$dot = q'_{prog} \cdot q'_{tiksl} \quad (13)$$

Geodezinis atstumas paskaičiuojamas pagal formulę:

$$d_{geo}(q'_{prog}, q'_{tiksl}) = 2 \cdot \arccos(|dot|); \quad (14)$$

čia $\arccos$ grąžina kampinį atstumą radianais.

Kvaterniono geodezinis nuostolis apskaičiuojamas pagal formulę:

$$Kvaterniono \text{ geodezinis nuostolis} = \frac{1}{N} \sum_{i=1}^N d_{geo}(q'_{prog}, q'_{tiksl}) = \frac{1}{N} \sum_{i=1}^N 2 \cdot \arccos(|q'_{prog} \cdot q'_{tiksl}|) \quad (15)$$

Kvaterniono geodezinis nuostolis skaičiuoja geodezinį kampų skirtumą tarp kvaternionų radianais, naudojant geodezinį atstumą. Kvaterniono kampinis nuostolis apskaičiuojamas kaip kampo kosinusas tarp kvaternionų pagal formulę:

$$Kampinis\ kvaterniono\ nuostolis = \frac{1}{N} \sum_{i=1}^N (1 - |q'_{prog} \cdot q'_{tiksl}|); \quad (16)$$

čia q'_{prog} ir q'_{tiksl} – normalizuoti prognozės ir tiksliniai kvaternionai.

Kuriama klaidos funkcija, leidžianti apjungti keletą klaidos funkcijų į vieną, naudojant α parametą, nurodantį klaidos funkcijos svorį:

$$Bendra\ klaidos\ funkcijos\ reikšmė = \alpha \times klaida_1 + (1 - \alpha) \times klaida_2 \quad (17)$$

Pozicijos prognozėms naudojama vidutinė Euklido klaida sujungta su sąnario atstumo pirštuose klaida kai $\alpha = 0,75$.

2.7. Testavimas

Pateikiamas testavimo planas ir testavimo kriterijai kodo peržiūrai, statinei kodo analizei atlikti, automatizuotų vienetų ir integracinių testų kūrimui ir rankinio testavimo atlikimui.

2.7.1. Testavimo planas

Sistemos testavimui palengvinti ir veikloms apibūdinti sukuriamas testavimo planas:

1. Programinio kodo peržiūra – atliekama po naujo funkcionalumo pridėjimo. Programinio kodo peržiūra leidžia išlaikyti kodo kokybę.
2. Statinė kodo analizė – automatizuota, naudojamas „ReSharper“ įrankis, integruotas į „Visual Studio 2022“ ir „JetBrains Rider“ integruotas kūrimo aplinkas. Statinė kodo analizė padeda išlaikyti pasirinktą formatavimą, skatina naudoti gerąsias programavimo praktikas. Kadangi šis įrankis veikia automatiškai, statinė analizė vykdoma lygiagrečiai su kodo rašymu.
3. Automatizuoti vienetų ir integraciniai testai – kuriami testai, leidžiantys patikrinti atskirų funkcijų ir bendrą sistemos veikimą. Testai kuriami lygiagrečiai su funkcionalumo įgyvendinimu. Testai paleidžiami kiekvieną kartą įgyvendinant pakeitimus ir išsaugant kodą debesyse.
4. Rankinis testavimas – kadangi tam tikrų komponentų ir veiksmų VR erdvėje negalime ištestuoti automatizuotais testais, šis funkcionalumas išbandomas rankiniu būdu. Testavimas atliekamas naudojant scenarijus.

2.7.2. Testavimo kriterijai

Pateikiami apibrėžti kriterijai testavimo veikloms siekiant išlaikyti gera testų kokybę.

Kodo peržiūros kriterijai:

- programiniame kode ir komentaruose nėra teksto rašybos klaidų;
- naudojamos gerosios c# programavimo kalbos praktikos;
- sudėtingi metodai paaiškinami naudojant komentarus anglų kalba;
- skirtingi veiksmi atliekami atskirose funkcijose;
- laikomasi švaraus kodo principų (naudojami tinkami kintamųjų pavadinimai, išlaikoma kodo struktūra naudojant naujas eilutes ir tarpus, didelės funkcijos skaidomos į mažesnes).

Statinės kodo analizės kriterijai:

- programoje neturi būti nenaudojamų kintamųjų ir funkcijų;
- programoje nėra nenaudojamų direktyvų;
- programoje nėra nepasiekiamo kodo;
- naudojamas apibrėžtas formatavimas („PascalCase“ naudojamas viešoms savybėms, metodų ir klasių pavadinimams, „camelCase“ naudojamas kintamųjų, esančių metodų viduje apibrėžimui, pabraukimo žymėjimas (underscore notation) naudojamas privačioms savybėms ir kintamiesiems.

Automatizuotų vienetų ir integracinių testų kriterijai:

- testai grupuojami pagal posistemas;
- testų metodų pavadinimai kuriami naudojant apibrėžtą notaciją: „MetodoPavadinimas_RezultatasKurioTikimasi“ - naudojamas „CamelCase“ ir pabraukimo skiriamasis simbolis. Pirmoje pavadinimo dalyje pateikiamas testuojamo metodo pavadinimas, antroje – rezultatas. Jei metodas testuoja daug metodų ir naudoja „Unity“ metodus, pavadinimas sudaromas tik iš rezultato apibrėžimo dalies;
- testai turi būti izoliuoti, t. y. testų rezultatai turi būti nepriklausomi nuo kitų testų;
- visi testai turi būti įvykdomi sėkmingai. Aptikus triktis, jos turi būti pašalinamos nedelsiant.

Rankinio testavimo kriterijai:

- rankinis testavimas aprašomas naudojant scenarijus;
- turi būti ištestuotas visas sistemos funkcionalumas;
- rankinio testavimo scenarijai turi būti detalūs, lengvai atkartojami.

3. Rankų sąnarių virtualiojoje realybėje prognozių sistemos įgyvendinimas

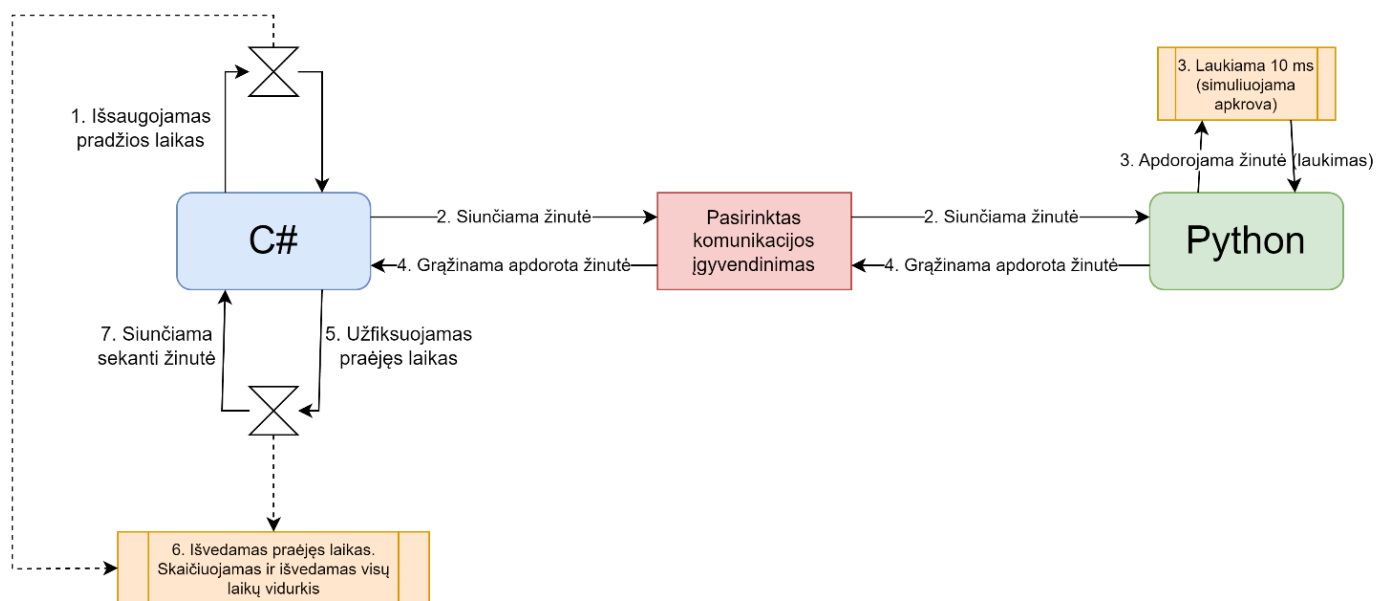
Šioje dalyje pateikiama informacija apie duomenų surinkimo, vizualizavimo, rankos pirštų sąnarių pozicijos ir pasukimo prognozių posistemį ir naudotų metodų įgyvendinimą. Rankos pirštų sąnarių erdvinės informacijos prognozėms atlikti reikalinga duomenų surinkimo sistema, leidžianti lengvai įrašyti įvairius judesius, kurių informaciją galima naudoti treniruojant dirbtinio intelekto modelius. Prognozių vykdymo atveju reikalingas įgyvendinimas, leidžiantis komunikuoti tarp *Unity* žaidimų variklio VR įvesties ir dirbtinio intelekto metodų. Atliekamas tyrimas, siekiant išsiaiškinti optimalų komunikacijos metodą informacijos perdavimui realiu laiku. Vizualizavimo sistema įgyvendinta siekiant palengvinti įrašytų rankų sekimo duomenų peržiūrą, palengvinti klaidų aptikimą įgyvendinant algoritmus ir pateikti prognozuotus rankų sąnarius naudotojui užtikrinant greitaveiką.

3.1. Duomenų surinkimo ir vizualizavimo posistemės įgyvendinimas

Sukurta posistemė, skirta duomenų surinkimui, skirtų rankų sekimo prognozavimui, kai sekimas nutrūksta, modeliams treniruoti ir vizualizuoti. Tiriami galimi komunikacijos metodai, siekiant aptikti optimalų sprendimą, leidžiantį vykdyti abipusę komunikaciją realiu laiku.

3.1.1. Komunikacija tarp C# ir Python posistemų

Viena iš problemų, su kuriomis buvo susidurta surenkant duomenis – komunikacijos užtikrinimas tarp C# programinio kodo („Unity“ sistemos dalis) ir Python (dirbtinio intelekto algoritmų sistemos dalis, naudojamas „PyTorch“). Atliekamas tyrimas siekiant pasirinkti dvikryptės komunikacijos metodą, kuris užtikrintų greitaveiką. Sudaryta bandymų aplinka – programinio kodo blokai C# ir Python programavimo kalboje, kai bandymų metu 1000 kartų kviečiamas komunikacijos metodas, kurio metu išsiunčiama nustatyto dydžio žinutė, gaunamas atsakas ir užfiksuojamas abipusės komunikacijos laikas. Python programavimo kalboje kviečiamas papildomas metodas, kurio metu gavus žinutę laukiama 10 milisekundžių, prieš išsiunčiant atsaką (simuliuojama metodo apkrova). Naudojama „Stopwatch“ C# klasė, skirta laiko stebėjimui. Maksimalus šios klasės tikslumas yra 1 milisekundė, po 1000 iteracijų išvedamas laiko delsos vidurkis, ignoruojant pirmas 10 žinučių. Schema pateikiama toliau (žr. 3.1 pav.).



3.1 pav. Komunikacijos greitaveikos bandymų schema

Išbandyti 5 komunikacijos sprendimo būdai:

- „Python Scripting“ „Unity“ paketas;
- UDP tinklo lizdai (angl. *socket*);
- TCP tinklo lizdai;
- gRPC sistema;
- „Pythonnet“ paketas.

„Python Scripting“ – „Unity“ paketas, skirtas palengvinti scenų kūrimą ir kitus procesus, leidžiant į „Unity“ integruoti *Python* programinį kodą. Šis paketas skirtas užtikrinti platesnį gamybos procesą, kadangi galima įgyvendinti sąveiką su *Python* paremtomis žiniasklaidos ir pramogų (angl. *media and entertainment*) pramonės sistemomis [47]. Šis paketas leidžia įrašyti *Python* „pip“ paketus, nustatyti papildomus nustatymus, naudojant *Python* scenarijaus rengyklės langą. Šis įrankis skirtas techniniams menininkams (angl. *technical artists*) ir technikos skyriaus direktoriams (angl. *pipeline technical directors*). Tačiau atliekant bandymus, pastebėta, kad šis paketas yra netinkamas, kadangi *Python* programinio kodo negalima vykdyti kodo vykdymo metu (angl. *at runtime*).

UDP tinklo lizdai – tai tinklo lizdai, kurie remiasi naudotojo duomenų datagramos protokolu (angl. *User Datagram Protocol*). UDP – tai komunikacijos protokolas, kuris naudojamas veikloms, kurios reikalauja realaus laiko apdorojimo: žaidimai, vaizdo transliacija, DNS (srities vardų struktūra, angl. *Domain Name System*) paieškoje. Šis protokolas yra ypač greitas, tačiau nėra patikimas, neužtikrina duomenų eiliškumo ir pristatymo, neturi klaidų taisymo mechanizmų. Siekiant užtikrinti didelį rankų sekimo tikslinimo ir gestų aptikimos sistemos tikslumą šiuos mechanizmus reiktų įgyvendinti papildomai nors bandymų metu (siunčiant 1000, 10000 ir 1000000 paketų) nepristatytų paketų ir neteisingu eiliškumu išsiųstų paketų nepastebėta. Naudojami 4096 baitų dydžio paketai, siekiant vienu metu išsiųsti pilną žinutę. Toliau pateikiamas paskutinių 10 išvesčių delsos laikas, vidutinis laikas ir gauta galutinė delsa atėmus 10 milisekundžių (kurios pridedamos *Python* pusėje) (žr. 3.2 pav., pateikiama anglų kalba kadangi programa pagal apibrėžtus reikalavimus turi anglų kalbos sąsają). Gauta vidutinė delsa yra lygi 1,27 milisekundėms.

```
Current latency: 11 ms
Average latency: 11,271805273833671 ms
Current latency: 11 ms
Average latency: 11,271529888551164 ms
Current latency: 12 ms
Average latency: 11,272267206477732 ms
Current latency: 11 ms
Average latency: 11,271991911021233 ms
Current latency: 11 ms
Average latency: 11,271717171717171 ms
Final average latency (-10ms): 1,271717171717171 ms
```

3.2 pav. UDP tinklo lizdų rezultatai

TCP tinkle lizdai – tai tinko lizdai, kurie remiasi transmisijos kontrolės protokolu (angl. *Transmission Control Protocol*). TCP – priešingai nei UDP, užtikrina patikimą duomenų perdavimą, turi klaidų atitaisymo mechanizmus, užtikrina eiliškumą. Šis protokolas bendru atveju yra lėtesnis, tačiau *C#* programavimo kalboje turi papildomų nustatymų ir režimų, leidžiančių padidinti greitaveiką. Atliktų

bandymu metu pastebėta, kad greیتaveika priklauso nuo paketo dydžio ir yra lėtesnė, lyginant su UDP tinklo lizdais (žr. 3.3 pav.). Gauta delsa lygi 1,73 milisekundėms.

```
Current latency: 11 ms
Average latency: 11,726166328600407 ms
Current latency: 11 ms
Average latency: 11,72644376899696 ms
Current latency: 11 ms
Average latency: 11,725708502024291 ms
Current latency: 11 ms
Average latency: 11,725985844287159 ms
Current latency: 11 ms
Average latency: 11,725252525252525 ms
Final average latency (-10ms): 1,7252525252525253 ms
```

3.3 pav. TCP tinklo lizdų rezultatai

gRPC – tai moderni keleto platformų atviro kodo nuotolinio procedūrų iškviėtimo sistema, kurios programinį kodą galima sugeneruoti ir lengvai pritaikyti skirtingoms sistemoms. Ši technologija leidžia efektyviai sujungti paslaugas duomenų centruose, palaiko apkrovos balansavimą, sekimą, patikrinimus ir autentifikavimą [48]. Ši sistema pasižymi greیتaveika ir yra skirta dvipusei komunikacijai. Naudojamas *proto* formato failas, kuriame nurodoma žinutės užklauso ir atsako formatai ir pagrindinis komunikacijos servisas. Iš *proto* failo informacijos sugeneruojamas *Python* ir *C#* programinis kodas. *Python* kodas generuojamas naudojant „grpcio-tools“ „pip“ paketą. *C#* naudojami „Grpc.Tools“, „Grpc.Core“ ir „Google.Protobuf“ „NuGet“ paketai. Gauta vidutinė abipusės komunikacijos delsa yra lygi 1,86 milisekundės (žr. 3.4 pav.).

```
Current latency: 11 ms
Average latency: 11,846385542168674 ms
Current latency: 11 ms
Average latency: 11,84553660982949 ms
Current latency: 20 ms
Average latency: 11,853707414829659 ms
Current latency: 13 ms
Average latency: 11,854854854854855 ms
Current latency: 13 ms
Average latency: 11,856 ms
Final average latency (-10ms): 1,8559999999999999 ms
```

3.4 pav. gRPC rezultatai

„Pythonnet“ – tai paketas, kuris suteikia *Python* programuotojams sklandžią integraciją su „.NET Common Language Runtime (CLR)“ ir suteikia įrankius *.NET* programinio kodo kūrėjams. Integracija leidžia abipusę komunikaciją – duomenų perdavimas vykdomas tarp *C#* ir *Python* programavimo kalbomis sukurtų sistemų. *C#* bandymams įrašytas „pythonnet“ „NuGet“ paketas. Kintamųjų apibrėžimui naudojamas dinaminis tipas. Pastebėta, kad komunikacija vykdoma ypač greitai – bandymų metu per 1000 iteracijų aptiktas tik vienas kartas, kai žinutės perdavimas užtruko ilgiau nei vieną milisekundę. Realų greیتaveikos tikslumą sunku užfiksuoti dėl naudojamos „Stopwatch“ klasės maksimalaus vienos milisekundės tikslumo. Pateikiami gauti rezultatai (žr. 3.5 pav.).

```

Current latency: 10 ms
Average latency: 10,008113590263692 ms
Current latency: 10 ms
Average latency: 10,008105369807497 ms
Current latency: 10 ms
Average latency: 10,008097165991902 ms
Current latency: 10 ms
Average latency: 10,00808897876643 ms
Current latency: 10 ms
Average latency: 10,008080808080807 ms
Final average latency (-10ms): 0,008080808080807245 ms

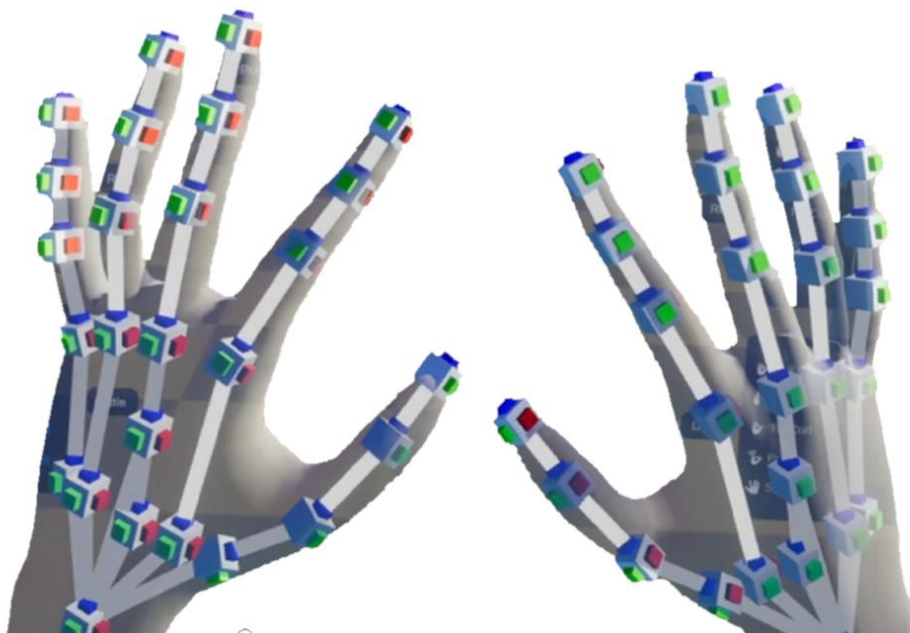
```

3.5 pav. „Pythonnet“ rezultatai

Nuspręsta komunikacijai naudoti „Pythonnet“ dėl ypač didelės greitaveikos. Sistemoje įgyvendinti papildomi metodai, leidžiantys sklandžiai perteikti duomenis ir pastebėti klaidas.

3.1.2. Duomenų surinkimas ir vizualizavimas

Darbe sprendžiama problema reikalauja įvesties ir prognozės duomenų pateikimo sistemos naudotojui VR erdvėje. Sukurta pradinė scena su plokštuma ir grafiniais elementais. Naudojami „Unity“ „XR“, „XR Core Utilities“, „XR Plugin Management“, „Mathematics“, „Input System“ ir „XR Hands“ paketai. Sukurta rankų sąnarių vizualizavimo sistema, jos pavyzdys pateikiamas toliau (žr. 3.6 pav.). Kiekvienai rankai užfiksuojami 26 sąnariai, pateikiami lentelėje (žr. 3.1 lentelė). Pateikiama sąnarių pasukimo vizualizacija naudojant tris spalvas: žalia rodo sąnario teigiama Y ašį, mėlyna rodo teigiamą Z ašį ir raudona rodo teigiamą X ašį lokaliaje sąnarių erdvėje.



3.6 pav. Įgyvendintas rankos sąnarių vaizdavimas virtualioje erdvėje

3.1 lentelė. Sekami rankos sąnariai

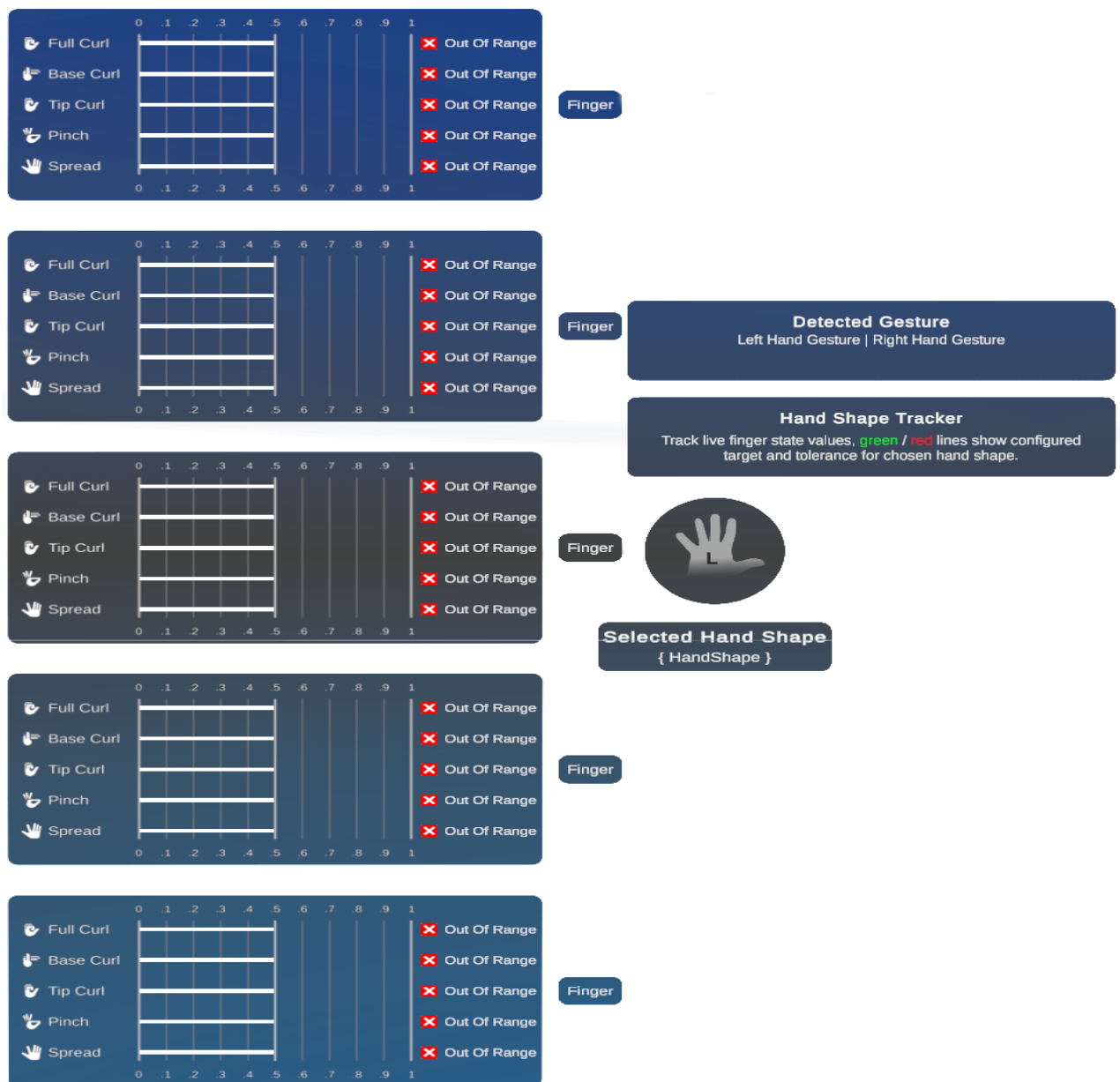
Nr.	Identifikatoriaus pavadinimas (anglų k.)	Sąnario pavadinimas (anglų k.)	Sąnario (rankos dalies) pavadinimas
1.	„Wrist“	<i>Wrist</i>	Riešas
2.	„Palm“	<i>Palm</i>	Delnas
3.	„ThumbMetacarpal“	<i>Thumb Metacarpal</i>	Nykščio delnakaulio
4.	„ThumbProximal“	<i>Thumb Proximal</i>	Nykščio pamatinio pirštakaulio
5.	„ThumbDistal“	<i>Thumb Distal</i>	Nykščio galinio pirštakaulio
6.	„ThumbTip“	<i>Thumb Tip</i>	Nykščio galo (tai nėra sąnarys, tačiau fiksuojama piršto pabaiga)
7.	„IndexMetacarpal“	<i>Index Metacarpal</i>	Smiliaus delnakaulio
8.	„IndexProximal“	<i>Index Proximal</i>	Smiliaus pamatinio pirštakaulio
9.	„IndexIntermediate“	<i>Index Intermediate</i>	Smiliaus vidurinio pirštakaulio
10.	„IndexDistal“	<i>Index Distal</i>	Smiliaus tolimojo pirštakaulio
11.	„IndexTip“	<i>Index Tip</i>	Smiliaus galo
12.	„MiddleMetacarpal“	<i>Middle Metacarpal</i>	Didžiojo delnakaulio
13.	„MiddleProximal“	<i>Middle Proximal</i>	Didžiojo pamatinio pirštakaulio
14.	„MiddleIntermediate“	<i>Middle Intermediate</i>	Didžiojo vidurinio pirštakaulio
15.	„MiddleDistal“	<i>Middle Distal</i>	Didžiojo galinio pirštakaulio
16.	„MiddleTip“	<i>Middle Tip</i>	Didžiojo galo
17.	„RingMetacarpal“	<i>Ring Metacarpal</i>	Bevardžio delnakaulio
18.	„RingProximal“	<i>Ring Proximal</i>	Bevardžio pamatinio pirštakaulio
19.	„RingIntermediate“	<i>Ring Intermediate</i>	Bevardžio vidurinio pirštakaulio
20.	„RingDistal“	<i>Ring Distal</i>	Bevardžio galinio pirštakaulio
21.	„RingTip“	<i>Ring Tip</i>	Bevardžio galo
22.	„LittleMetacarpal“	<i>Little Metacarpal</i>	Mažylio delnakaulio
23.	„LittleProximal“	<i>Little Proximal</i>	Mažylio pamatinio pirštakaulio
24.	„LittleIntermediate“	<i>Little Intermediate</i>	Mažylio vidurinio pirštakaulio
25.	„LittleDistal“	<i>Little Distal</i>	Mažylio galinio pirštakaulio
26.	„LittleTip“	<i>Little Tip</i>	Mažylio galo

Virtualioje erdvėje galima vizualizuoti surinktus duomenų rinkinius. Pasukus kairę ranką pateikiama grafinė naudotojo sąsaja, leidžianti pasirinkti duomenų rinkinį ir vizualizavimo tipą. Virtualioje erdvėje pateikiamos naudotojo rankų kopijos, kurios atkartoja surinktus duomenis. Tai leidžia vizualizuoti surinktus duomenų rinkinius ir pastebėti rankų dinamiką po transformacijų.

Naudojama „XR Hands“ paketo 1.4.1 versijos teikiama naudotojo sąsaja, leidžianti aptikti statinius gestus. Toliau pateikiamas eskizas, kuriame pateikiama statinių gestų aptikimo naudotojo sąsaja anglų kalba. (žr. 3.7 pav.). Šios sistemos veikimo principas – aptinkami penki veiksmai kiekvienam pirštui:

- pilnas piršto sulenkimas (angl. *full curl*);
- pamatinio pirštakaulio pradžios sąnario sulenkimas 90 laipsnių kampu (angl. *base curl*);
- išorinės piršto dalies sulenkimas (pamatinio, vidurinio, tolimojo pirštakaulio ir piršto galo) (angl. *tip curl*);
- piršto galo sulietimas su nykščiu (angl. *pinch*);
- išskleidimas lyginant su kaimyniniais pirštais (angl. *spread*).

Galima apibrėžti statinius gestus, nurodant intervalus, įvertinančius ar veiksmas atliekamas, ar ne. Kombinuojant penkių pirštų kombinacijas apibrėžiami gestai, kuriuos vėliau galima aptikti.



3.7 pav. „XR Hands“ statinių gestų aptikimo naudotojo sąsaja (identiška informacija pateikiama ir dešinei rankai)

Sudarinėjant duomenų rinkinius nuspręsta rinkti didžiąją dalį „XR Hands“ paketo pateikiamos informacijos: sąnarių poziciją, pasukimą, tiesinį greitį, kampinį greitį ir anksčiau minėtų 5 pirštų atliekamų veiksmų įverčius.

Atliekant duomenų įrašymą duomenys saugomi keliose erdvėse. Galimų erdvių paaiškinimai pateikiami eksperimentų dalyje. Naudojamas „MessagePack“ duomenų formatas, leidžiantis informaciją saugoti neišnaudojant didelio kiekio atminties. Išsaugoti duomenų failai nuskaitomi prieš modelių treniravimą naudojant papildomą programinio kodo įgyvendinimą.

Papildomai atliekama ypatybių inžinerija (angl. *feature engineering*) siekiant išgauti papildomas sąveikas tarp rankos pirštų sąnarių. Pagal nutylėjimą SDK leidžia pasiekti sąnarių erdvinę informaciją tik tada, kai aptinkami visi sąnariai – virtualiosios realybės akiniams neaptikus bent vieno sąnario visiškai neteikiama informacija ir apie kitus sąnarius. Tokiu atveju sudarant duomenų rinkinį kadrai yra neįrašomi ir ignoruojami.

3.2. Rankų sekimo tobulinimo posistemės įgyvendinimas

Kuriama posistemė, skirta rankų sąnarių sekimo tobulinimui, siekiant prognozuoti sąnarių pozicijas ir pasukimą erdvėje kai nutrunka sekimas. Įgyvendinimas išskaidomas į dvi dalis: sąnarių pozicijų aptikimas naudojant (*Python*) ir aptiktų sąnarių pateikimas virtualioje erdvėje („Unity“ žaidimų variklis).

Sukurta dinamiška sistema, leidžianti nuskaityti ir transformuoti duomenų rinkinius, juos išskaidyti į treniravimo, validavimo ir testavimo imtis, treniruoti rekurentinių neuroninių tinklų modelius, kontroliuoti įvesties ir išvesties ypatybes, naudojamas prognozėse, keisti treniravimo parametrus, palengvinti treniravimo vizualizavimą pateikiant grafikus realiu laiku ir pateikti metrikas, leidžiančias nustatyti prognozių kokybę ir greitaveiką.

3.2.1. Duomenų nuskaitymas ir transformavimas

Sprendžiamos problemos specifika reikalauja papildomo įgyvendinimo siekiant apdoroti surinktą duomenų rinkinį. Įgyvendinta sistema leidžia naudoti duomenų rinkinių sąrašą pagal pavadinimus. Siekiant aptikti sąnarių pozicijas ir pasukimą nutarta naudoti tik surinktas esamų rankos pirštų sąnarių pozicijas, pasukimus, linijinį greitį ir kampinį greitį kiekvienam sąnariui. Prieš pradedant duomenų rinkinių nuskaitymą sukuriamas globalus kintamasis, nurodantis neuroninių tinklų įvestis ir išvestis. Duomenų rinkinys yra nuskaitymas, serializuojamas iš „MessagePack“ formato į *Python* objektą ir transformuojamas į vienmatį masyvą. Formuojant masyvą duomenys pateikiami iš eilės. Duomenų formatas kiekvienam duomenų rinkinio kadrai:

- Dvi rankos
 - Kairė ranka (apdorojama pirma)
 - Dešinė ranka
- Kiekviena ranka turi 26 sąnarius, kiekvienas sąnarys turi:
 - Poziciją (x, y, z)
 - Pasukimą (x, y, z, w)
 - Linijinį greitį (x, y, z)
 - Kampinį greitį (x, y, z)

Kiekvienas kadras f turi 676 ypatybes. Kiekviena ranka h turi 338 ypatybes, kiekvienas sąnarys j turi 13 ypatybių. Specifinės ypatybės indeksą vienmačiame masyve galime rasti pagal formulę:

$$\text{Kadro poslinkis} = f \times 676; \quad (18)$$

$$\text{Rankos poslinkis} = h \times 338; \quad (19)$$

$$\text{Sąnario poslinkis} = j \times 13; \quad (20)$$

$$\text{Ypatybės poslinkis} = k; \quad (21)$$

$$\begin{aligned} \text{Indeksas} = & \text{Kadro poslinkis} + \\ & \text{Rankos poslinkis} + \\ & \text{Sąnario poslinkis} + \\ & \text{Ypatybės poslinkis}; \end{aligned} \quad (22)$$

$$\begin{aligned} \text{Indeksas} = & f \times (H \times J \times K) + h \times (J \times K) + j \times K + k, \\ & \text{kai } f \in [0, F), h \in [0, 2), j \in [0, 26), k \in [0, 13); \end{aligned} \quad (23)$$

čia F – bendras kadrų skaičius duomenų rinkinyje, H – rankų kiekis kadre, J – sąnarių kiekis rankoje, K – ypatybių kiekis sąnaryje.

Bandant aptikti optimalų būdą duomenų įvesties pateikimui, kai treniruojami rekurentiniai neuroniniai tinklai, sukurti trys metodai:

1. Kuriamas modelis, kai įvestyje pateikiamos dvi rankos – gaunamas vienas vienmatis masyvas, kai $H=2$ ir viename kadre yra 676 ypatybės.
2. Kuriamas du modeliai kiekvienai rankai atskirai – gaunami du vienmačiai masyvai, kai $H=1$, viename kadre yra po 338 ypatybės.
3. Kuriamas vienas modelis, transformuojant dešinę ranką lyg ji būtų kairė – gaunamas vienas vienmatis masyvas, kuris yra papildomas du kartus (F padvigubėja, $H=1$)

Prieš pateikiant duomenis treniravimui, vienmatį masyvą reikia paversti tensoriumi – „PyTorch“ paketo centrine duomenų formato abstrakcija, leidžiančia atlikti skaičiavimus efektyviai ir naudojant grafinio vaizdo procesoriaus greitinimą. Atliekant treniravimą nuspręsta naudoti grafinį vaizdo procesorių siekiant pagreitinti treniravimo procesą bei geriau įvertinti greitaveikos galimybes, kai vienu metu naudojama virtualiosios realybės sistema. Sukurtos dvi tensorių kūrimo galimybės, įgyvendinant „PyTorch“ paketo „Dataset“ klasę:

1. Įprastas atvejis – duomenų rinkinys treniravimo metu yra perkėlinėjamas į grafinį vaizdo procesorių.
2. Duomenys iniciavimo metu perkeliama į grafinį vaizdo procesorių.

Darbe nuspręsta naudoti antrąjį atvejį, kadangi tai padidina greitaveiką treniruojant, kadangi išvengiama duomenų perkėlimo tarp centrinio procesoriaus ir grafinio vaizdo procesoriaus kiekvienoje epochoje. Išvengiama butelio kaklelio efekto (angl. *bottleneck*) dėl duomenų judėjimo.

Visi nuskaityti duomenų rinkiniai yra sujungiami į vieną „ConcatDataset“ tipo elementą. Šis elementas naudojamas siekiant užtikrinti vientisumą tarp duomenų formuojant treniravimo, validavimo ir testavimo duomenų rinkinius. Rekurentiniai neuroniniai tinklai reikalauja, kad duomenys treniruojant būtų pateikiami kaip vientisas rinkinys su iš eilės pateikiamais duomenimis. Šio rinkinio ilgis priklauso nuo rekurentinio neuroninio tinklo sekos ilgio L . Duomenys skaidomi į treniravimo, validavimo ir testavimo rinkinius naudojant 60%, 20% ir 20% paskirstymą. Treniravimo ir validavimo rinkiniai yra išmaišomi – sukuriama nuoseklios L ilgio grupės taip, kad viduje rinkinio eiliškumas išlieka tas pats, tačiau kinta bendras eiliškumas. Tai leidžia modeliui treniruojantis mažiau prisirišti prie tam tikrų duomenų rinkinių, gerėja modelio apibendrinimas (angl. *generalization*).

„Dataset“ klasės įgyvendinimas reikalauja sukurti metodą, kuris grąžintų du elementus – įvesties ypatybių rinkinį ir išvesties ypatybių rinkinį, naudojamą modelio treniravimui. Vienmatis masyvas pertvarkomas taip, kad įvesties ypatybių elementas atliekant iteracijas susiformuotų kaip trimatis masyvas, sudarytas iš duomenų rinkinio dydžio (angl. *batch size*), rekurentinio neuroninio tinklo sekos ilgio L ir bendro įvesties ypatybių kiekio kadre $H_{įvesties} \times J_{įvesties} \times K_{įvesties}$. Išvesties rinkinys formuojamas kaip dvimatis masyvas, sudarytas iš duomenų rinkinio dydžio ir bendro išvesties ypatybių kiekio kadre $H_{išvesties} \times J_{išvesties} \times K_{išvesties}$.

3.2.2. Modelių architektūra

Darbe nuspręsta lyginti modifikuotus GRU ir LSTM neuroninio tinklo tipus ir atvejus, kai pridedamas dėmesio mechanizmas (angl. *attention mechanism*) naudojant mažą kiekį neuronų ir sluoksnių

siekiant užtikrinti greitaveiką. Visuose neuroninių tinklų įgyvendinimuose naudojama sluoksnio normalizacija (angl. *layer normalization*) siekiant stabilizuoti ir pagerinti modelio konvergenciją. Naudojamas Xavier svorių inicializavimas siekiant pagerinti tinklo treniravimo procesą pradinio momentu. Šie metodai leidžia lengviau išvengti sprogsančių gradientų (angl. *exploding gradients*) problemos.

3.2.2.1. GRU architektūra

Darbe įgyvendinta paprasta GRU architektūra, susidedanti iš:

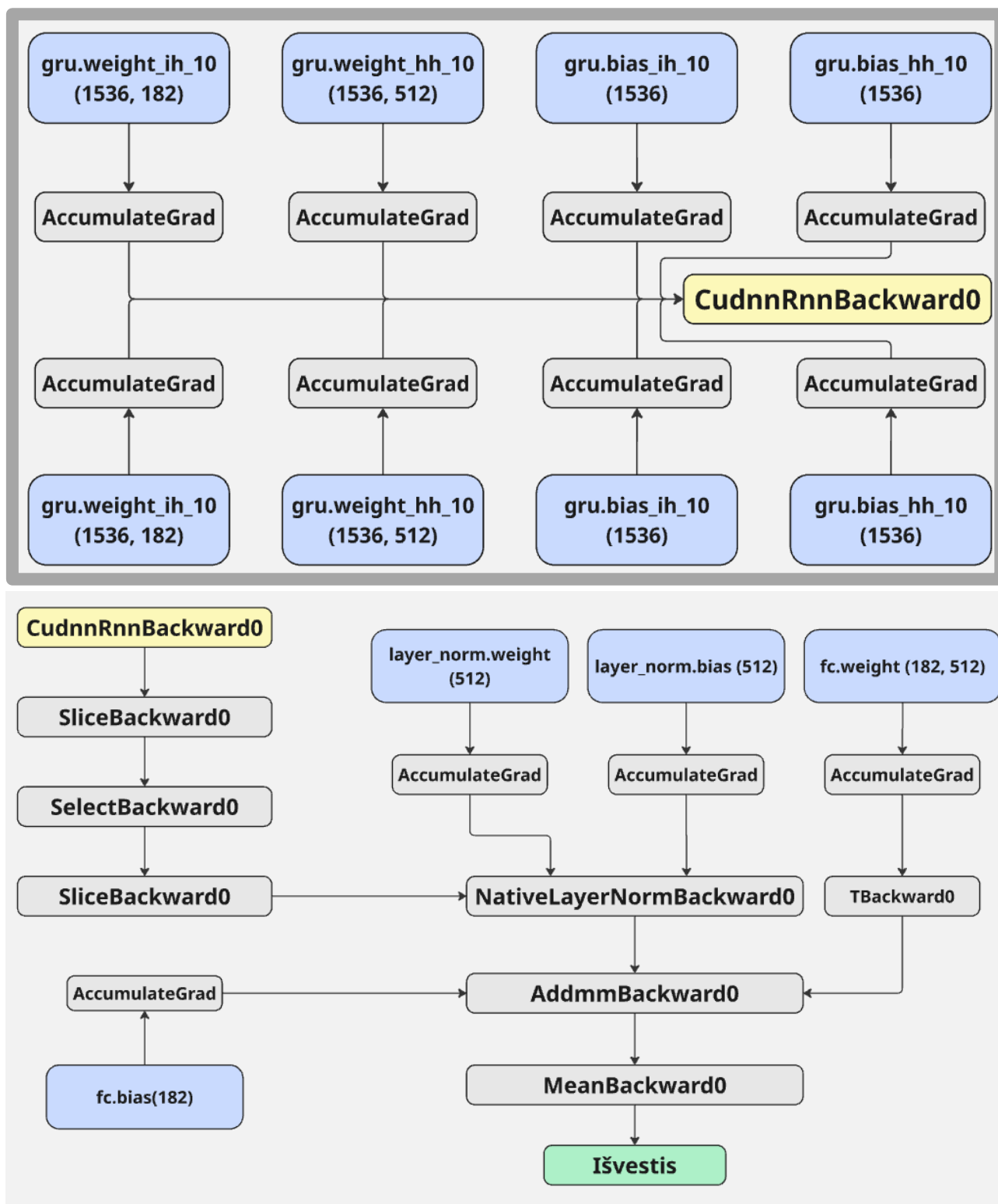
- „PyTorch“ „GRU“ sluoksnio įgyvendinimo – galima keisti sluoksnių kiekį ir paslėptų neuronų skaičių.
- Sluoksnio normalizacijos sluoksnis – normalizuoja paslėptų būsenų (*hidden states*) reikšmes.
- Pilnai sujungtas sluoksnis – transformuoja išėjimus iš GRU sluoksnio į norimą išvesties dimensiją. Naudojamas prognozei atlikti.

Toliau pateikiama architektūra su 2 sluoksniais ir 512 paslėptų svorių, sugeneruota naudojant „torchviz“ paketą (žr. 3.8 pav.). Įgyvendinta dinamiška sistema leidžia lengvai pakeisti sluoksnių ir paslėptų vienetų kiekį.

Mėlyna spalva pateikiami parametru mazgai. Šie mazgai atvaizduoja mokomuosius modelio parametrus (poslinkius ir svorius). „*gru.weight_ih_LX*“ rodo sluoksnio *X* iš įvesties į paslėptus svorius, „*gru.weight_hh_LX*“ rodo sluoksnio *X* iš paslėpto į paslėptą svorius, „*gru.bias_ih_LX*“ rodo sluoksnio *X* įėjimo į paslėptą sluoksnį poslinkius, „*fc.weight*“ ir „*fc.bias*“ rodo pilnai sujungto sluoksnio svorius ir poslinkius, „*layer_norm.weight*“ ir „*layer_norm.bias*“ rodo sluoksnio normalizavimo svorius ir poslinkius.

Pateikiami pilka spalva apibraukti mazgai, susiję su gradientais arba atgalinės sklaidos (angl. *backwards propogation*) skaičiavimo žingsniais. „*AccumulateGrad*“ kaupia gradientus per atgalinę sklaidą kiekvienam parametru, po atgalinės sklaidos šie gradientai naudojami atnaujinti optimizatoriaus parametrus. „*CudnnRnnBackward0*“ – atgalinės sklaidos mazgas, skaičiuoja gradientus GRU sluoksnio parametrams, remiantis klaidos funkcija atgalinės sklaidos metu, naudoja išvestis iš „*weight_ih_LX*“, „*weight_hh_LX*“, „*bias_ih_LX*“ ir „*bias_hh_LX*“ gradientų. „*NativeLayerNormBackward0*“ atlieka sluoksnio normalizacijos atgalinę sklaidą, apskaičiuoja normalizacijos svorius ir poslinkius. „*SliceBackward0*“ mazgas naudojamas pjovimo operacijoms (*slicing*), šiuo atveju tai vykdoma keičiant išvesties formą. „*TBackward0*“ tvarko matricų transponavimo operacijas, „*AddmmBackward0*“ rodo matricų daugybos atgalinę sklaidą pilnai sujungtame sluoksnyje, „*MeanBackward0*“ apskaičiuoja gradientus vidurkinimo operacijai.

Žalia spalva žymima išvestis. Vaizduoja galutinę klaidos funkciją (angl. *loss function*), apskaičiuotą treniravimo metu. Ši klaida naudojama atgalinei klaidai atnaujinant visus parametrus.



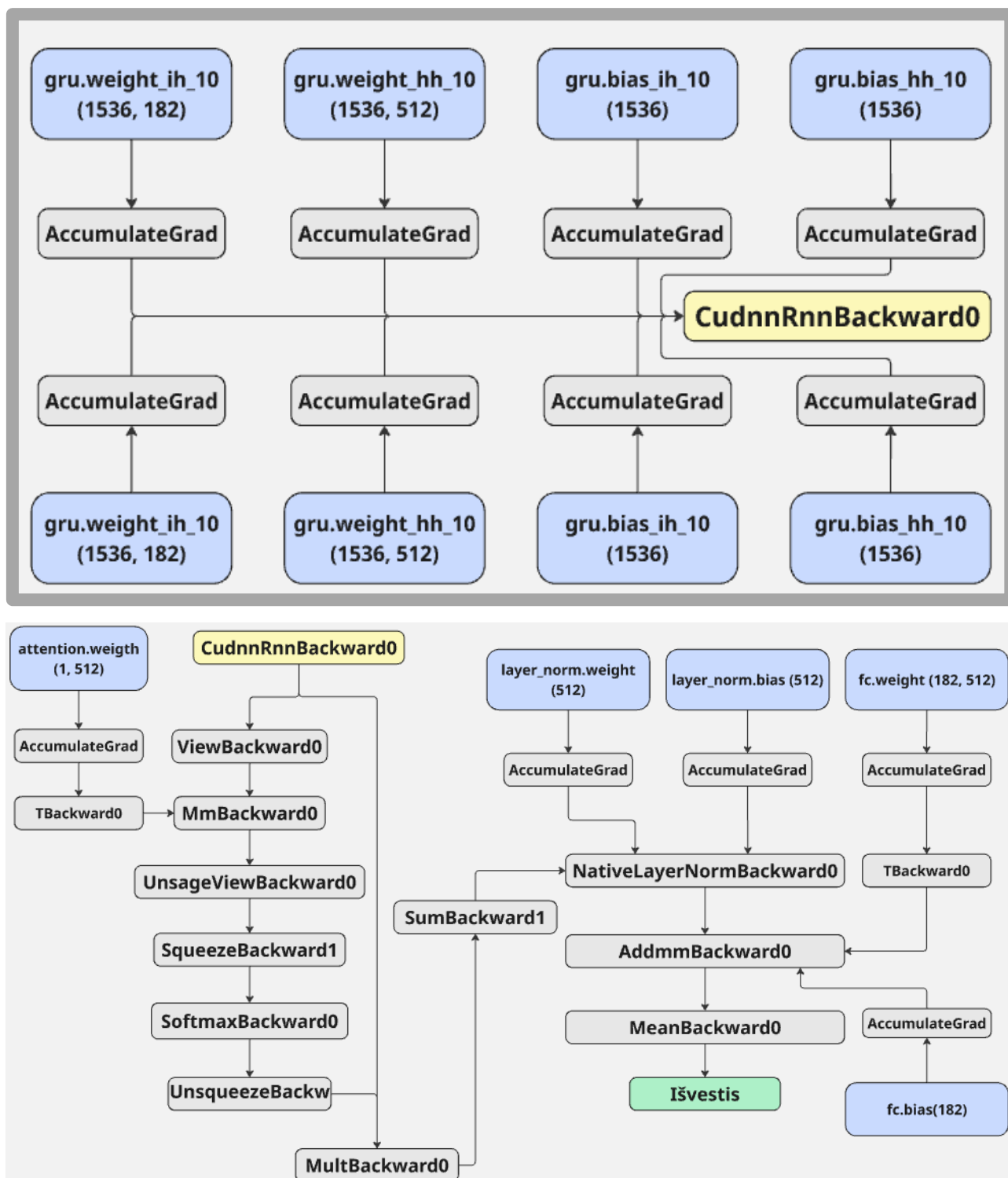
3.8 pav. GRU architektūra

3.2.2.2. GRU architektūra su dėmesio mechanizmu

Igyvendintas modelis su dėmesio mechanizmu. Šis mechanizmas leidžia automatiškai nuspręsti, kurie laiko žingsniai yra svarbiausi, pagerina tinklo gebėjimą naudoti informaciją tiek iš ankstesnių, tiek vėlesnių sekos dalių.

Toliau pateikiama GRU architektūra su dėmesio mechanizmu (žr. 3.9 pav.). Architektūra panaši, esminiai skirtumai:

- Pridedamas papildomas parametrų mazgas – „*attention.weight*“. Šis mazgas pateikia dėmesio mechanizmo svorius.
- „*SoftmaxBackward0*“ – normalizuoja dėmesio įverčius per laiko žingsnius.
- „*MulBackward0*“ – pritaiko dėmesio svorius paslėptoms būsenoms.
- „*SumBackward1*“ – agreguoja paslėptas būsenas, remiantis dėmesio mechanizmu.

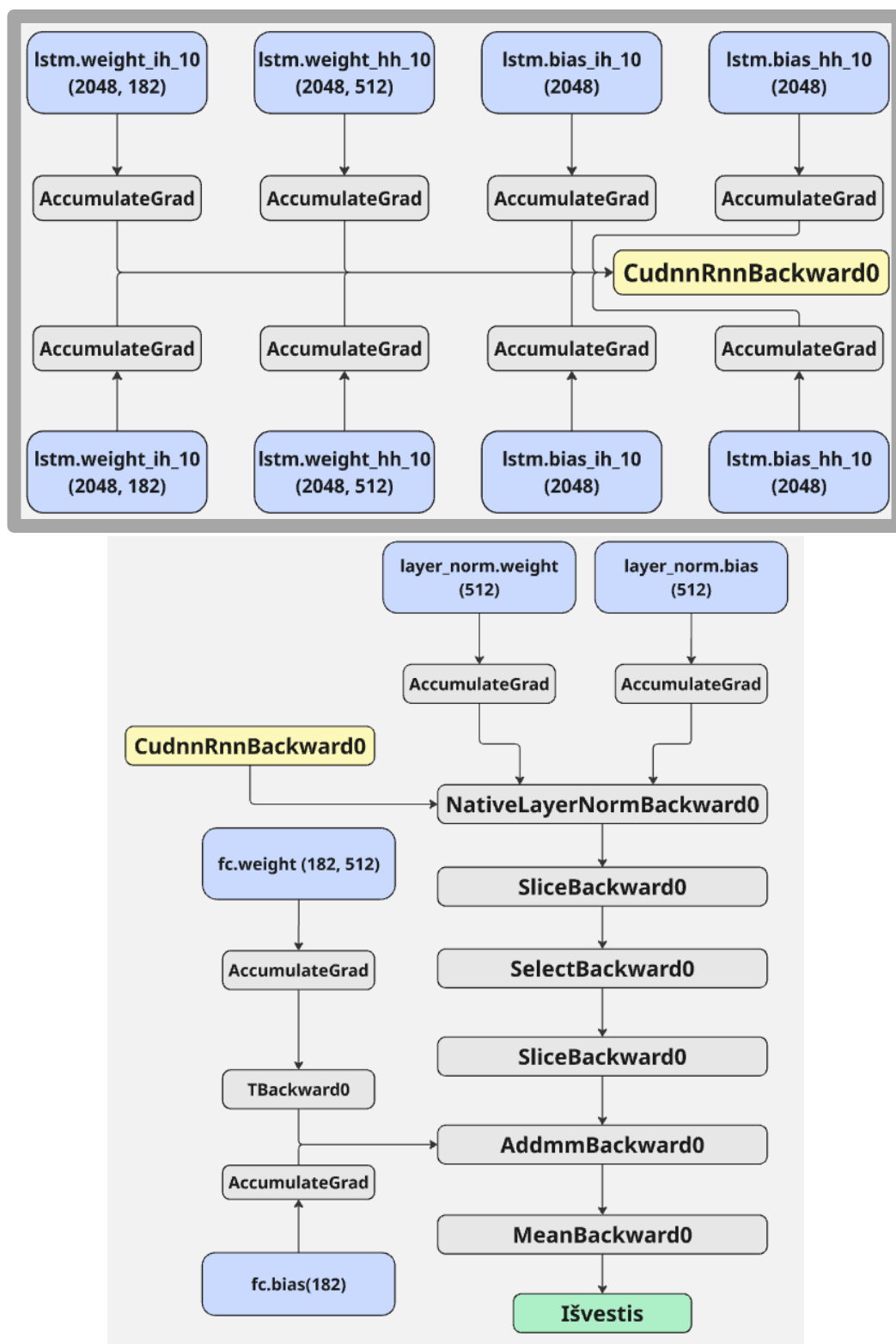


3.9 pav. GRU su dėmesio mechanizmu

3.2.2.3. LSTM architektūra

Lyginant su GRU architektūromis, LSTM yra didesnis parametų kiekiu dėl skirtingo kiekio vartų ir geriau fiksuoja ilgo laiko priklausomybes, tačiau yra lėtesnis.

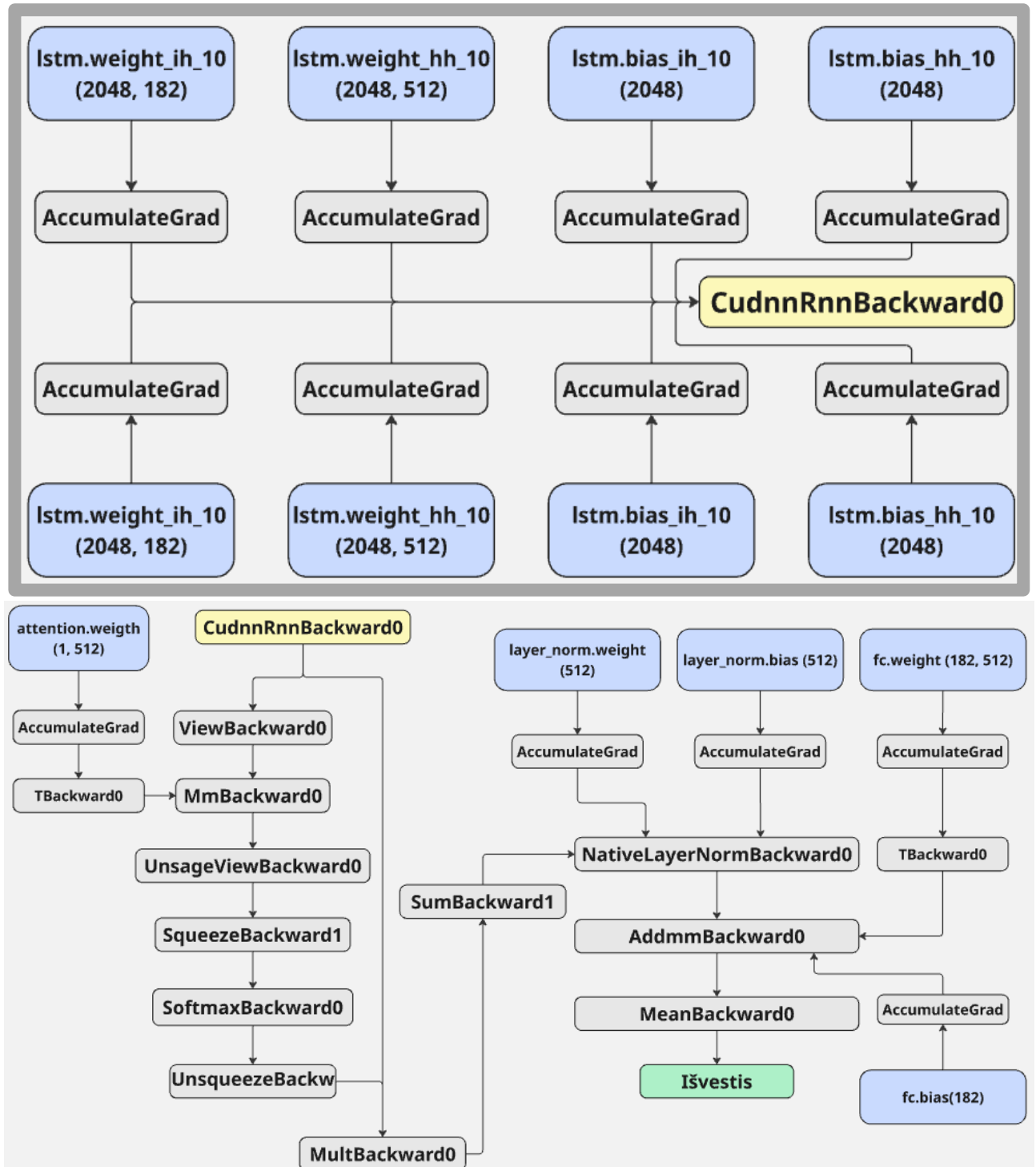
LSTM architektūra pateikiama toliau (žr. 3.10 pav.). Pagrindinis skirtumas architektūros vaizde – vietoje GRU svorių ir poslinkių naudojami LSTM svoriai ir poslinkiai. LSTM svoriai ir poslinkiai yra didesni kiekio prasme.



3.10 pav. LSTM architektūra

3.2.2.4. LSTM architektūra su dėmesio mechanizmu

Įgyvendintas LSTM tinklas su dėmesio mechanizmu. Architektūra pateikiama toliau (žr. 3.11 pav.). Tai kombinacija tarp anksčiau pateikto GRU su dėmesio mechanizmu ir LSTM tinklo. Architektūrose skiriasi svoriai ir poslinkiai.



3.11 pav. LSTM su dėmesio mechanizmu

3.2.3. Papildomas funkcionalumas tikslumui didinti

Įgyvendintas dinamiškas svorių priskyrimas ypatybėms skaičiuojant bendrą klaidos funkciją. Tai leidžia skirti skirtingą svorinį koeficientą pozicijos, pasukimo, linijinio greičio ir kampinio greičio

prognozėms. Kadangi darbe svarbiausia yra pozicija, jai skiriamas svoris lygus 1. Pasukimui skiriamas 0,5 svoris. Linijinį greitį ir kampinį greitį galima apskaičiuoti naudojant prieš tai buvusį ir prognozuojamo kadro poziciją ir pasukimą, todėl nuspręsta šių duomenų neprognozuoti.

Sukurtas papildomas funkcionalumas, leidžiantis priskirti sąnario svarbą (svorį) pagal sąnario indeksą. Kadangi atliekant sąnario pozicijos prognozę svarbiausi sąnariai yra riešas (virtualioje erdvėje tai sekamos rankos pradžia), pirštų galai (virtualioje erdvėje atlieka sąveiką su objektais) ir krumpliai (padeda vizualiai formuoti ranką) – šiems sąnariams skiriami didesni svoriai. Svoriai pateikiami lentelėje (žr. 3.2 lentelė).

3.2 lentelė. Specifinių sąnarių svoriai

Sąnario pavadinimas (anglų k.)	Sąnario (rankos dalies) pavadinimas	Svorio reikšmė
<i>Wrist</i>	Riešas	5
<i>Palm</i>	Delnas	1
<i>Thumb Metacarpal</i>	Nykščio delnakaulio	1
<i>Thumb Proximal</i>	Nykščio pamatinio pirštakaulio	2
<i>Thumb Distal</i>	Nykščio galinio pirštakaulio	1
<i>Thumb Tip</i>	Nykščio galo	3
<i>Index Metacarpal</i>	Smiliaus delnakaulio	1
<i>Index Proximal</i>	Smiliaus pamatinio pirštakaulio	2
<i>Index Intermediate</i>	Smiliaus vidurinio pirštakaulio	1
<i>Index Distal</i>	Smiliaus tolimojo pirštakaulio	1
<i>Index Tip</i>	Smiliaus galo	3
<i>Middle Metacarpal</i>	Didžiojo delnakaulio	1
<i>Middle Proximal</i>	Didžiojo pamatinio pirštakaulio	2
<i>Middle Intermediate</i>	Didžiojo vidurinio pirštakaulio	1
<i>Middle Distal</i>	Didžiojo galinio pirštakaulio	1
<i>Middle Tip</i>	Didžiojo galo	3
<i>Ring Metacarpal</i>	Bevardžio delnakaulio	1
<i>Ring Proximal</i>	Bevardžio pamatinio pirštakaulio	2
<i>Ring Intermediate</i>	Bevardžio vidurinio pirštakaulio	1
<i>Ring Distal</i>	Bevardžio galinio pirštakaulio	1
<i>Ring Tip</i>	Bevardžio galo	3
<i>Little Metacarpal</i>	Mažylio delnakaulio	1
<i>Little Proximal</i>	Mažylio pamatinio pirštakaulio	2
<i>Little Intermediate</i>	Mažylio vidurinio pirštakaulio	1
<i>Little Distal</i>	Mažylio galinio pirštakaulio	1
<i>Little Tip</i>	Mažylio galo	3

3.2.4. Treniravimas

Siekiant palengvinti sistemos modelių treniravimą įgyvendinti papildomi metodai – pateikiami grafikai realiu laiku atliekant treniravimą, naudojami profiliavimo įrankiai, pateikiama tarpinė

išvestis su klaidos vertėmis. Įgyvendintas treniravimo metodas priima keletą parametrų: maksimalų epochų skaičių, pradinį mokymosi greitį, kantrybės parametą (angl. *patience*, kontroliuoja planuotojo mokymosi greičio keitimą, kai validavimo klaidos funkcijos vertė nekintą tam tikrą kiekį epochų), treniravimo sustabdymo kantrybės parametą (kontroliuoja kada išankstiniu būdu sustabdomas treniravimas), svorio nykimas (angl. *weight decay*, optimizatoriaus parametras, kontroliuojantis regularizavimo techniką). Taip pat pateikiami parametrai, kontroliuojantys profiliavimo įrankį (angl. *profiler*) ir grafikų vaizdavimą realiu laiku.

Naudojamas „AdamW“ optimizatorius (angl. *optimizer*) – stochastinis optimizavimo metodas su „Adam“ optimizatoriaus modifikacija. Naudojant šį optimizatorių svorių nykimas yra nebesiejamas su gradientų atnaujinimu. Naudojamas „ReduceLROnPlateau“ planuotojas (angl. *scheduler*) – naudoja kantrybės parametą, nurodantį epochų skaičių su nekintančia klaidos funkcijos validavimo reikšme, kurią peržengus mažinamas mokymosi greitis (dalijamas iš keturių). Išankstinis treniravimo nutraukimas įvyksta kai validavimo klaidos funkcijos reikšmė negerėja tam tikrą kiekį epochų (šis parametras yra didesnis nei kantrybės parametras).

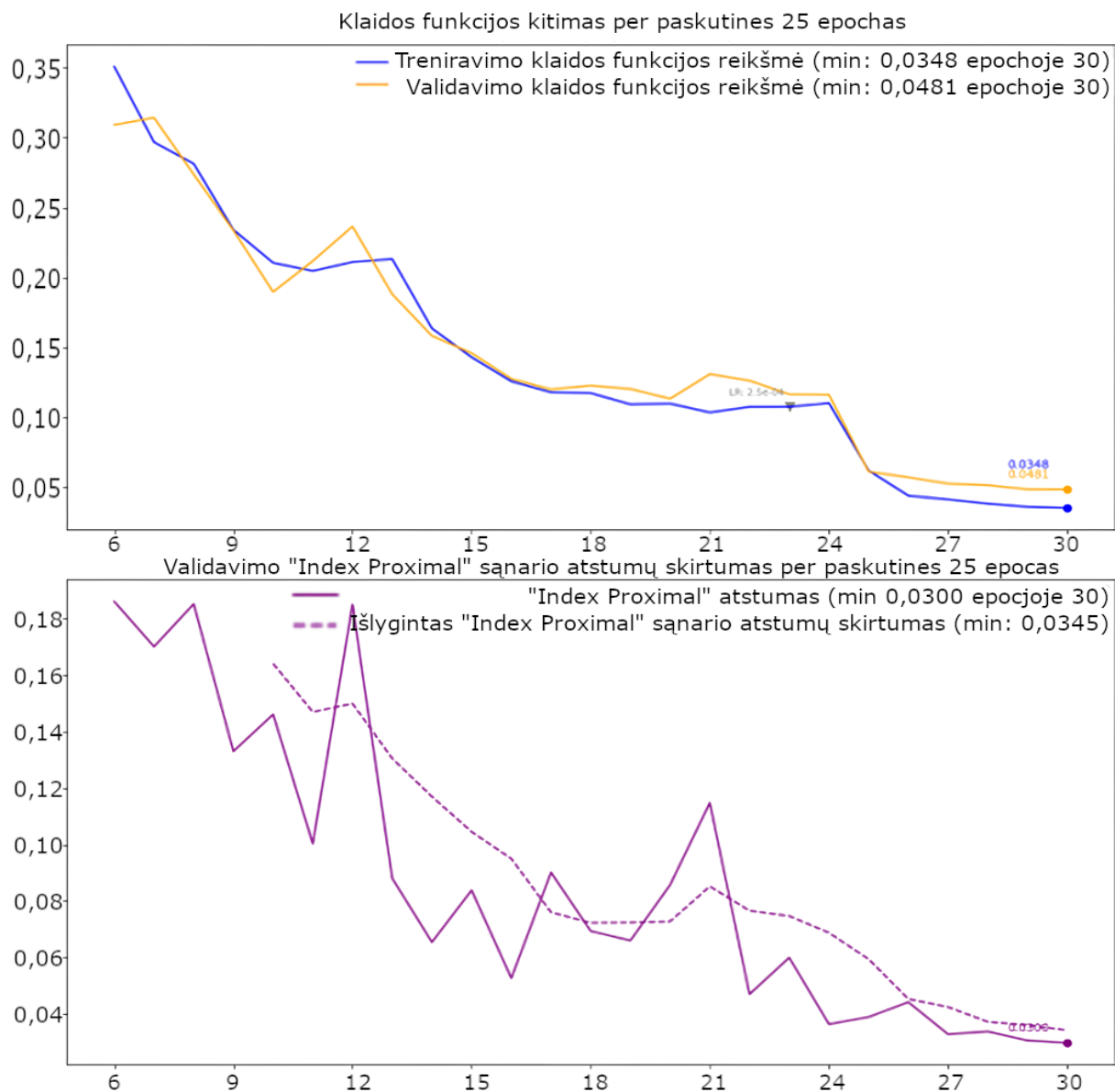
Vykdamas modelių treniravimą pateikiama informacija apie treniruojamo modelio dabartinę epochą, bendrą epochų kiekį, treniravimo klaidos funkcijos vertę, validavimo klaidos funkcijos vertę ir epochos treniravimo laiką. Pateikiama informacija apie pakitusį mokymosi greitį (jei jis epochoje kinta dėl naudojamo planuotojo). Atspausdinama informacija apie paskutinę epochą, jei mokymasis nutrūko dėl išankstinio sustabdymo, kurį sukelia per ilgai negerėjanti validavimo klaidos funkcijos vertė. Treniravimo pabaigoje pateikiamas bendras treniravimo laikas, testavimo rinkinio metrikos ir 20 kadro kairės rankos visų sąnarių pozicijų ir pasukimų realios ir prognozuotos vertės palyginimas.

Įgyvendinta galimybė naudoti profiliavimo įrankį, leidžiantį nustatyti funkcijas, vykdomas treniravimo metu, kurios užtrunka daugiausiai laiko. Šis įrankis padėjo pastebėti keletą spragų, susijusių su duomenų perkėlimu tarp pagrindinio procesoriaus ir grafinio vaizdo procesoriaus ir leido pagreitinti treniravimą. Toliau pateikiamas pavyzdys, kai vaizduojamos pozicijos prognozės metrikos ir pirmų penkių sąnarių tikslinių ir prognozuotų reikšmių palyginimas anglų kalba (žr. 3.12 pav.)

```
-----
Metrics:
-----
RMSE: 0.0108
MAE: 0.0070
MED (Mean Euclidean Distance): 0.0140
Max Euclidean Distance: 0.2289
Temporal Jerk: 0.6550
Wrist MED: 0.0162
Index Tip MED: 0.0142
-----
Joint 0:
Position: Target: [-0.10945977 -0.21211334 0.26412457], Prediction: [-0.10282362 -0.2032463 0.26666406], Euclidean Distance: 0.0114
-----
Joint 1:
Position: Target: [-0.10702064 -0.16662157 0.31542492], Prediction: [-0.10824261 -0.16759856 0.31612074], Euclidean Distance: 0.0017
-----
Joint 2:
Position: Target: [-0.11733329 -0.16188899 0.27551553], Prediction: [-0.10971478 -0.15798819 0.27352932], Euclidean Distance: 0.0088
-----
Joint 3:
Position: Target: [-0.11514692 -0.12886354 0.28225538], Prediction: [-0.10936748 -0.1313832 0.280047 ], Euclidean Distance: 0.0067
-----
Joint 4:
Position: Target: [-0.10662744 -0.09838349 0.29745024], Prediction: [-0.10162468 -0.10615164 0.3041328 ], Euclidean Distance: 0.0114
-----
Joint 5:
Position: Target: [-0.09103227 -0.08347936 0.31119516], Prediction: [-0.08992101 -0.10077573 0.32299903], Euclidean Distance: 0.0210
-----
```

3.12 pav. Pozicijos prognozės metrikos, prognozuotų ir tikslinių reikšmių palyginimas (anglų k.)

Treniruojant modelį realiu laiku galima pateikti grafikus, kuriuose vaizduojamos klaidos funkcijų kitimas, riešo, smiliaus pamatinio pirštakaulio ir smiliaus galo sąnarių validavimo rinkinio prognozuojamos ir realios reikšmės atstumų skirtumas ir sulyginta (naudojant slenkantį vidurkį per 5 kadrus) vertė. Toliau pateikiamas pavyzdys vykdant treniravimą pateikiant 25 paskutines epochas (žr. 3.14 pav., **Error! Reference source not found.**).



3.13 pav. Klaidos funkcijų ir sąnarių atstumų kitimas per paskutines 25 epochas treniravimo metu.



3.14 pav. Sąnarių atstumų kitimas per paskutines 25 epochas treniravimo metu.

Šie grafikai leidžia matyti kitimus, kurie gali padėti nuspręsti anksčiau laiko sustabdyti treniravimą arba modifikuoti svorius sąnariams.

3.2.5. Prognozuotų pozicijų ir pasukimų erdvėje vizualizavimas naudojant „Unity“

„Unity“ žaidimų kūrimo įrankyje realizuota prognozių vizualizavimo sistema siekiant vizualiai įvertinti prognozių tikslumą ir greitaveiką.

Prognozavimo sistema, paleidus žaidimą, sukuria fone veikiančią begalinį ciklą su naudojamu *Python* visuotinio vertėjo užraktu (angl. *global interpreter lock*), kuris leidžia kviesti programinį kodą, įgyvendintą *Python* programavimo kalba. Naudojami įvykiai (angl. *event*) aptikti, kada reikalinga prognozė. Eksperimentavimo tikslais prognozė kviečiama kiekvieną kartą atnaujinant sąnarius naudojant VR posistemę – tai leidžia vienu metu pateikti VR sistemos aptiktą sąnarį ir prognozuotą sąnarį. Atliekant eksperimentus pastebėta, kad rankų sekimas dažnu atveju nutrūksta vienam arba keliems kadrams. Prognozė atliekama tik vienam sąnariui į ateitį. Galima daryti prielaidą, kad vykdomas eksperimentavimas atitinka realius scenarijus, kai rankų sekimas trūkinėja dėl išorės veiksnių.

Kadangi rekurentiniai neuroniniai tinklai naudoja paskutinių kadro seką, o ne paskutinį turimą kadrą, reikia kaupti L paskutinių kadro prognozei atlikti. Tai nuspręsta įgyvendinti *Python* posistemėje, siekiant sumažinti siunčiamų duomenų kiekį. Šiuo atveju komunikuojant tarp *C#* ir *Python* reikia perduoti tik vieną paskutinį kadrą o ne paskutinius L kadro – skatinama greitimeika. Naudojama ciklinė eilė – saugojami paskutiniai 15 kadro, pridėdant naują kadrą seniausias kadras yra pašalinamas.

„Unity“ rankų sąnarių ypatybės saugojamos naudojant sukurta duomenų formatą. Prieš perduodant duomenis į *Python* posistemę turimi rankų duomenys yra transformuojami į vienmatį masyvą. Vienmatis masyvas paverčiamas į formatą, tinkamą prognozėms. Naudojant grafinį vaizdo procesorių atliekama prognozė, kuri yra grąžinama į „Unity“ posistemę tolimesniam apdorojimui. „Unity“ posistemėje šis vienmatis masyvas yra serializuojamas į anksčiau minėtą duomenų formatą ir pateikiamas vizualizavimo posistemėi.

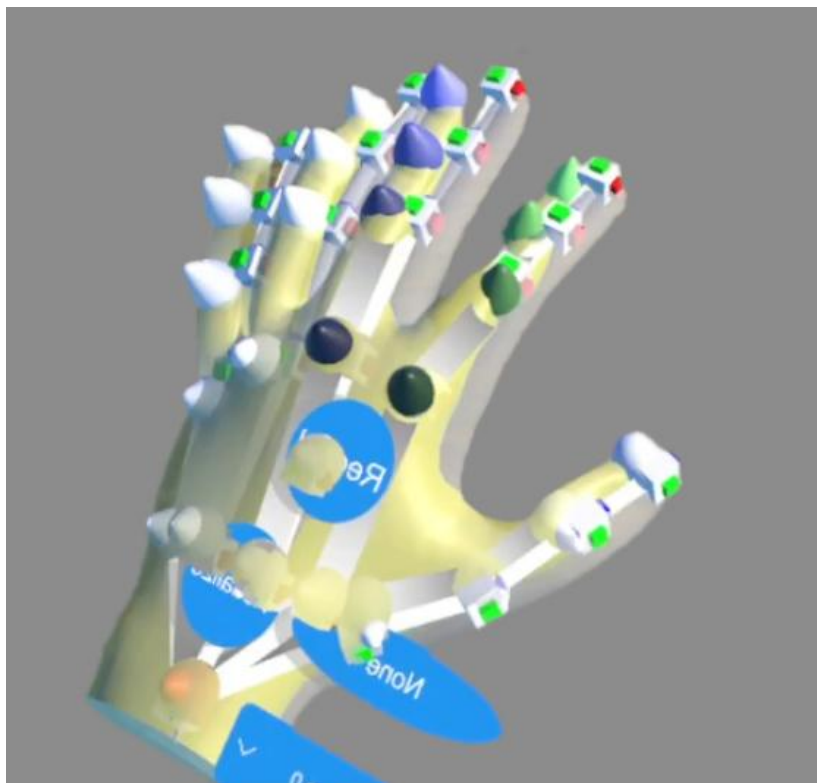
Vizualizavimo sistema yra atsakinga už gautų prognozių pateikimą virtualioje erdvėje. Naudojant rankų sąnarių hierarchiją sukuriamos linijos, vaizduojančios sąnarius jungiančius kaulus. Naudojamas įvykis, kuris yra kviečiamas kai prognozė yra atlikta. Prognozės duomenys (pozicijos, pasukimai) yra pateikiami kaip virtualios rankos erdvėje.

Vizualizavimo sistema pateikia papildomą funkcionalumą, susijusį su rankų sąnarių prognozių vizualizavimu. Jei sąnariai prognozuojami naudojant veidrodinį rankos atspindį (t. y. atvejis, kai treniruojamas vienas modelis su dviem rankomis, kai dešinė ranka yra paverčiama kaire naudojant transformacijas) prognozė turi būti transformuojama naudojant veidrodinį atspindį prieš pateikiant dešinę ranką erdvėje. Taip pat pateikiama galimybė naudoti papildoma lyginimą su slenkančio vidurkio įgyvendinimu. Ši funkcija leidžia sumažinti drebulį, atsirandantį dėl prognozių netikslumų, tačiau kadangi naudojamos paskutinių kadro pozicijos, nauja gaunama pozicija gali atsilikti nuo naujos realios pozicijos ir pasukimo. Toliau pateikiamas vizualizuojamos prognozuotos rankos pavyzdys kai matoma tik prognozuota ranka (žr. 3.15 pav.). Rankoje skirtingomis spalvomis pažymėti smiliaus ir didžiojo piršto sąnariai (žalia ir mėlyna spalva) ir riešo sąnarys (raudona spalva).



3.15 pav. Vizualizuota prognozuota ranka

Paveikslėlyje skirtingomis spalvomis nuspalvinti smiliaus ir didžiojo piršto sąnariai ir riešo sąnarys. Testuojant sistemą vienu metu pateikiama ir sekama ranka, ir bandomas prognozuoti kadrą į ateitį. Spalvinė informacija leidžia lengviau interpretuoti prognozuotą ranką. Žemiau pateikiamas pavyzdys, kai vizualizuojama prognozuota ir sekama rankos (žr. 3.16 pav.).



3.16 pav. Vizualizuota sekama (pilka) ranka ir prognozuota (geltona) ranka, kai atliekamas judesys į kairę

4. Rankų sąnarių pozicijų ir pasukimo prognozių eksperimentai ir rezultatai

Šioje dalyje pateikiama informacija apie atliktus eksperimentus, bandant išsiaiškinti geriausias modelių, naudojamų duomenų ir parametrų kombinacijas. Parametrai, kurie gali būti keičiami eksperimentų metu:

- Įvesties duomenų formatas
 - Įrenginio erdvė (sąnarių pozicijos ir pasukimas saugojamas pagal atstumą ir pasukimą nuo virtualiosios realybės sistemos centrinio taško)
 - Riešo lokali sąnarių erdvė – riešo sąnario pozicija ir pasukimas fiksuojamas nuo VR sistemos centrinio taško, kitų sąnarių pozicijos ir pasukimas fiksuojamas kaip lokalus atstumas nuo riešo sąnario
 - Hierarchinė lokali sąnarių erdvė (pagal vizualizacijos hierarchiją naudojama kiekvieno sąnario lokali pozicija ir pasukimas, riešo sąnario pozicija ir pasukimas sekama nuo virtualiosios realybės sistemos centrinio taško)
 - Įrenginio erdvė su veidrodiniu atspindžiu (dešinė ranka transformuojama į kairę, gaunamas dvigubai ilgesnis duomenų rinkinys)
- Modelio architektūros
 - GRU
 - GRU su dėmesio mechanizmu
 - LSTM
 - LSTM su dėmesio mechanizmu
 - Visoms architektūroms gali kisti paslėptų reikšmių kiekis ir sluoksnių kiekis
- Įvesties ypatybės
 - Naudojama tik pozicija (jei prognozuojama tik pozicija)
 - Naudojama pozicija ir pasukimas
 - Naudojama visos ypatybės
- Išvesties ypatybės
 - Prognozuojama tik pozicija
 - Prognozuojamas tik pasukimas
 - Prognozuojama pozicija ir pasukimas kartu
- Papildomi metodai
 - Elementų kiekio kitimas (angl. *batch size*)
 - Sviurių taikymas specifiniams sąnariams (riešui, pirštų galams, krumpliams)
 - Sviurių nykimo naudojimas

Eksperimentų lentelėse ir aprašuose pateikiama informacija apie naudojamą duomenų erdvę (įrenginio, lokali riešo, lokali hierarchinė, veidrodinė), įvesties ypatybės, išvesties ypatybės, architektūra (GRU, GRU su dėmesio mechanizmu, LSTM, LSTM su dėmesio mechanizmu), duomenų kiekį (angl. *batch size*), pozicijos klaidos funkcija (čia „MED“ – vidutinio Euklido atstumo (*mean Euclidean distance*)), „FingerHierarchy“ – sąnario atstumo pirštuose, kai skaičius skliaustuose nurodo sviurį), pasukimo klaidos funkcija (čia „QuaternionGeodesic“ – kvaterniono geodezinis nuostolis, „QuaternionAngle“ – kvaterniono kampinis nuostolis, kai skaičius skliaustuose nurodo sviurį), sąnarių sviurių naudojimas ir sviurių nykimo (angl. *weight decay*) naudojimas. Lentelėse paryškintas ir pabrauktas tekstas rodo geriausią gautą metriką. Tik pabrauktas tekstas rodo antrą geriausią rezultatą. Pateikiami lyginamieji grafikai.

Eksperimentų metu atliktų prognozių kokybė įvertinama naudojant šaknies iš vidutinės kvadratinės paklaidos (angl. RMSE, *root mean square error*), vidutinės absoliutinės paklaidos (angl. MAE, *mean absolute error*), vidutinio Euklido atstumo (angl. MED, *mean Euclidean distance*), maksimalaus Euklido atstumo (MaxED) antro maksimalaus Euklido atstumo (Antras MaxED), vidutinio absoliutinio laikinio skirtumo metrikos. Pateikiamos ir specifinių sąnarių (riešo ir smiliaus piršto galo) vidutinio Euklido atstumo metrikos. Pasukimo prognozėms įvertinti naudojama kvaterniono kampinio skirtumo ir geodezinės paklaidos. Metrikų aprašai pateikiami 2.6 poskyryje.

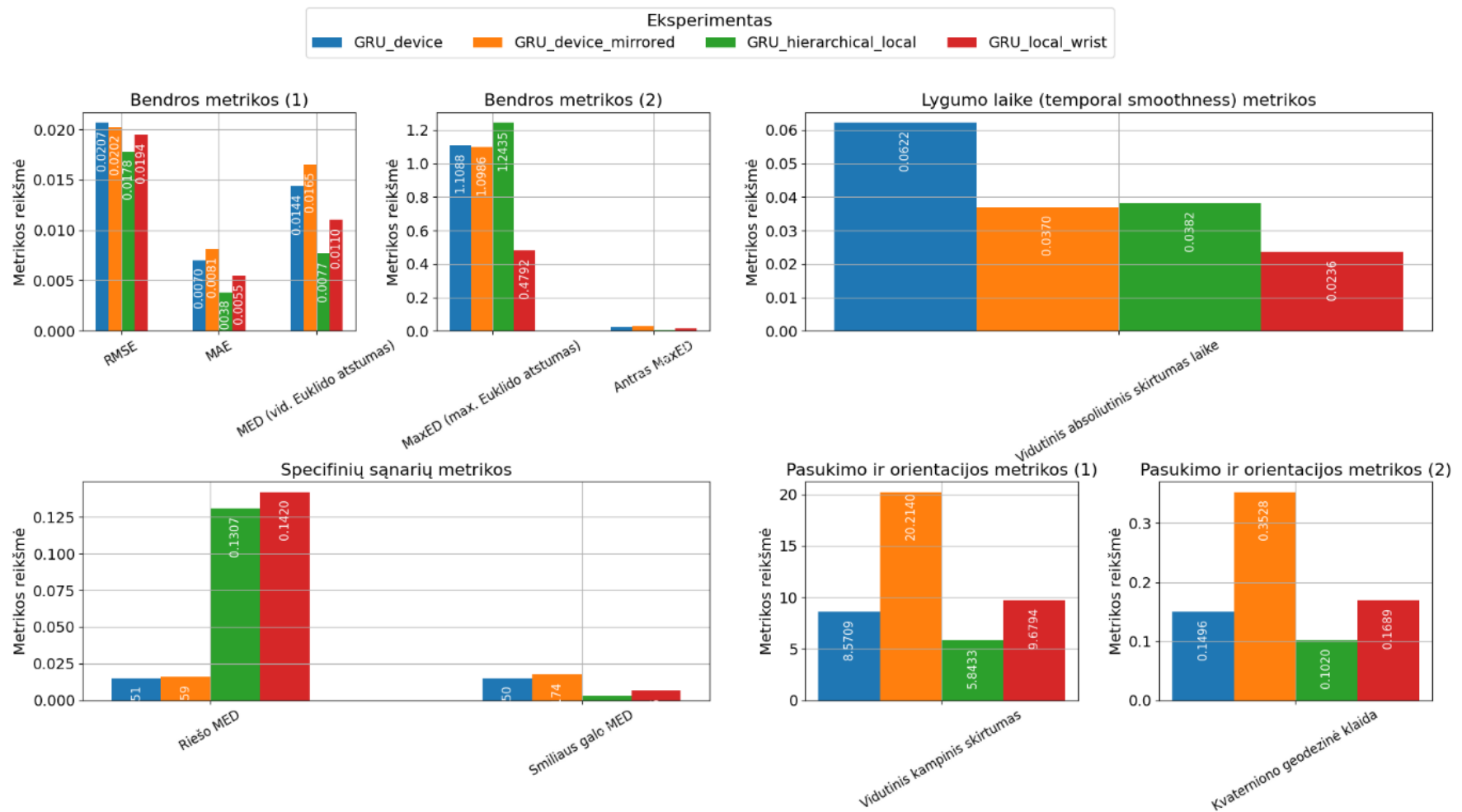
Eksperimentuojant taikomas nuoseklus parametrų optimizavimas. Iteratyviai keičiamas vienas parametras, kai kiti parametrai lieka fiksuoti. Kiekvieno eksperimento metu nustatomas geriausias atitinkamas parametras, kuris yra pernaudojamas tolimesniuose eksperimentuose. Išskirtiniais atvejais pasirinktas sprendimas pateikiamas specifinio eksperimento aprašuose. Eksperimentuose naudojamas 26000 kadrų duomenų rinkinys, kuriame fiksuojami įvairūs virtualiai realybei būdingi veiksmai – judinamos rankos keliose padėtyse, atliekami įprasti virtualių objektų paėmimo, dydžio keitimo (tempimo) judesiai, simuliuojama sąveika su naudotojo sąsaja, bandoma užfiksuoti veiksmus lyg rankoje būtų laikoma lazda ar kardas arba kelių tipų ginklai (siekiama sukurti bendrą VR tinkančią prognozių sistemą, tinkančią ir žaidimams). Iš 26000 kadrų duomenų rinkinio atrenkami 10000 elementų parenkant atsitiktinę imtį iš skirtingų veiksmų įrašų. Eksperimentų metu užtikrinamas pakartojamumas – visuose eksperimentuose naudojama ta pati imtis. Mažesnis duomenų rinkinys naudojamas siekiant sumažinti reikalingą treniravimo laiką, bet užtikrinant, kad duomenys liktų bendri (angl. *general*).

4.1. Eksperimentai keičiant naudojamą duomenų formatą

Eksperimente naudojamas LSTM ir GRU modeliai (512 paslėptų, 2 sluoksniai), treniravimas vykdomas kuriant atskirus modelius kiekvienai rankai (šiuo atveju treniruojama tik kairės rankos modelis, nes dešinės rankos modelis turi panašias metrikas). Eksperimento metu bandoma iširti, kuris duomenų formatas yra tinkamiausias pozicijoms ir pasukimui prognozuoti. Pateiktuose eksperimentuose naudojama viena ranka, visos įvesties ypatybės, prognozuojama pozicija ir pasukimas, 512 elementų duomenų kiekis (*batch size*), sujungta vidutinio Euklido atstumo ir hierarchinė pirštų sąnarių atstumo klaidos funkcijos, kampinio kvaterniono klaidos funkcija pasukimui, naudojami specifinių sąnarių svoriai ir nenaudojamas svorių nykimas. Toliau pateikiamos erdvės kitimų eksperimentų lentelės ir lyginamieji grafikai (žr. 4.1 lentelė, 4.1 pav.).

4.1 lentelė. Erdvės kitimo eksperimentai naudojant GRU architektūrą

Nr.	Erdvė	Architektūra	RMSE	MAE	MED	MaxED	Antras MaxED	Riešo MED	Smiliaus galo MED	Vidutinis kampinis skirtumas	Kvaterniono geodezinė klaida	Vidutinis absoliutinis skirtumas	Treniravimo laikas (s)	Prognozės vid. laikas (s)
1.	Įrenginio	GRU (512, 2)	0.02066	0.00703	0.01438	1.10877	0.02374	0.01505	0.01499	<u>8.57094</u>	<u>0.14959</u>	0.06219	31.14	0.001057
2.	Įrenginio veidrodinė	GRU (512, 2)	0.02023	0.00812	0.01654	<u>1.09861</u>	0.02848	0.01585	0.01744	20.21400	0.35280	<u>0.03695</u>	43.02	0.001071
3.	Hierarchinė lokali	GRU (512, 2)	<u>0.01781</u>	<u>0.00383</u>	<u>0.00768</u>	1.24347	<u>0.00546</u>	<u>0.13075</u>	<u>0.00294</u>	<u>5.84333</u>	<u>0.10199</u>	0.03818	29.41	0.001104
4.	Lokali riešo	GRU (512, 2)	<u>0.01943</u>	<u>0.00548</u>	<u>0.01101</u>	<u>0.47917</u>	<u>0.01453</u>	<u>0.14196</u>	<u>0.00655</u>	9.67942	0.16894	<u>0.02358</u>	22.02	0.001068

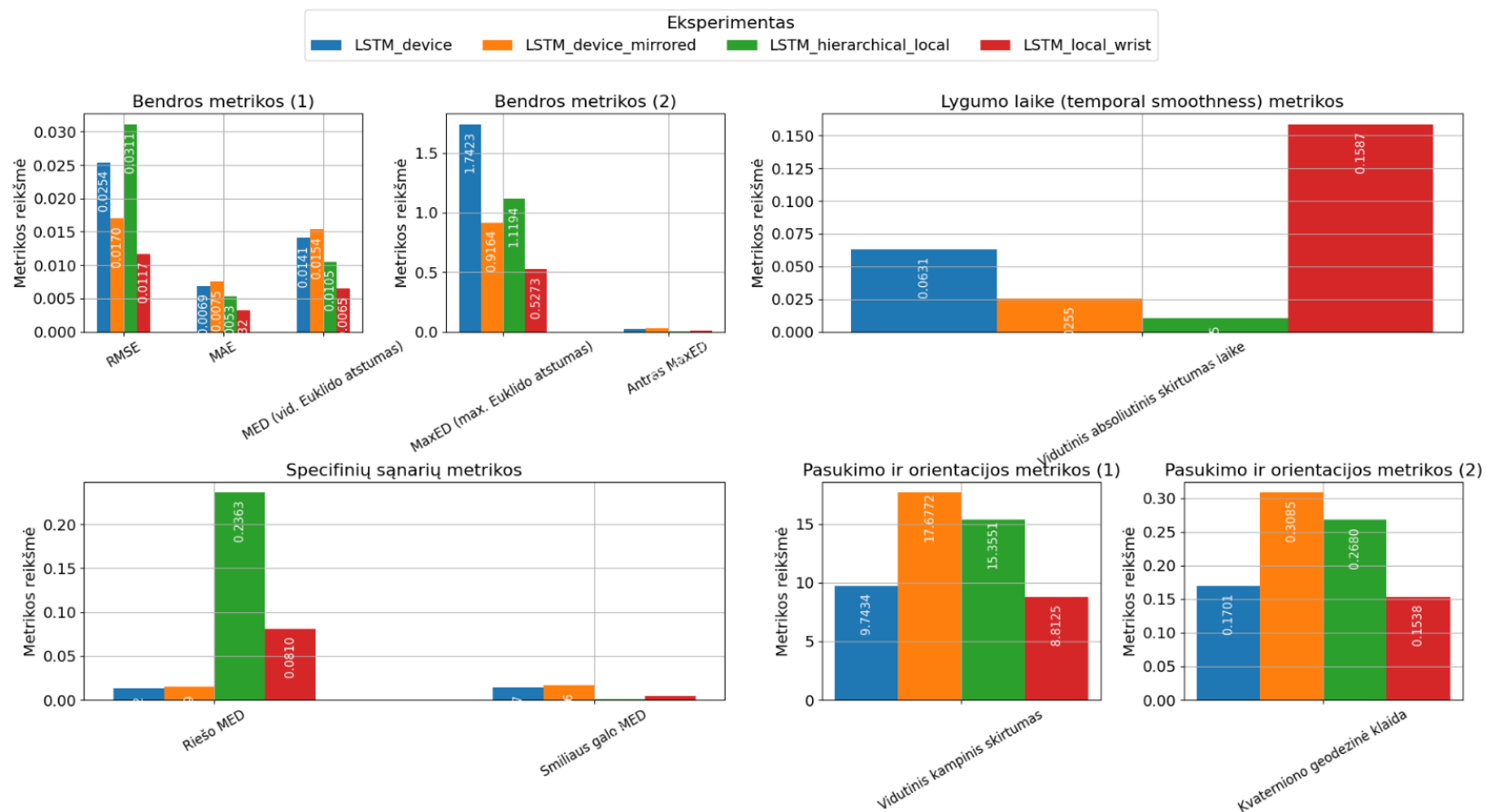


4.1 pav. Erdvės kitimo eksperimentai naudojant GRU architektūrą

GRU architektūros atveju geriausi rezultatai pastebimi naudojant lokalias erdves. LSTM architektūros eksperimentai pateikiami toliau (žr. 4.2 lentelė, 4.2 pav.).

4.2 lentelė. Erdvės kitimo eksperimentai kai naudojant LSTM architektūrą

Nr.	Erdvė	Architektūra	RMSE	MAE	MED	MaxED	Antras MaxED	Riešo MED	Smiliaus galo MED	Vidutinis kampinis skirtumas	Kvaterniono geodezinė klaida	Vidutinis absoliutinis skirtumas	Treniravimo laikas (s)	Prognozės vid. laikas (s)
5.	Įrenginio	LSTM (512, 2)	0.02537	0.00689	0.01409	1.74228	<u>0.02323</u>	0.01324	0.01469	<u>9.74340</u>	<u>0.17005</u>	0.06314	35.12	0.001192
6.	Įrenginio veidrodinė	LSTM (512, 2)	<u>0.01699</u>	0.00754	0.01535	<u>0.91642</u>	0.02664	<u>0.01485</u>	0.01662	17.67716	0.30852	0.02550	36.43	0.001070
7.	Hierarchinė lokali	LSTM (512, 2)	0.03113	<u>0.00528</u>	<u>0.01052</u>	1.11942	0.00374	0.23627	0.00137	15.35509	0.26800	0.01046	15.60	0.001141
8.	Lokali riešo	LSTM (512, 2)	0.01166	0.00322	0.00650	0.52726	0.01121	0.08099	<u>0.00461</u>	8.81250	0.15381	0.15873	55.31	0.001161



4.2 pav. Erdvės kitimo eksperimentai naudojant LSTM architektūrą

LSTM architektūros atveju geriausi rezultatai taip pat pastebimi naudojant lokalias erdves. Tačiau tolimesniuose eksperimentuose nuspręsta šių erdvių nenaudoti, kadangi, nors šios metrikos yra mažiausios, skiriasi įvesties (ir metrikų) mastelis. Realiau atveju (pateikiant sekimą su vizualizavimu) įrenginio ir veidrodinė įrenginio erdvė veikia geriau. Kadangi prognozės vidutinis laikas yra pakankamai mažas, nuspręsta veidrodinės erdvės nenaudoti, kadangi gaunamos šiek tiek prastesnės metrikos ir reikalingas papildomas įvesties ir išvesties transformavimas, kuris gali sumažinti galutinį tikslumą dėl skaičių apvalinimo paklaidų.

4.2. Eksperimentai keičiant architektūros parametrus

Eksperimentuose kinta neuroninių tinklų dydis keičiant neuronų kiekį ir sluoksnių skaičių. Išbandomi GRU ir LSTM tinklai su dėmesio mechanizmu. Pateiktuose eksperimentuose naudojama viena ranka, įrenginio erdvė, visos įvesties ypatybės, prognozuojama pozicija ir pasukimas, 512 elementų duomenų kiekis, sujungta vidutinio Euklido atstumo ir hierarchinė pirštų sąnarių atstumo klaidos funkcijos, kampinio kvaterniono klaidos funkcija pasukimui, naudojami specifinių sąnarių svoriai ir nenaudojamas svorių nykimas. Eksperimentų rezultatai pateikiami lentelėse ir grafikuose (žr. 4.3 lentelė-4.5 lenteles, 4.3-4.5 pav.). Atliekant eksperimentus pastebėta, tai, kad skirtingos architektūros turi skirtingas optimalias metrikas. Siekiant parinkti optimalią architektūrą sekantiems eksperimentams, apibūdinamos bendros metrikos įvertis, sujungiantis pozicijos ir pasukimo metrikas su svoriais. Metrikos normalizuojamos ir invertuojamos, formulės pateikiamos toliau:

$$Norm_x = \frac{x - \min(x)}{\max(x) - \min(x)}; \quad (24)$$

čia x – metrika, $\min(x)$ – minimali metrikos vertė, $\max(x)$ – maksimali metrikos vertė.

$$Ivertis_x = 1 - Norm_x; \quad (25)$$

čia x – metrika, $Norm_x$ – normalizuota metrika.

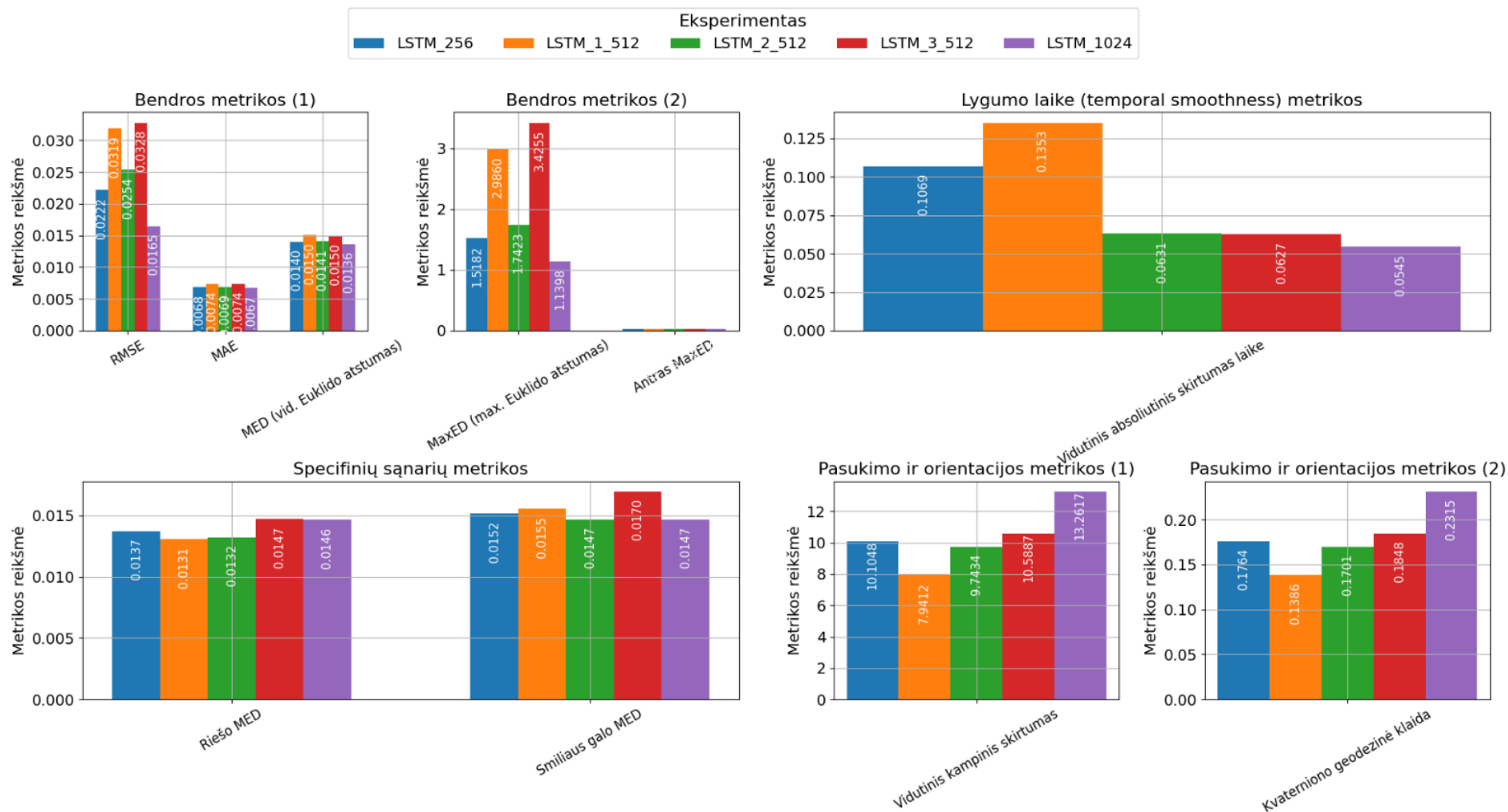
$$Pozicijos\ ivertis = 0,25\ Ivertis_{MED} + 0,2\ Ivertis_{riešo\ MED} + Ivertis_{antras\ MaxED} + 0,15\ Ivertis_{VAS} + 0,1\ Ivertis_{smiliaus\ MED} + 0,05\ Ivertis_{RMSE} + 0,03\ Ivertis_{MAE} + 0,02\ Ivertis_{MaxED}; \quad (26)$$

$$Orientacijos\ ivertis = 0,5\ Ivertis_{vid.\ kamp.\ skirt.} + 0,5\ Ivertis_{kvat.\ geod.\ kl.}; \quad (27)$$

$$Bendras\ ivertis = 0,7\ Pozicijos\ ivertis + 0,3\ Orientacijos\ ivertis; \quad (28)$$

4.3 lentelė. Architektūros dydžio kitimo eksperimentai naudojant LSTM architektūrą

Nr.	Architektūra	RMSE	MAE	MED	MaxED	Antras MaxED	Riešo MED	Smiliaus galo MED	Vidutinis kampinis skirtumas	Kvaterniono geodezinė klaida	Vidutinis absoliutinis skirtumas	Treniravimo laikas (s)	Prognozės vid. laikas (s)
9.	LSTM (256, 2)	<u>0.02220</u>	<u>0.00684</u>	<u>0.01398</u>	<u>1.51820</u>	<u>0.02303</u>	0.01374	0.01517	10.10484	0.17636	0.10686	22.34	0.001131
10.	LSTM (512, 1)	0.03187	0.00738	0.01503	2.98603	0.02376	<u>0.01307</u>	0.01554	<u>7.94122</u>	<u>0.13860</u>	0.13529	39.08	0.0000650
11.	LSTM (512, 2)	0.02537	0.00689	0.01409	1.74228	<u>0.02323</u>	<u>0.01324</u>	<u>0.01469</u>	<u>9.74340</u>	<u>0.17005</u>	0.06314	35.12	0.001192
12.	LSTM (512, 3)	0.03280	0.00738	0.01495	3.42554	0.02489	0.01470	0.01696	10.58872	0.18481	<u>0.06270</u>	44.43	0.001720
13.	LSTM (1024, 2)	<u>0.01646</u>	<u>0.00672</u>	<u>0.01364</u>	<u>1.13984</u>	0.02354	0.01464	<u>0.01468</u>	13.26167	0.23146	<u>0.05453</u>	116.14	0.003423

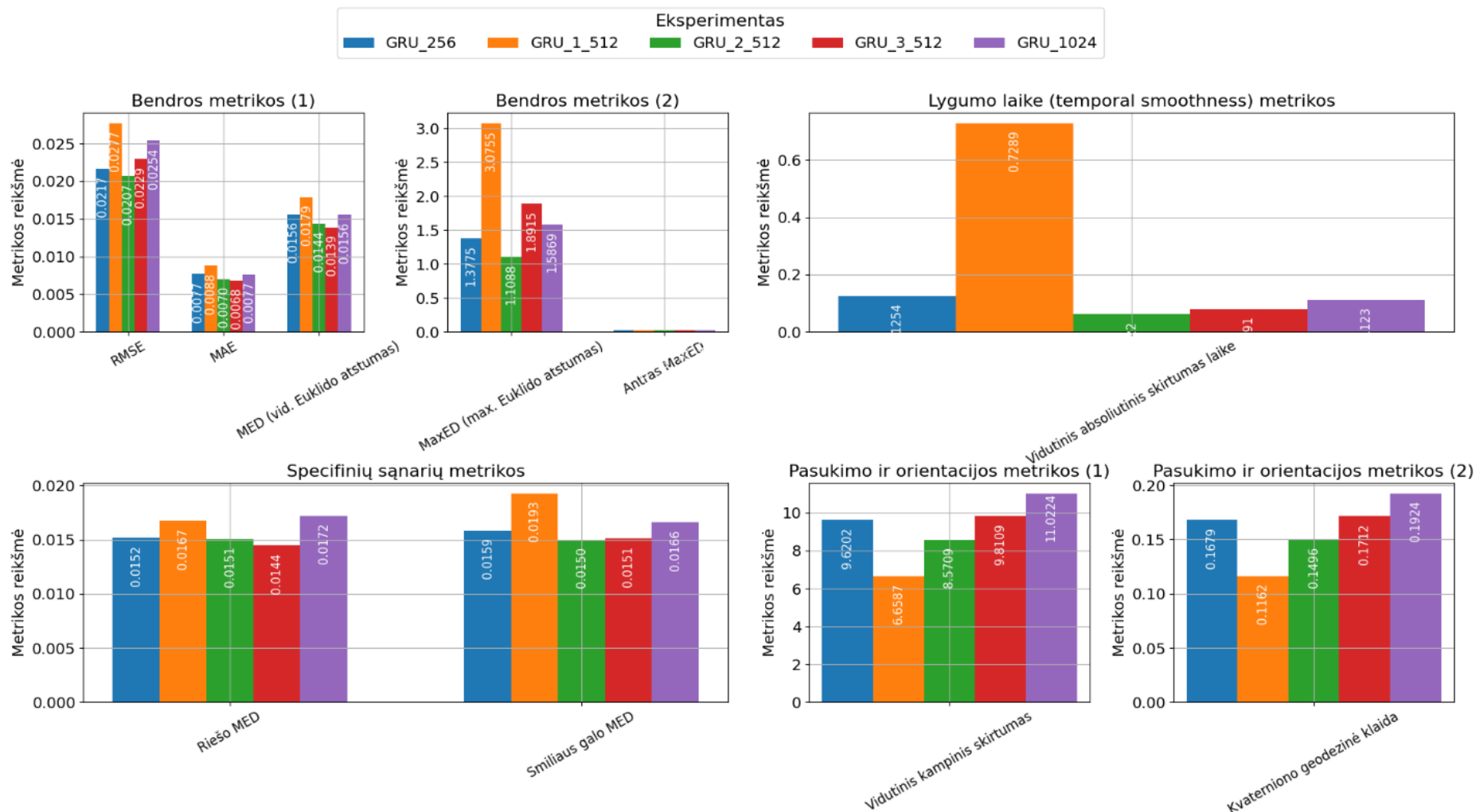


4.3 pav. Architektūros dydžio kitimo eksperimentai naudojant LSTM architektūrą

Naudojant LSTM architektūrą optimalios metrikos pasiekiamos naudojant 1024 paslėptus vienetus ir 2 sluoksnius, tačiau skaičiavimai užtrunka ilgiausiai. Prognozuojant orientaciją geriausios metrikos pasiekiamos naudojant LSTM su 512 paslėptų vienetų ir 1 sluoksniu.

4.4 lentelė. Architektūros dydžio kitimo eksperimentai naudojant GRU architektūrą

Nr.	Architektūra	RMSE	MAE	MED	MaxED	Antras MaxED	Riešo MED	Smiliaus galo MED	Vidutinis kampinis skirtumas	Kvaterniono geodezinė klaida	Vidutinis absoliutinis skirtumas	Treniravimo laikas (s)	Prognozės vid. laikas (s)
14.	GRU (256, 2)	<u>0.02168</u>	0.00767	0.01560	<u>1.37753</u>	0.02559	0.01516	0.01586	9.62018	0.16790	0.12537	24.87	0.001053
15.	GRU (512, 1)	0.02771	0.00882	0.01787	3.07554	0.02955	0.01674	0.01926	<u>6.65871</u>	<u>0.11622</u>	0.72886	25.25	0.000645
16.	GRU (512, 2)	<u>0.02066</u>	<u>0.00703</u>	<u>0.01438</u>	<u>1.10877</u>	<u>0.02374</u>	<u>0.01505</u>	<u>0.01499</u>	<u>8.57094</u>	<u>0.14959</u>	<u>0.06219</u>	31.14	0.001057
17.	GRU (512, 3)	0.02291	<u>0.00679</u>	<u>0.01386</u>	1.89148	<u>0.02361</u>	<u>0.01444</u>	<u>0.01508</u>	9.81086	0.17123	<u>0.07914</u>	38.60	0.001595
18.	GRU (1024, 2)	0.02537	0.00765	0.01555	1.58695	0.02610	0.01719	0.01660	11.02240	0.19238	0.11231	84.35	0.002723

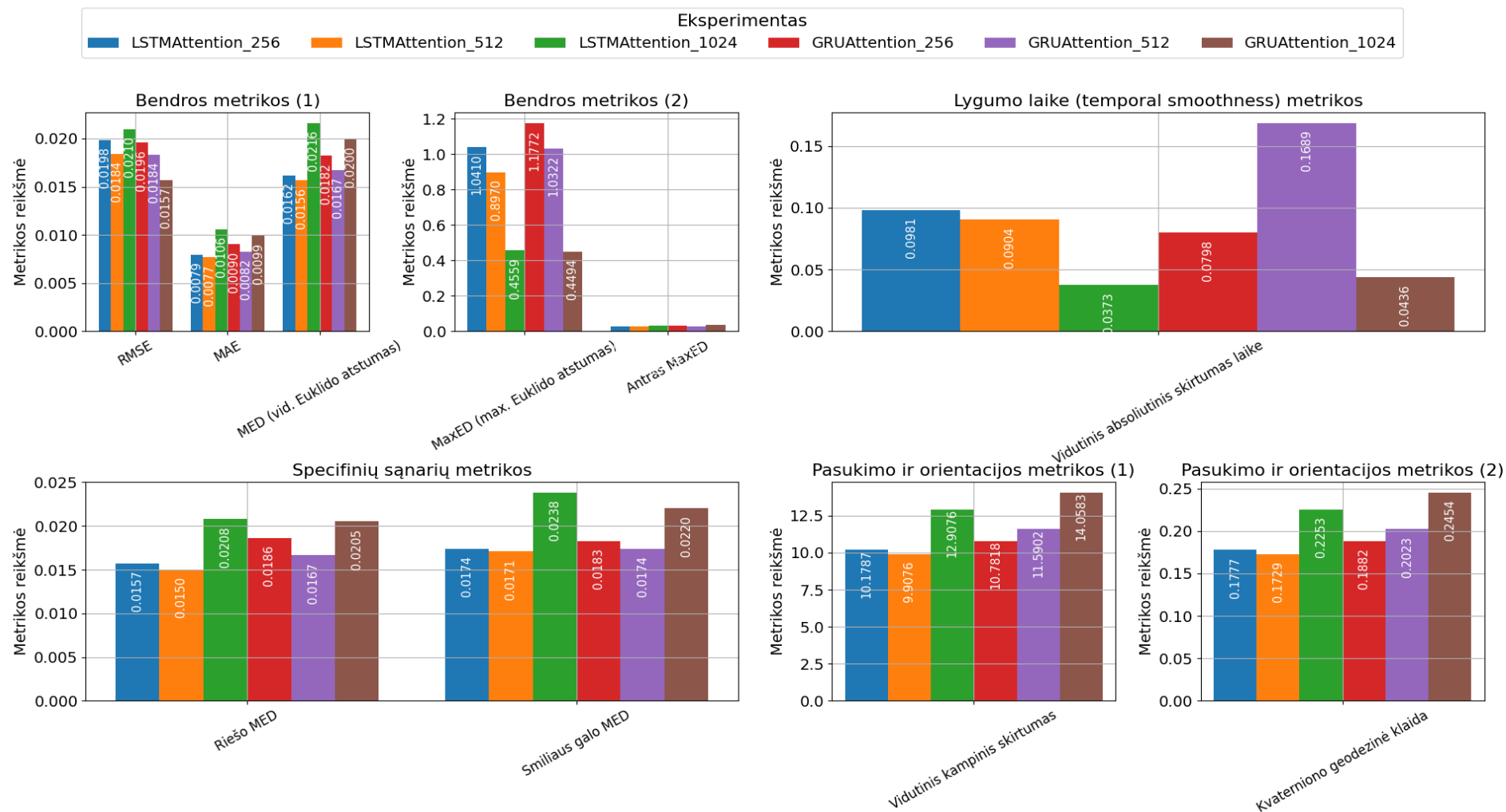


4.4 pav. Architektūros dydžio kitimo eksperimentai naudojant GRU architektūrą

Naudojant GRU architektūrą optimalios metrikos gaunamos su 512 paslėptų vienetų ir 2 arba 3 sluoksnius turinčiomis architektūromis. Geriausios orientacijos metrikos gaunamos naudojant GRU su 1 sluoksniu ir 512 paslėptų vienetų

4.5 lentelė. Architektūros dydžio kitimo eksperimentai naudojant LSTM ir GRU architektūras su dėmesio mechanizmais

Nr.	Architektūra	RMSE	MAE	MED	MaxED	Antras MaxED	Riešo MED	Smiliaus galo MED	Vidutinis kampinis skirtumas	Kvaterniono geodezinė klaida	Vidutinis absoliutinis skirtumas	Treniravimo laikas (s)	Prognozės vid. laikas (s)
19.	LSTMAAttention(256, 2)	0.01984	<u>0.00794</u>	<u>0.01618</u>	1.04097	<u>0.02649</u>	<u>0.01574</u>	0.01741	<u>10.17866</u>	<u>0.17765</u>	0.09812	20.32	0.001267
20.	LSTMAAttention(512, 2)	0.01840	<u>0.00766</u>	<u>0.01564</u>	0.89703	<u>0.02538</u>	<u>0.01504</u>	<u>0.01709</u>	<u>9.90759</u>	<u>0.17292</u>	0.09036	51.75	0.001289
21.	LSTMAAttention(1024, 2)	0.02097	0.01059	0.02163	<u>0.45588</u>	0.03289	0.02081	0.02383	12.90756	0.22528	<u>0.03734</u>	111.60	0.003676
22.	GRUAttention(256, 2)	0.01958	0.00901	0.01821	1.17720	0.03011	0.01863	0.01828	10.78178	0.18818	0.07978	32.06	0.001283
23.	GRUAttention(512, 2)	<u>0.01835</u>	0.00825	0.01675	1.03223	0.02744	0.01667	<u>0.01740</u>	11.59020	0.20229	0.16887	45.30	0.001217
24.	GRUAttention(1024, 2)	<u>0.01570</u>	0.00989	0.01996	<u>0.44942</u>	0.03372	0.02052	0.02200	14.05830	0.24536	<u>0.04362</u>	116.14	0.002826



4.5 pav. Architektūros dydžio kitimo eksperimentai naudojant LSTM ir GRU architektūras su dėmesio mechanizmais

Atliekant bandymus su dėmesio mechanizmus turinčiomis architektūromis pastebėti prastesni rezultatai, lyginant su įprastomis architektūromis. Nuspręsta šių architektūrų nenaudoti bandant prognozuoti vieną kadrą į ateitį. Naudojamas horizontas (15 kadrų) yra per trumpas.

Toliau pateikiami rezultatai, kai eksperimentai yra surikiuoti pagal paskaičiuotą pozicijos, orientacijos ir bendrus įverčius, pateikiant 5 geriausias reikšmes (žr. 4.6 lentelė, 4.7 lentelė ir 4.8 lentelė).

4.6 lentelė. 5 geriausios architektūros lyginant bendrą įvertį

Nr.	Architektūra	Pozicijos įvertis	Orientacijos įvertis	Bendras įvertis
16	GRU (512, 2)	0.8822	0.7415	0.8400
10	LSTM (512, 1)	0.8427	0.8266	0.8379
11	LSTM (512, 2)	0.9337	0.5831	0.8286
9	LSTM (256, 2)	0.9244	0.5342	0.8074
17	GRU (512, 3)	0.9021	0.5740	0.8036

4.7 lentelė. 5 geriausios architektūros lyginant pozicijos įvertį

Nr.	Architektūra	Pozicijos įvertis	Orientacijos įvertis	Bendras įvertis
13	LSTM (1024, 2)	0.9392	0.1076	0.6898
11	LSTM (512, 2)	0.9337	0.5831	0.8286
9	LSTM (256, 2)	0.9244	0.5342	0.8074
17	GRU (512, 3)	0.9021	0.5740	0.8036
16	GRU (512, 2)	0.8822	0.7415	0.8400

4.8 lentelė. 5 geriausios architektūros lyginant orientacijos įvertį

Nr.	Architektūra	Pozicijos įvertis	Orientacijos įvertis	Bendras įvertis
15	GRU (512, 1)	0.3817	1.0000	0.5672
10	LSTM (512, 1)	0.8427	0.8266	0.8379
16	GRU (512, 2)	0.8822	0.7415	0.8400
14	GRU (256, 2)	0.7736	0.5997	0.7215
11	LSTM (512, 2)	0.9337	0.5831	0.8286

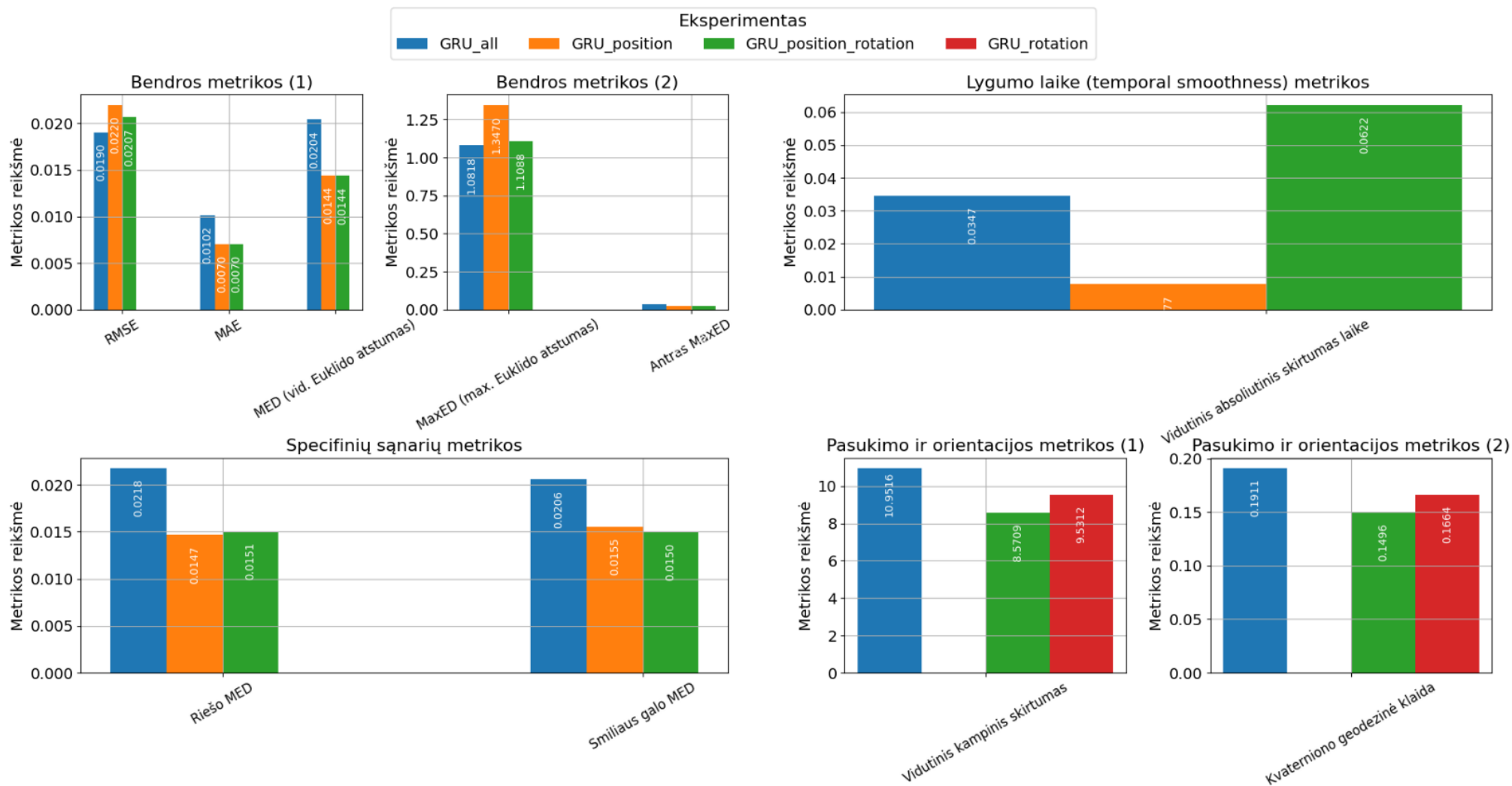
Bendro įvėrcio atveju prognozėms, atsižvelgiant į metrikas, tinkamiausia architektūra yra GRU su 512 paslėptų vienetų ir 2 sluoksniais. Pozicijos prognozių atveju optimaliai veikia LSTM 1024 paslėptų vienetų ir 2 sluoksnius turinti architektūra, tačiau ši architektūra pakankamai prastai prognozuoja rankos sąnarių orientaciją. Pastebėta, kad orientacijų prognozėms atlikti tinkamesnės mažą sluoksnių ir neuronų kiekį turinčios architektūros – GRU arba LSTM su 512 neuronų ir 1 sluoksniu. Kadangi atliktų prognozių vidutinis laikas yra pakankamai mažas, siekiant išgauti rezultatus galima naudoti ensamblinius modelius arba kombinuoti skirtingus modelius pozicijos ir orientacijos prognozėms atlikti tačiau darbe šie eksperimentai nėra išbandyti ar aprašyti. Tolimesniuose eksperimentuose nuspręsta naudoti LSTM (1024 paslėptų vienetų ir 2 sluoksniai), GRU (512 paslėptų vienetų ir 1 arba 2 sluoksnių) architektūras.

4.3. Eksperimentai keičiant įvesties ypatybes

Eksperimente bandoma išsiaiškinti, kuri įvesties ypatybių kombinacija prognozuoja tiksliausiai. Bandoma prognozuoti poziciją ir pasukimą naudojant tik poziciją, poziciją ir pasukimą arba visas ypatybes. Pateiktuose eksperimentuose naudojama viena ranka, įrenginio erdvė, prognozuojama pozicija ir pasukimas (arba tik pozicija, arba tik pasukimas, kai naudojama viena įvestis), 512 elementų duomenų kiekis, sujungta vidutinio Euklido atstumo ir hierarchinė pirštų sąnarių atstumo klaidos funkcijos, kampinio kvaterniono klaidos funkcija pasukimui, naudojami specifinių sąnarių svoriai ir nenaudojamas svorių nykimas. Eksperimentų metrikos ir lyginamieji grafikai pateikiami toliau (žr. 4.9-4.11 lenteles ir 4.6-4.8 pav.).

4.9 lentelė. Įvesties ypatybių kitimo eksperimentai naudojant GRU (512, 2 sluoksnių) architektūrą

Nr.	Įvesties ypatybės	RMSE	MAE	MED	MaxED	Antras MaxED	Riešo MED	Smiliaus galo MED	Vidutinis kampinis skirtumas	Kvaterniono geodezinė klaida	Vidutinis absoliutinis skirtumas	Treniravimo laikas (s)	Prognozės vid. laikas (s)
25.	Visos	<u>0.01905</u>	<u>0.01016</u>	0.02041	<u>1.08184</u>	0.03407	0.02177	0.02057	10.95163	0.19114	0.03469	52.03	0.001172
26.	Pozicija + orientacija	<u>0.02066</u>	<u>0.00703</u>	<u>0.01438</u>	<u>1.10877</u>	<u>0.02374</u>	<u>0.01505</u>	<u>0.01499</u>	<u>8.57094</u>	<u>0.14959</u>	0.06219	31.14	0.001057
27.	Pozicija	0.02198	<u>0.00703</u>	<u>0.01443</u>	1.34702	<u>0.02484</u>	<u>0.01469</u>	<u>0.01554</u>	-	-	-	23.08	0.001197
28.	Pasukimas	-	-	-	-	-	-	-	<u>9.53116</u>	<u>0.16635</u>	-	8.58	0.001235

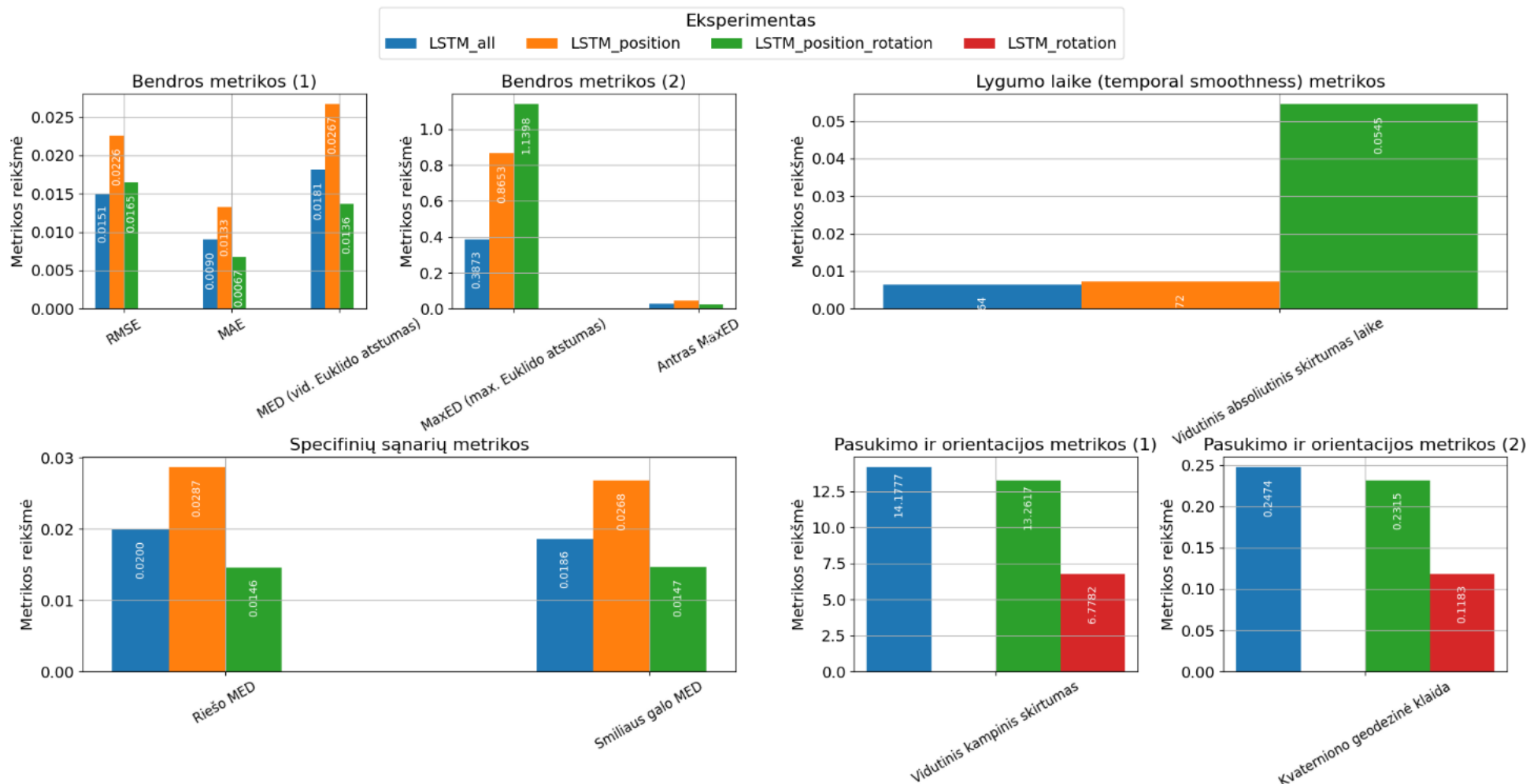


4.6 pav. Įvesties ypatybių kitimo eksperimentai naudojant GRU (512, 2 sluoksnių) architektūrą

Naudojant GRU (512 vienetų, 2 sluoksniai) architektūrą optimalios metrikos gaunamos kaip įvesti naudojant poziciją ir pasukimą.

4.10 lentelė. Įvesties ypatybių kitimo eksperimentai naudojant LSTM (1024, 2 sluoksnių) architektūrą

Nr.	Įvesties ypatybės	RMSE	MAE	MED	MaxED	Antras MaxED	Riešo MED	Smiliaus galo MED	Vidutinis kampinis skirtumas	Kvaterniono geodezinė klaida	Vidutinis absoliutinis skirtumas	Treniravimo laikas (s)	Prognozės vid. laikas (s)
29.	Visos	<u>0.01505</u>	<u>0.00900</u>	<u>0.01808</u>	<u>0.38731</u>	<u>0.02966</u>	<u>0.01996</u>	<u>0.01860</u>	14.17773	0.24745	<u>0.00641</u>	167.20	0.003606
30.	Pozicija + orientacija	<u>0.01646</u>	<u>0.00672</u>	<u>0.01364</u>	1.13984	<u>0.02354</u>	<u>0.01464</u>	<u>0.01468</u>	<u>13.26167</u>	<u>0.23146</u>	<u>0.05453</u>	116.14	0.003423
31.	Pozicija	0.02257	0.01327	0.02669	<u>0.86533</u>	0.04777	0.02867	0.02680	-	-	-	148.60	0.003702
32.	Pasukimas	-	-	-	-	-	-	-	<u>6.77815</u>	<u>0.11830</u>	-	27.98	0.003487

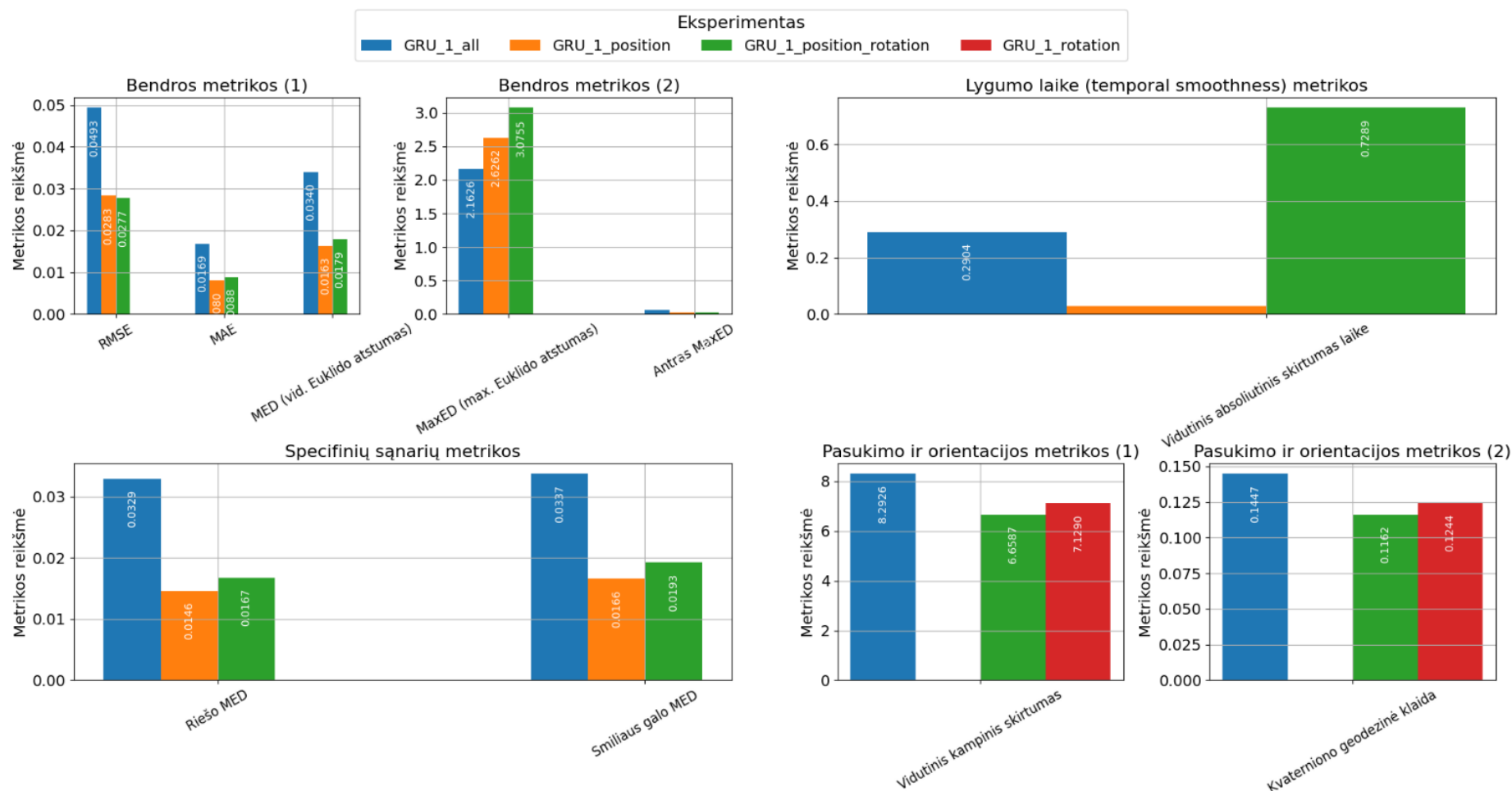


4.7 pav. Įvesties ypatybių kitimo eksperimentai naudojant LSTM (1024, 2 sluoksnių) architektūrą

LSTM (1024, 2) architektūros eksperimentuose pastebima, kad geriausios metrikos pozicijai prognozuoti gaunamos kaip įvesti naudojant poziciją ir orientaciją. Pasukimo prognozėms tinkamiausia įvestis yra tik orientacijos kvaternionų naudojimas.

4.11 lentelė. Įvesties ypatybių kitimo eksperimentai naudojant GRU (512, 1 sluoksnio) architektūrą

Nr.	Ypatybės	RMSE	MAE	MED	MaxED	Antras MaxED	Riešo MED	Smiliaus galo MED	Vidutinis kampinis skirtumas	Kvaterniono geodezinė klaida	Vidutinis absoliutinis skirtumas	Treniravimo laikas (s)	Prognozės vid. laikas (s)
33.	Visos	0.04934	0.01688	0.03398	<u>2.16257</u>	0.05703	0.03295	0.03373	8.29264	0.14473	0.29038	72.03	0.000652
34.	Pozicija + orientacija	<u>0.02771</u>	<u>0.00882</u>	<u>0.01787</u>	3.07554	<u>0.02955</u>	<u>0.01674</u>	<u>0.01926</u>	<u>6.65871</u>	<u>0.11622</u>	<u>0.72886</u>	25.25	0.000645
35.	Pozicija	<u>0.02833</u>	<u>0.00804</u>	<u>0.01629</u>	<u>2.62618</u>	<u>0.02730</u>	<u>0.01463</u>	<u>0.01657</u>	-	-	-	44.22	0.000684
36.	Pasukimas	-	-	-	-	-	-	-	<u>7.12896</u>	<u>0.12442</u>	-	7.88	0.000718



4.8 pav. Įvesties ypatybių kitimo eksperimentai naudojant GRU (512, 1 sluoksnio) architektūrą

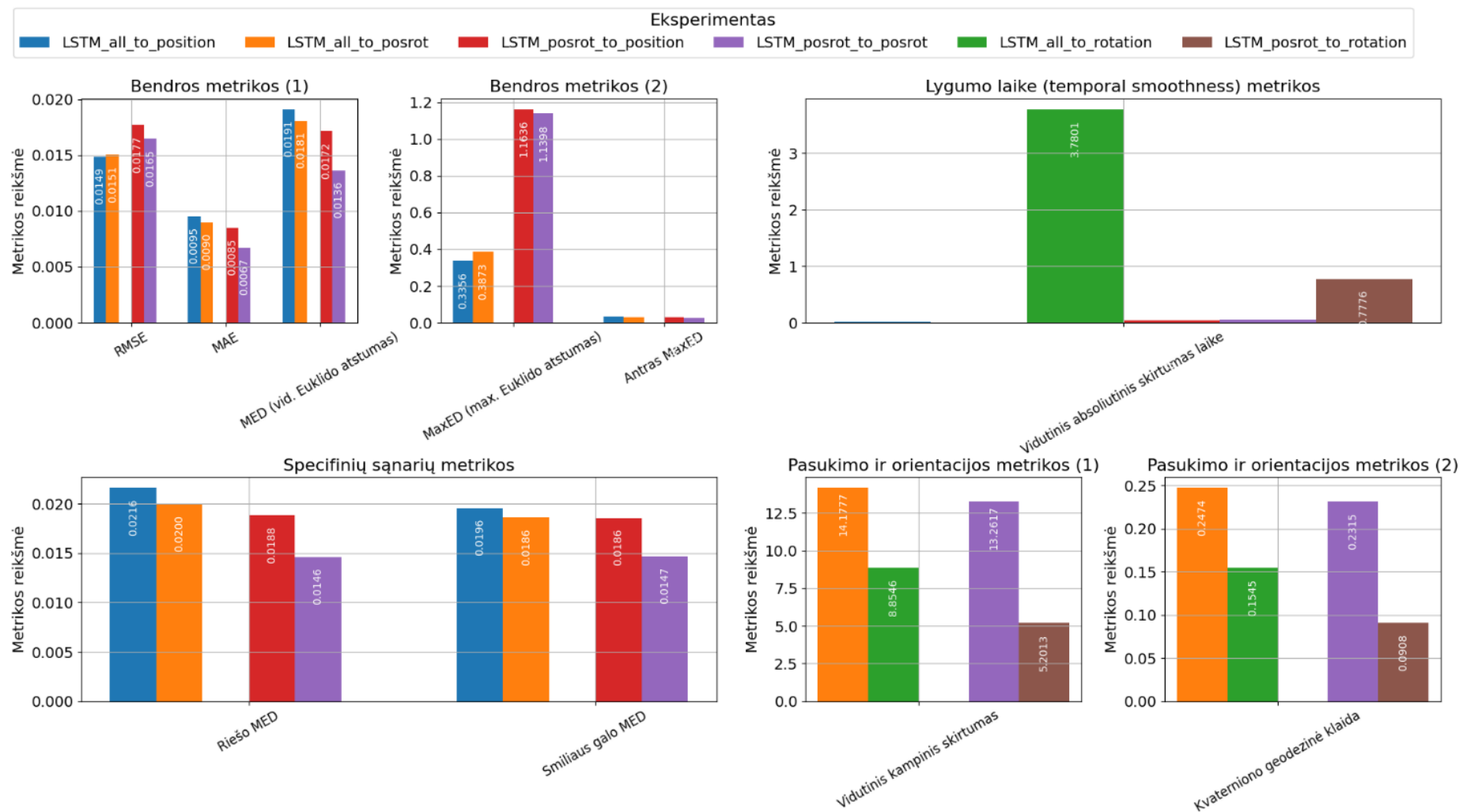
GRU (512, 1) architektūros atveju geriausios pozicijos metrikos gaunamos kaip įvesti naudojant tik poziciją (ir prognozuojant tik poziciją). Naudojant šią architektūrą orientacijai prognozuoti tinkamiausia pozicijos ir orientacijos įvestis. Tolimesniuose eksperimentuose nuspręsta naudoti poziciją ir pasukimą arba tik pasukimą kaip įvestį.

4.4. Eksperimentai keičiant įvesties ir išvesties ypatybes

Eksperimento metu siekiama išsiaiškinti geriausias įvesties ypatybių ir išvesties ypatybių kombinacijas. Naudojama LSTM 1024 neuronų architektūra su 2 sluoksniais. Pateiktuose eksperimentuose naudojama viena ranka, įrenginio erdvė, prognozuojama pozicija ir pasukimas, 512 elementų duomenų kiekis, sujungta vidutinio Euklido atstumo ir hierarchinė pirštų sąnarių atstumo klaidos funkcijos, kampinio kvaterniono klaidos funkcija pasukimui, naudojami specifinių sąnarių svoriai ir nenaudojamas svorių nykimas. Papildomi eksperimentai leidžia geriau suprasti dinamiką tarp įvesties ir išvesties ypatybių, bandoma subalansuoti įvestis ir išvestis taip, kad jų nebūtų per daug ir būtų gaunamas kuo didesnis tikslumas. Eksperimentai ir gauti lyginamieji grafikai pateikiami toliau (žr. 4.12 lentelė, 4.9 pav.).

4.12 lentelė. Įvesties ir išvesties ypatybių kitimo eksperimentai naudojant LSTM (1024, 2 sluoksnių) architektūrą

Nr.	Įvesties ypatybės	Išvesties ypatybės	RMSE	MAE	MED	MaxED	Antras MaxED	Riešo MED	Smiliaus galo MED	Vidutinis kampinis skirtumas	Kvaterniono geodezinė klaida	Vidutinis absoliutinis skirtumas	Treniravimo laikas (s)	Prognozės vid. laikas (s)
37.	Visos	Pozicija + orientacija	<u>0.01505</u>	0.00900	0.01808	<u>0.38731</u>	<u>0.02966</u>	0.01996	0.01860	14.17773	0.24745	0.00641	167.20	0.003606
38.	Visos	Pozicija	<u>0.01488</u>	0.00953	0.01911	<u>0.33562</u>	0.03227	0.02163	0.01957	-	-	0.00908	98.22	0.003586
39.	Visos	Orientacija	-	-	-	-	-	-	-	<u>8.85462</u>	<u>0.15454</u>	-	36.67	0.003694
40.	Pozicija + orientacija	Pozicija + orientacija	0.01646	<u>0.00672</u>	<u>0.01364</u>	1.13984	<u>0.02354</u>	<u>0.01464</u>	<u>0.01468</u>	13.26167	0.23146	0.05453	116.14	0.003423
41.	Pozicija + orientacija	Pozicija	0.01770	<u>0.00850</u>	<u>0.01717</u>	1.16359	0.02982	<u>0.01884</u>	<u>0.01857</u>	-	-	0.03958	90.97	0.003759
42.	Pozicija + orientacija	Orientacija	-	-	-	-	-	-	-	<u>5.20135</u>	<u>0.09078</u>	-	32.65	0.003591



4.9 pav. Įvesties ir išvesties ypatybių kitimo eksperimentai naudojant LSTM (1024, 2 sluoksnių) architektūrą

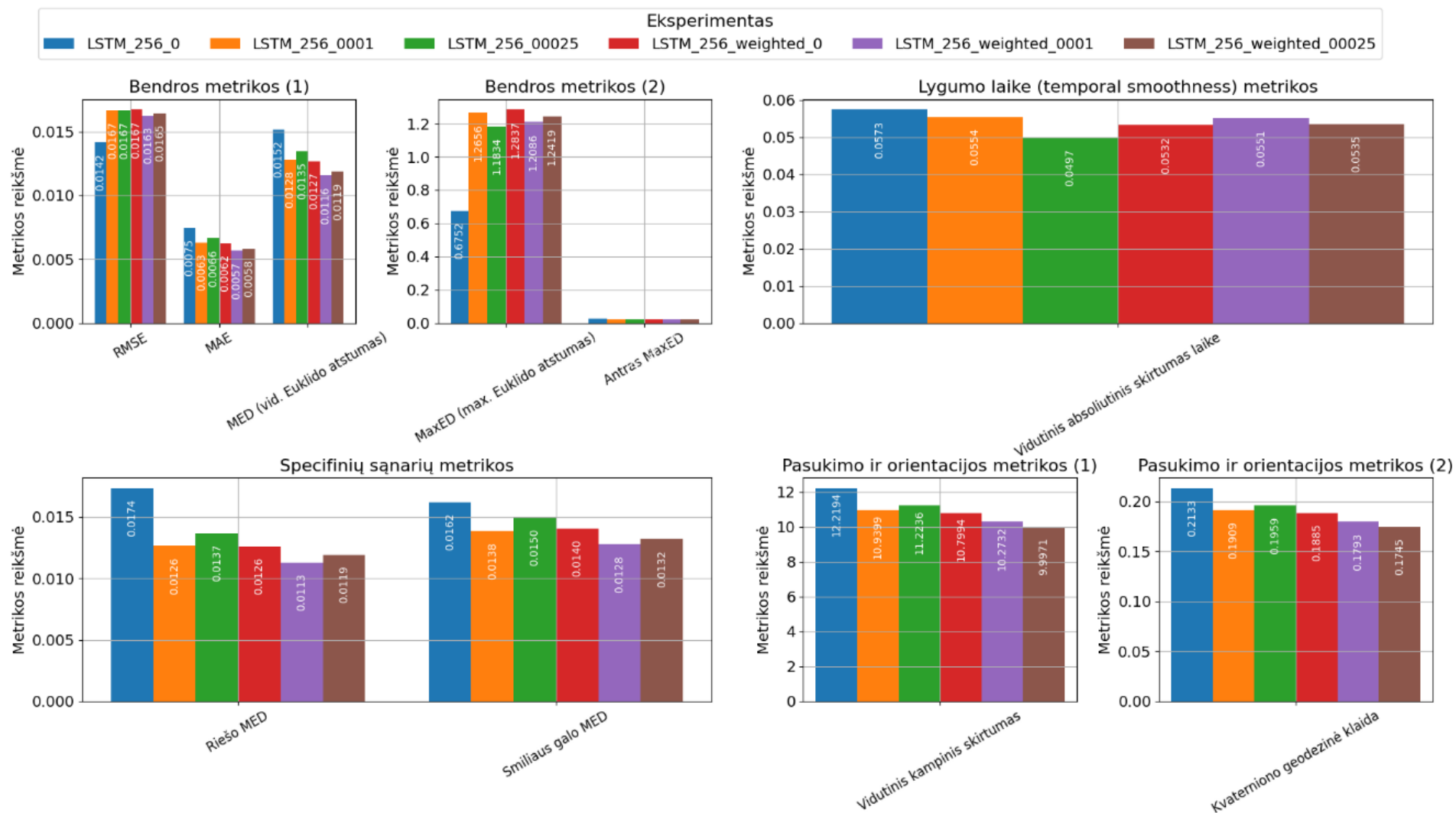
Naudojant LSTM architektūrą pastebėta, kad didžiausias pozicijos tikslumas gaunamas naudojant poziciją ir pasukimą kaip įvesti kai prognozuojama pozicija ir įvestis. Didžiausias orientacijos tikslumas gaunamas kaip įvestį naudojant poziciją ir orientaciją, bet prognozuojant tik orientaciją.

4.5. Eksperimentai keičiant kitus parametrus

Eksperimentų metu bandoma išsiaiškinti geriausią duomenų kiekį (*batch size*), sąnarių svorių naudojimo naudą ir svorių nykimo parametro naudojimo efektą. Naudojama LSTM 1024 neuronų architektūra su 2 sluoksniais. Pateiktuose eksperimentuose naudojama viena ranka, įrenginio erdvė, visos įvesties ypatybės, prognozuojama pozicija ir pasukimas, sujungta vidutinio Euklido atstumo ir hierarchinė pirštų sąnarių atstumo klaidos funkcijos, kampinio kvaterniono klaidos funkcija pasukimui. Eksperimentai pateikiami 256, 512 ir 1024 duomenų kiekiui. Gauti rezultatai patiekiami lentelėse ir lyginamuosiuose grafikuose (žr. 4.13-4.15 lenteles ir 4.10-4.12 pav.).

4.13 lentelė. Įvesties ir išvesties ypatybių kitimo eksperimentai naudojant LSTM (1024, 2 sluoksnių) architektūrą su 256 duomenų kiekiu

Nr.	Sąnarių svoriai?	Svorių nykimas	RMSE	MAE	MED	MaxED	Antras MaxED	Riešo MED	Smiliaus galo MED	Vidutinis kampinis skirtumas	Kvaterniono geodezinė klaida	Vidutinis absoliutinis skirtumas	Treniravimo laikas (s)	Prognozės vid. laikas (s)
43.	Taip	-	0.01672	0.00625	0.01268	1.28374	0.02185	0.01261	0.01405	10.79942	0.18849	<u>0.05316</u>	82.13	0.003893
44.	Taip	0.001	0.01627	<u>0.00567</u>	<u>0.01156</u>	1.20863	<u>0.02013</u>	<u>0.01128</u>	<u>0.01278</u>	10.27318	0.17930	0.05508	127.23	0.003635
45.	Taip	0.0025	0.01645	0.00582	0.01186	1.24192	0.02030	0.01192	0.01320	<u>9.99706</u>	<u>0.17448</u>	0.05348	113.39	0.003727
46.	Ne	-	<u>0.01418</u>	0.00746	0.01518	<u>0.67525</u>	0.02575	0.01735	0.01622	12.21938	0.21327	0.05734	81.21	0.003691
47.	Ne	0.001	0.01666	0.00631	0.01279	1.26561	0.02199	0.01265	0.01384	10.93994	0.19094	0.05539	84.01	0.003478
48.	Ne	0.0025	0.01667	0.00664	0.01346	1.18336	0.02303	0.01369	0.01496	11.22362	0.19589	0.04973	76.60	0.003578

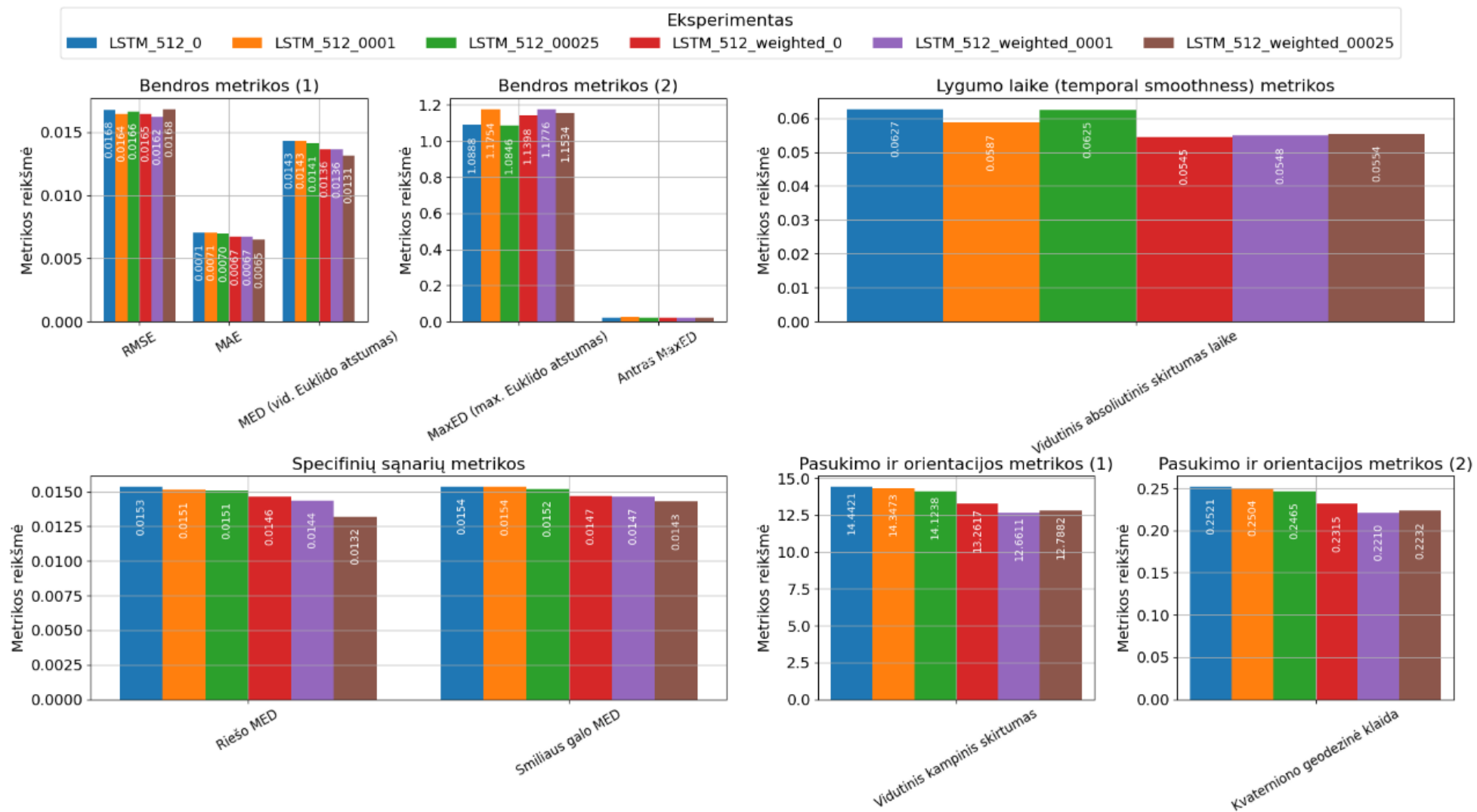


4.10 pav. Įvesties ir išvesties ypatybių kitimo eksperimentai naudojant LSTM (1024, 2 sluoksnių) architektūrą su 256 duomenų kiekiu

Naudojant 256 duomenų kiekį optimali parametų kombinacija gaunama naudojant sąnarių svorius ir 0,001 svorio nykimo reikšmę.

4.14 lentelė. Įvesties ir išvesties ypatybių kitimo eksperimentai naudojant LSTM (1024, 2 sluoksnių) architektūrą su 512 duomenų kiekiu

Nr.	Šanarių svoriai?	Svorių nykimas	RMSE	MAE	MED	MaxED	Antras MaxED	Riešo MED	Smiliaus galo MED	Vidutinis kampinis skirtumas	Kvaterniono geodezinė klaida	Vidutinis absoliutinis skirtumas	Treniravimo laikas (s)	Prognozės vid. laikas (s)
49.	Taip	-	0.01646	0.00672	0.01364	1.13984	0.02354	0.01464	0.01468	13.26167	0.23146	<u>0.05453</u>	116.14	0.003423
50.	Taip	0.001	<u>0.01621</u>	0.00672	0.01360	1.17758	0.02349	0.01437	0.01467	<u>12.66113</u>	<u>0.22098</u>	0.05483	111.28	0.003884
51.	Taip	0.0025	0.01683	<u>0.00648</u>	<u>0.01315</u>	1.15343	<u>0.02271</u>	<u>0.01322</u>	<u>0.01430</u>	12.78818	0.22320	0.05539	104.01	0.003687
52.	Ne	-	0.01678	0.00706	0.01432	1.08883	0.02467	0.01535	0.01538	14.44210	0.25206	0.06272	108.28	0.003473
53.	Ne	0.001	0.01644	0.00706	0.01430	1.17540	0.02494	0.01514	0.01535	14.34729	0.25041	0.05866	105.56	0.003560
54.	Ne	0.0025	0.01664	0.00696	0.01411	<u>1.08462</u>	0.02446	0.01508	0.01522	14.12377	0.24651	0.06250	112.59	0.003522

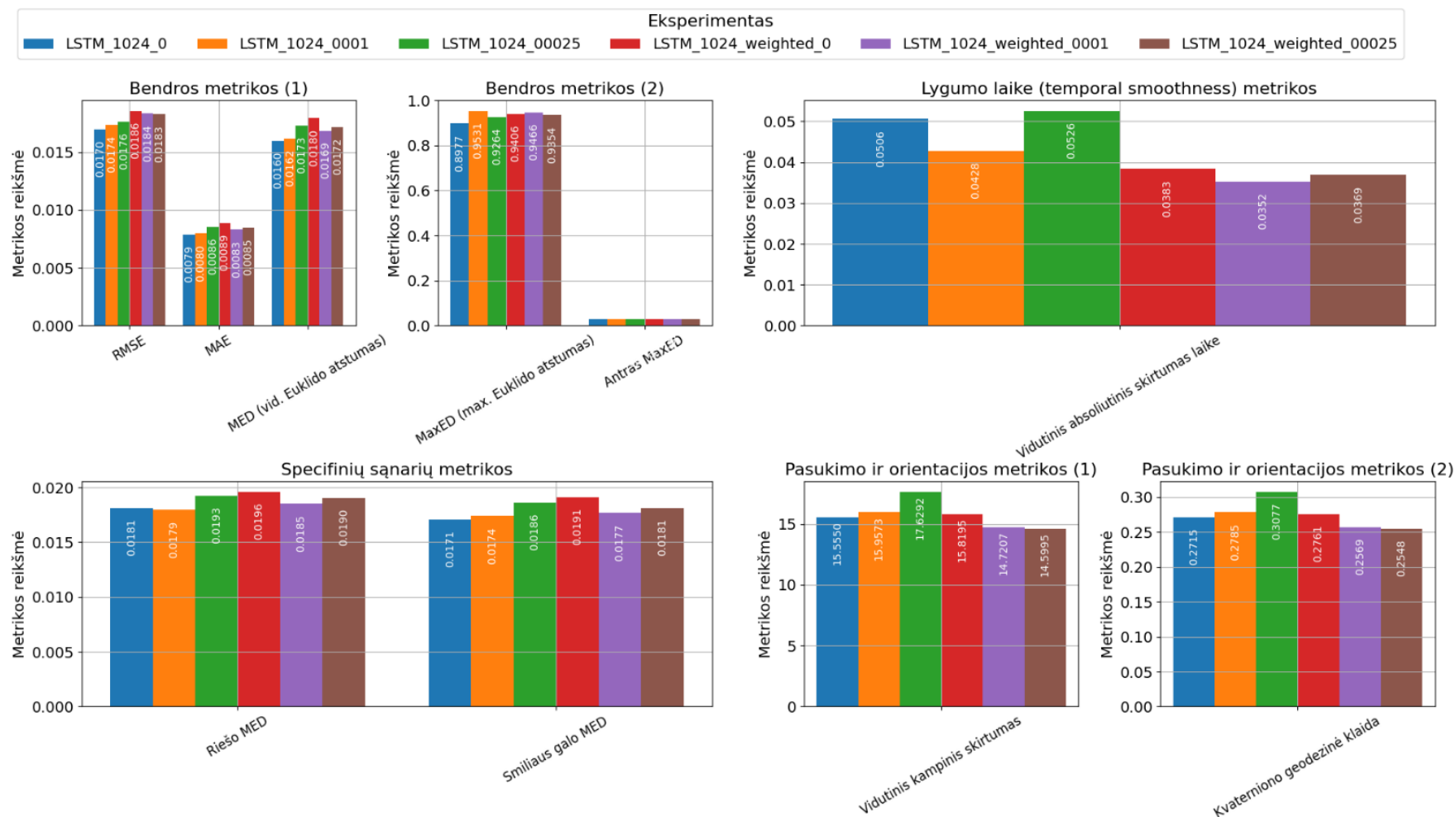


4.11 pav. Įvesties ir išvesties ypatybių kitimo eksperimentai naudojant LSTM (1024, 2 sluoksnių) architektūrą su 512 duomenų kiekiu

512 duomenų kiekio atveju optimali kombinacija gaunama naudojant sąnarių svorius ir 0,0025 svorio nykimo reikšmę.

4.15 lentelė. Įvesties ir išvesties ypatybių kitimo eksperimentai naudojant LSTM (1024, 2 sluoksnių) architektūrą su 1024 duomenų kiekiu

Nr.	Šanarių svoriai?	Svorių nykimas	RMSE	MAE	MED	MaxED	Antras MaxED	Riešo MED	Smiliaus galo MED	Vidutinis kampinis skirtumas	Kvaterniono geodezinė klaida	Vidutinis absoliutinis skirtumas	Treniravimo laikas (s)	Prognozės vid. laikas (s)
55.	Taip	-	0.01858	0.00887	0.01797	0.94063	0.02991	0.01959	0.01913	15.81948	0.27610	0.03833	151.06	0.003518
56.	Taip	0.001	0.01837	0.00831	0.01685	0.94663	0.02821	0.01852	0.01768	14.72069	0.25692	<u>0.03525</u>	191.84	0.003515
57.	Taip	0.0025	0.01830	0.00850	0.01720	0.93535	0.02873	0.01904	0.01813	<u>14.59948</u>	<u>0.25481</u>	0.03687	168.53	0.003422
58.	Ne	-	<u>0.01698</u>	<u>0.00790</u>	<u>0.01599</u>	<u>0.89765</u>	0.02753	0.01814	<u>0.01709</u>	15.55504	0.27149	0.05061	135.18	0.003500
59.	Ne	0.001	0.01739	0.00800	0.01620	0.95315	<u>0.02748</u>	<u>0.01794</u>	0.01739	15.95730	0.27851	0.04283	141.32	0.003477
60.	Ne	0.0025	0.01765	0.00856	0.01734	0.92643	0.02929	0.01927	0.01862	17.62921	0.30769	0.05255	142.52	0.003670



4.12 pav. Įvesties ir išvesties ypatybių kitimo eksperimentai naudojant LSTM (1024, 2 sluoksnių) architektūrą su 1024 duomenų kiekiu

1024 duomenų kiekio eksperimentuose optimali kombinacija gaunama nenaudojant sąnarių svorio ar svorių nykimo.

Eksperimentų metu pastebėta, kad optimalu naudoti mažesnę duomenų kiekį. Svorių nykimas daugeliu atveju padidina tikslumą.

4.6. Eksperimentų rezultatų palyginimas

Siekiant įvertinti optimalią kombinaciją, naudojami pozicijos, orientacijos ir bendri įverčiai. Pateikiamos eksperimentų ir rezultatų lentelės pagal įvertį, kuriose pateikiami geriausi 5 eksperimentai (žr. 4.16 lentelė, 4.17 lentelė ir 4.18 lentelė).

4.16 lentelė. Geriausios kombinacijos pagal bendrą įvertį

Nr.	RMSE	MAE	MED	MaxED	Antras MaxED	Riešo MED	Smiliaus galo MED	Vidutinis kampinis skirtumas	Kvaterniono geodezinė klaida	Vidutinis absoliutinis skirtumas	Pozicijos įvertis	Orientacijos įvertis	Bendras įvertis
44.	0.01627	0.00567	0.01156	1.20863	0.02013	0.01128	0.01278	10.27318	0.1793	0.05508	0.9812	0.6621	0.8855
45.	0.01645	0.00582	0.01186	1.24192	0.0203	0.01192	0.0132	9.99706	0.17448	0.05348	0.9685	0.6805	0.8821
43.	0.01672	0.00625	0.01268	1.28374	0.02185	0.01261	0.01405	10.79942	0.18849	0.05316	0.9388	0.6270	0.8453
26.	0.02066	0.00703	0.01438	1.10877	0.02374	0.01505	0.01499	8.57094	0.14959	0.06219	0.8742	0.7755	0.8446
1.	0.02066	0.00703	0.01438	1.10877	0.02374	0.01505	0.01499	8.57094	0.14959	0.06219	0.8742	0.7755	0.8446

Bendro įverčio atveju tinkamiausios kombinacijos gaunamos naudojant LSTM 1024 neuronų architektūrą su 2 sluoksniais su viena ranka, 256 duomenų kiekiu, įrenginio erdvė, naudojant pozicijos ir pasukimo įvestį, prognozuojant poziciją ir pasukimą, naudojant sujungtą vidutinio Euklido atstumo ir hierarchinę pirštų sąnarių atstumo klaidos funkcijas, kampinio kvaterniono klaidos funkcija pasukimui kai pasitelkiamas specifinių sąnarių svoris ir svorių nykimas, lygus 0,001. Gaunamas 1,156 cm vidutinis Euklido atstumas, maksimalus 2 cm atstumas (išmetant pirmąjį, kuris dažniausiai rodo kraštutinį atvejį), kai vidutinis kampinis skirtumas yra lygus 10,27 laipsniams.

4.17 lentelė. Geriausios kombinacijos pagal pozicijos įvertį

Nr.	RMSE	MAE	MED	MaxED	Antras MaxED	Riešo MED	Smiliaus galo MED	Vidutinis kampinis skirtumas	Kvaterniono geodezinė klaida	Vidutinis absoliutinis skirtumas	Pozicijos įvertis	Orientacijos įvertis	Bendras įvertis
44.	0.01627	0.00567	0.01156	1.20863	0.02013	0.01128	0.01278	10.27318	0.1793	0.05508	0.9812	0.6621	0.8855
45.	0.01645	0.00582	0.01186	1.24192	0.0203	0.01192	0.0132	9.99706	0.17448	0.05348	0.9685	0.6805	0.8821
43.	0.01672	0.00625	0.01268	1.28374	0.02185	0.01261	0.01405	10.79942	0.18849	0.05316	0.9388	0.6270	0.8453
47.	0.01666	0.00631	0.01279	1.26561	0.02199	0.01265	0.01384	10.93994	0.19094	0.05539	0.9370	0.6177	0.8412

51.	0.01683	0.00648	0.01315	1.15343	0.02271	0.01322	0.0143	12.78818	0.2232	0.05539	0.9217	0.4946	0.7935
-----	---------	---------	---------	---------	---------	---------	--------	----------	--------	---------	--------	--------	--------

Vertinant tik pozicijos įvertį gaunami panašūs rezultatai. Pirmas tris vietas užima tos pačios parametrų kombinacijos kaip ir bendro įverčio atveju. To galima tikėtis, kadangi skaičiuojant bendrą įvertį naudojamas didelis pozicijos įverčio svoris, kadangi pozicijos prognozės yra svarbiausios rankų sąnarių prognozėms formuoti.

4.18 lentelė. Geriausios kombinacijos pagal orientacijos įvertį

Nr.	RMSE	MAE	MED	MaxED	Antras MaxED	Riešo MED	Smiliaus galo MED	Vidutinis kampinis skirtumas	Kvaterniono geodezinė klaida	Vidutinis absoliutinis skirtumas	Pozicijos įvertis	Orientacijos įvertis	Bendras įvertis
42	-	-	-	-	-	-	-	5.20135	0.09078	-	-	1.0	-
34	0.02771	0.00882	0.01787	3.07554	0.02955	0.01674	0.01926	6.65871	0.11622	0.72886	0.6018	0.9029	0.6921
32	-	-	-	-	-	-	-	6.77815	0.1183	-	-	0.8949	-
36	-	-	-	-	-	-	-	7.12896	0.12442	-	-	0.8716	-
10	0.03187	0.00738	0.01503	2.98603	0.02376	0.01307	0.01554	7.94122	0.1386	0.13529	0.8382	0.8174	0.8320

Orientacijos įverčio eksperimentų atveju optimaliai veikia įvairios kombinacijos. Geriausi rezultatai gaunami kai naudojama LSTM 1024 neuronų architektūra su 2 sluoksniais, kaip įvestis naudojama pozicija ir orientacija, tačiau prognozuojama tik orientacija. Šiame eksperimente naudojama viena ranka, įrenginio erdvė, 512 elementų duomenų kiekis, sujungta vidutinio Euklido atstumo ir hierarchinė pirštų sąnarių atstumo klaidos funkcijos, kampinio kvaterniono klaidos funkcija pasukimui, naudojami specifinių sąnarių svoriai ir nenaudojamas svorių nykimas. Pasiekiamas 5,2 laipsnių vidutinis kampinis skirtumas.

4.7. Kelių kadrų prognozių eksperimentai

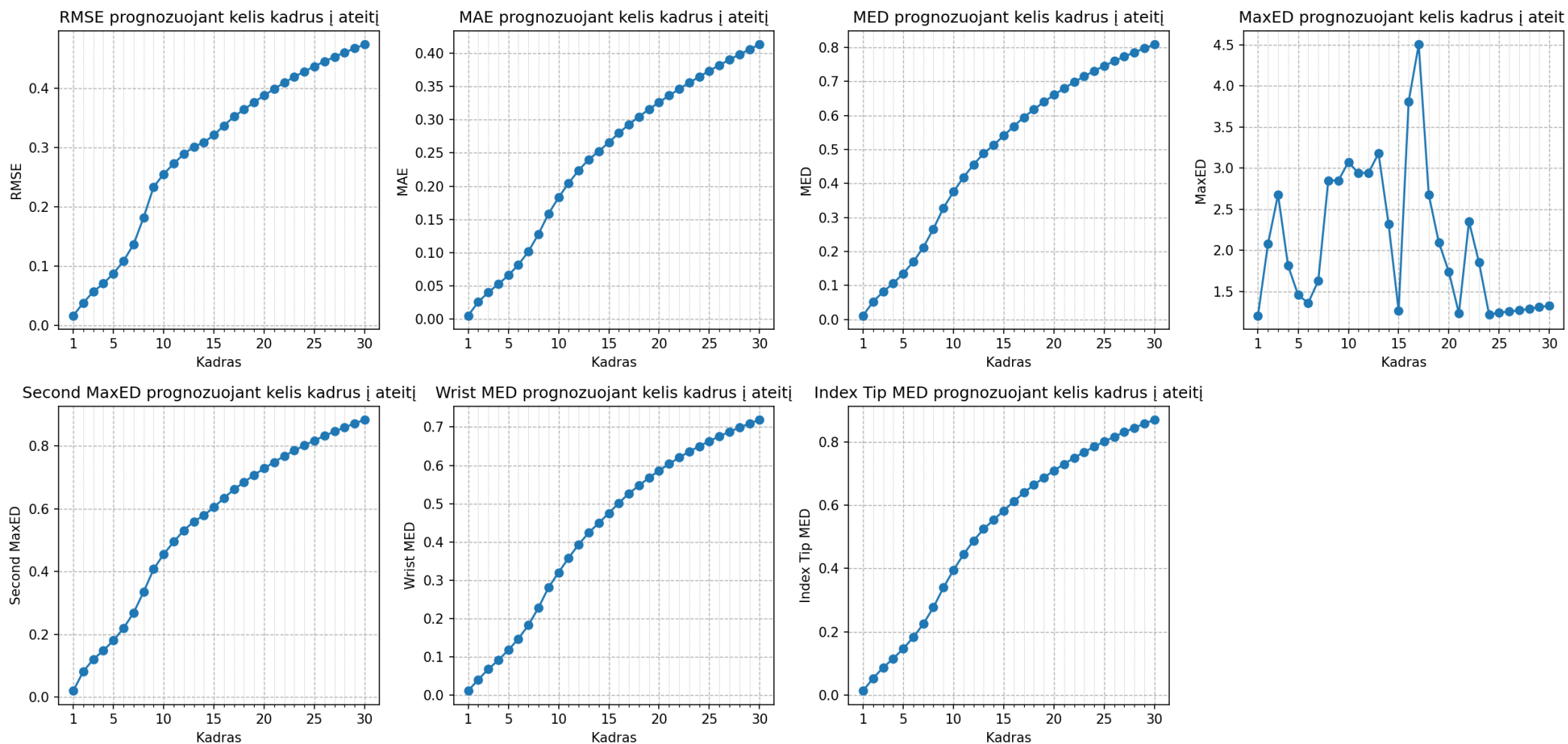
Papildomi kelių kadrų prognozių eksperimentai atliekami siekiant įvertinti modelio efektyvumą prognozuojant daugiau nei vieną kadrą į ateitį. Eksperimentuose palyginamas ankstesniuose eksperimentuose naudotas vieno kadro prognozės modelis ir papildomai pridėtas autoregresinis modelis. Naujas modelis prognozuoja ateities kadrus nuosekliai, kiekvieną prognozuotą kadrą naudodamas kaip įvestį naujai prognozei parinktą prognozių kadrų kiekį. Siekiant pagerinti naujo modelio prognozių kokybę, naudojami keletas patobulinimų, skirtų autoregresiniams modeliams. Naudojamas suplanuotas mokymasis (angl. *scheduled sampling*), kuris palaipsniui mažina priklausomybę nuo mokymo duomenų. Pridedamas papildomas klaidos funkcijos skaičiavimas kai skaičiuojami fiziniai apribojimai – ribojami per dideli pozicijos pokyčiai, klaida mažinama jei judesiai yra sklandūs (nėra didelių

skirtumų tarp kadru). Papildomas modelis prognozuoja tik vieną kadrą į ateitį, tačiau naudojamas modifikuotas treniravimas, leidžiantis modeliui geriau įsiminti ilgesnes sekas ir pasiekti geresnių rezultatų daugelio kadru prognozavimo režime.

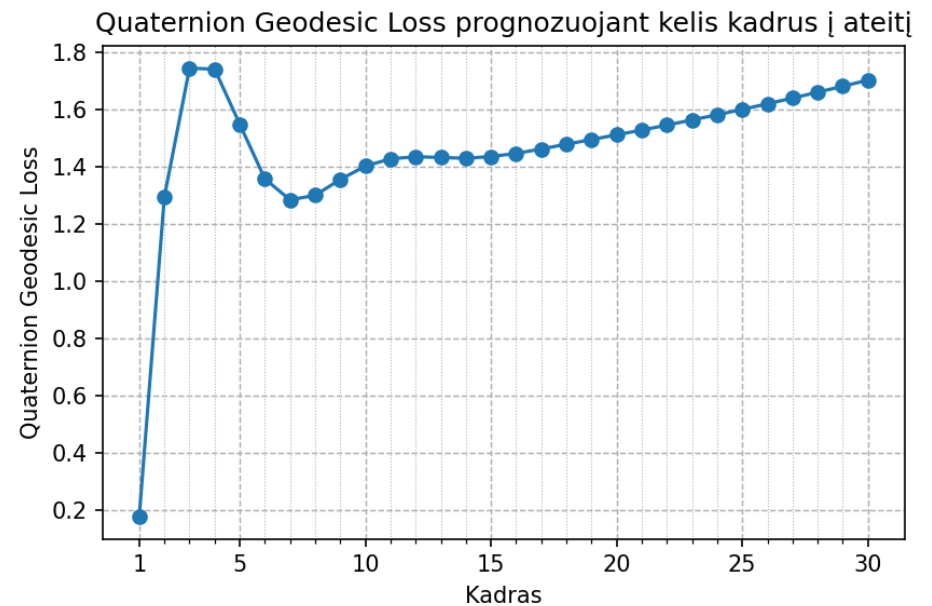
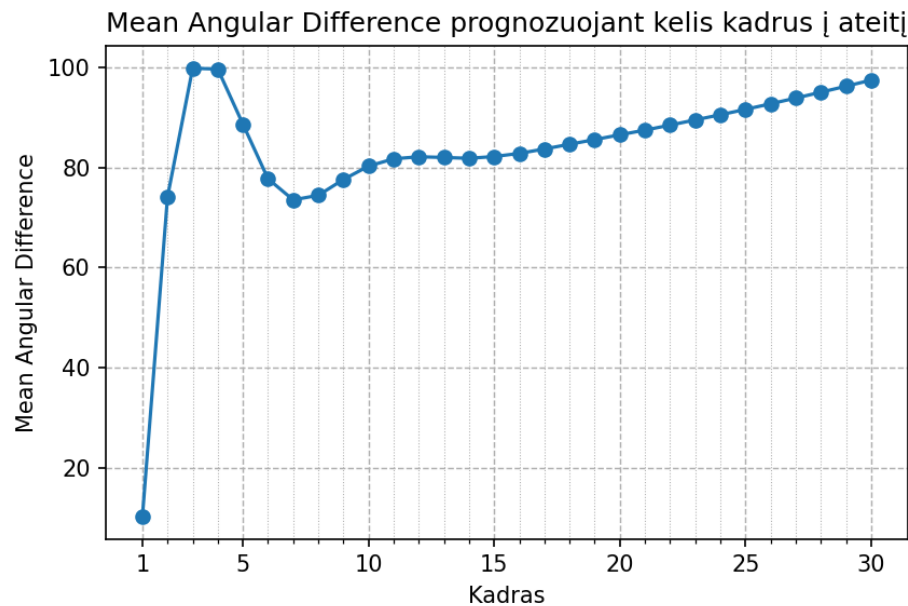
Naudojama 44 eksperimento parametrų kombinacija: LSTM (1024 paslėptų vienetų, 2 sluoksniai) su viena ranka, 256 duomenų kiekiu, įrenginio erdvė, pozicijos ir pasukimo įvestimi, prognozuojant poziciją ir pasukimą, naudojant sujungtą vidutinio Euklido atstumo ir hierarchinę pirštų sąnarių atstumo klaidos funkcijas, kampinio kvaterniono klaidos funkciją pasukimui kai pasitelkiamas specifinių sąnarių svoris ir svorių nykimas, lygus 0,001. Rezultatai pateikiami lentelėje ir grafikuose (žr. 4.19 lentelė, 4.13-4.16 pav.). Lentelėje pateikiamas prognozių kadru kiekis nurodo autoregresijos kadru kiekį (pvz. prognozių kadru kiekis 5 reiškia, kad klaidos funkcija skaičiuojama naudojant 5 kadru prognozes, kai ankstesnio kadro prognozės pridedamos į naudojamą 15 kadru horizontą). Metrikų kadrus nurodo prognozuoto kadro numerį, kuriam pateikiamos metrikos.

4.19 lentelė. Kelių kadru prognozių eksperimentų rezultatai

Nr.	Prognozių kadru kiekis	Metrikų kadrus	RMSE	MAE	MED	MaxED	Antras MaxED	Riešo MED	Smiliaus galo MED	Vidutinis kampinis skirtumas	Kvaterniono geodezinė klaida	Treniravimo laikas (s)	Prognozės vid. laikas (s)
61.	1 (įprastas modelis)	1	0.01627	0.00567	0.01156	1.20863	0.02013	0.01128	0.01278	10.27318	0.17930	127.23	0.003635
		5	0.08729	0.06609	0.13502	1.45762	0.02013	0.11755	0.14593	88.68572	1.54786		
		10	0.25508	0.18313	0.37589	3.07311	0.02013	0.32040	0.39507	80.30257	1.40154		
		30	0.47385	0.41306	0.80882	1.3292	0.02013	0.71909	0.87099	97.48645	1.70146		
62.	5	1	0.02923	0.00840	0.01729	2.16962	0.02774	0.01562	0.01912	12.43849	0.21709	393,49	0,003119
		5	0.02680	0.00891	0.01812	1.97231	0.02774	0.01709	0.02017	14.21153	0.24804		
		10	0.02651	0.01839	0.03648	0.71426	0.02774	0.03391	0.04046	24.44498	0.42665		
		30	0.09008	0.07313	0.14132	0.70981	0.02774	0.12574	0.15158	62.06615	1.08326		
63.	10	1	0.02287	0.00962	0.01970	1.30750	0.03163	0.01768	0.02143	15.28884	0.26684	1147,37	0,003069
		5	0.01956	0.00819	0.01662	1.01985	0.03163	0.01542	0.01843	15.09787	0.26351		
		10	0.02194	0.01064	0.02134	1.08550	0.03163	0.01978	0.02330	17.56566	0.30658		
		30	0.05649	0.03651	0.07360	1.16007	0.03163	0.06904	0.07806	40.78667	0.71186		

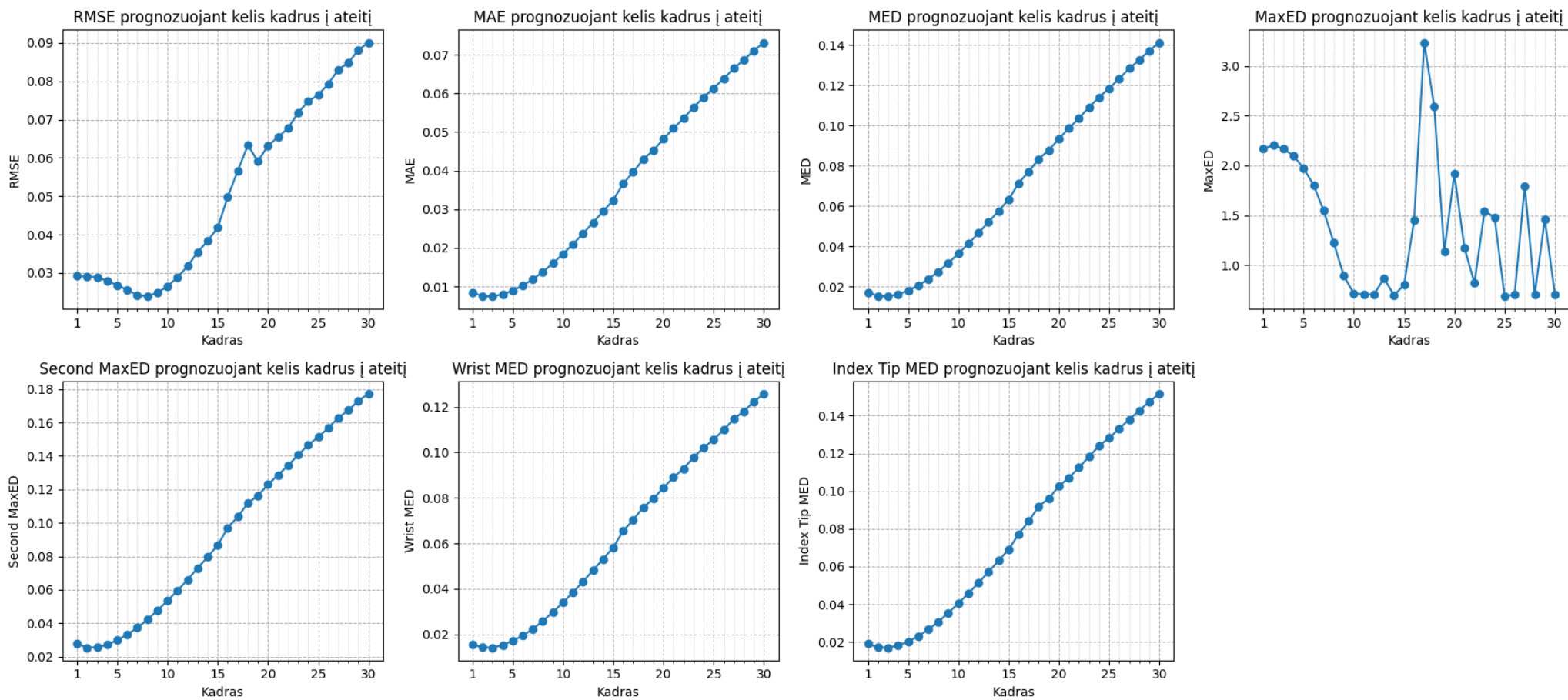


4.13 pav. Pozicijos metrikų rezultatai, kai naudojamas įprastas modelis

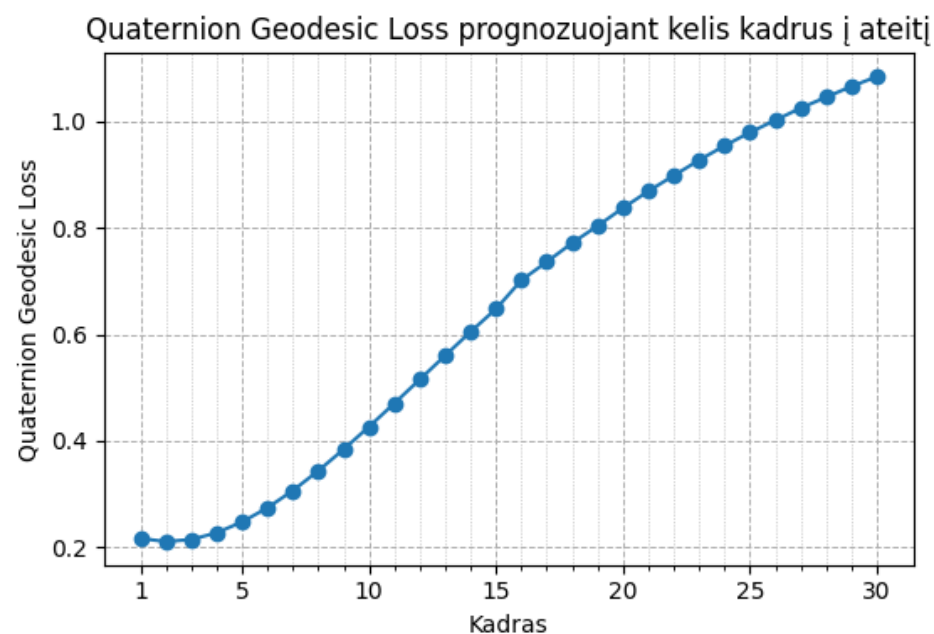
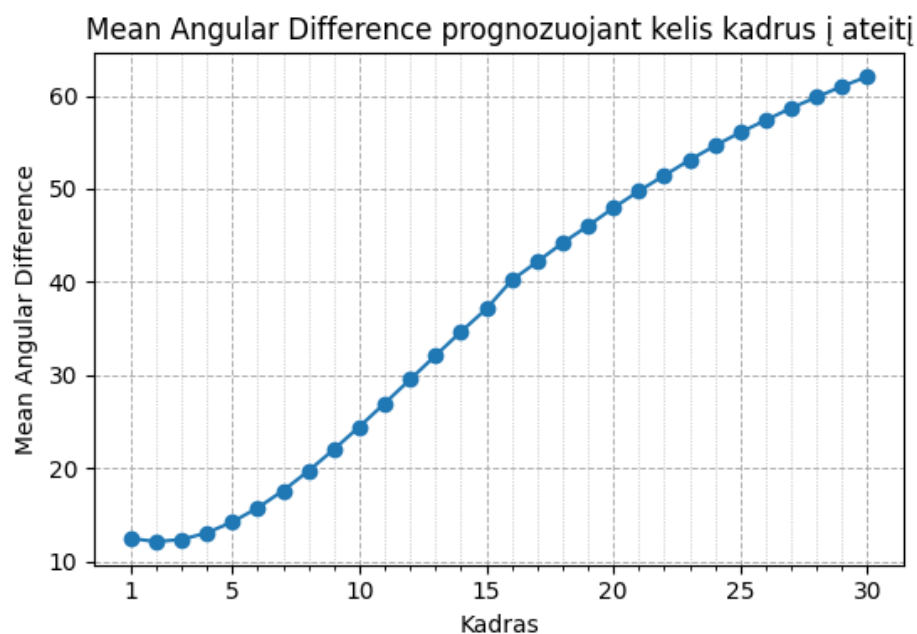


4.14 pav. Orientacijos metrikų rezultatai, kai naudojamas įprastas modelis

Įprasto modelio atveju pastebėta, kad pakankamai tikslios (<10 cm vidutinis Euklido atstumas) prognozės pozicijos atžvilgiu atliekamos tik 4 kadrams. Pasukimo prognozės atžvilgiu pastebima, kad pakankamas tikslumas išgaunamas tik pirmame prognozuojamame kadre. Eksperimentai patvirtina hipotezę, kad įprastas modelis, kurį treniruojant naudojami tik testų rinkinio duomenys (nėra autoregresijos) kelių kadrų prognozės atveju veikia pakankamai prastai.

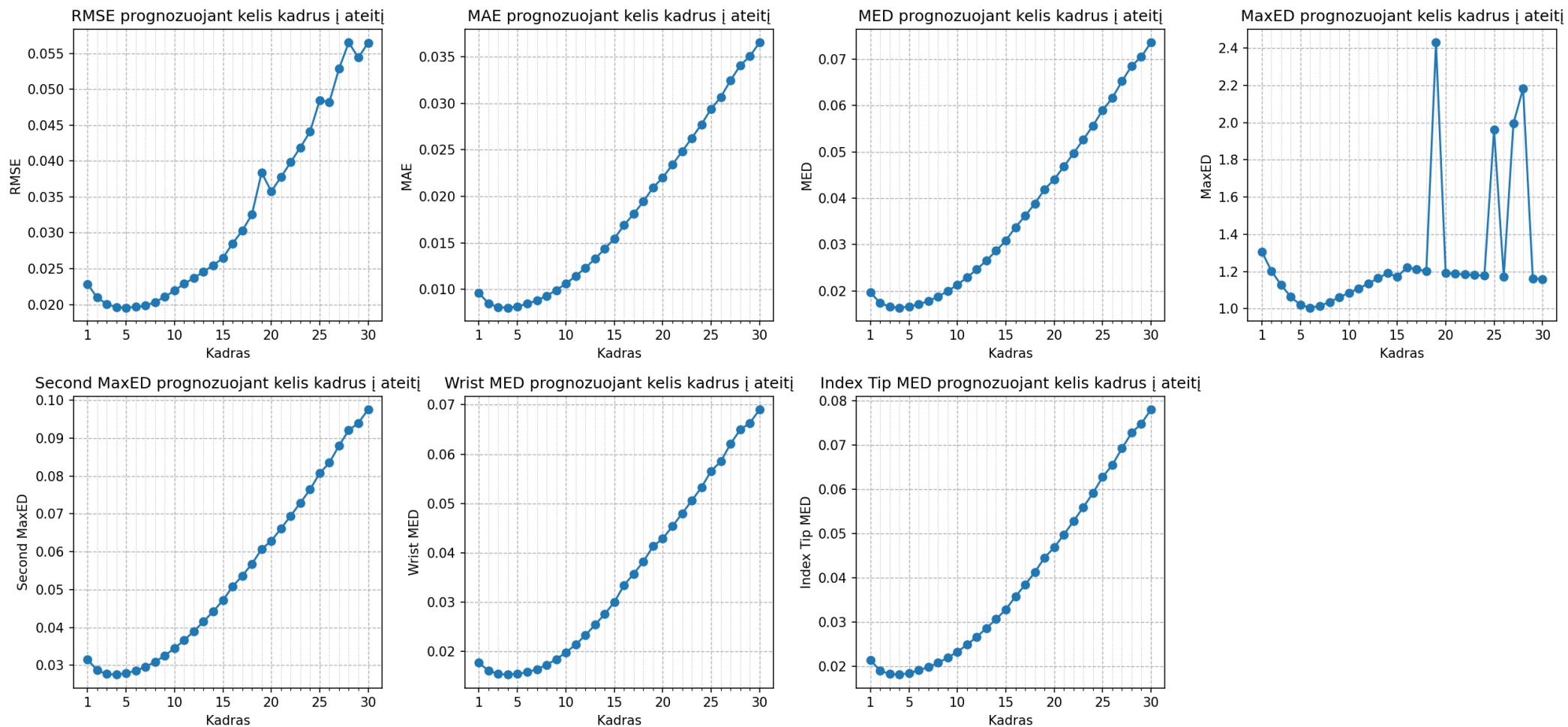


4.15 pav. Pozicijos metrikų rezultatai, kai naudojamas naujas modelis, treniruotas naudojant 5 kadrus

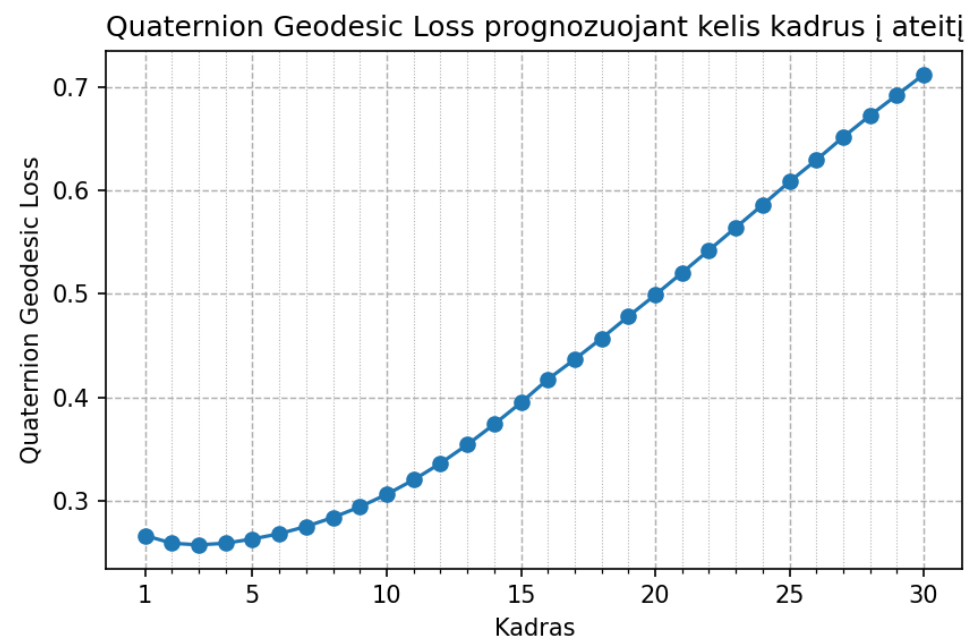
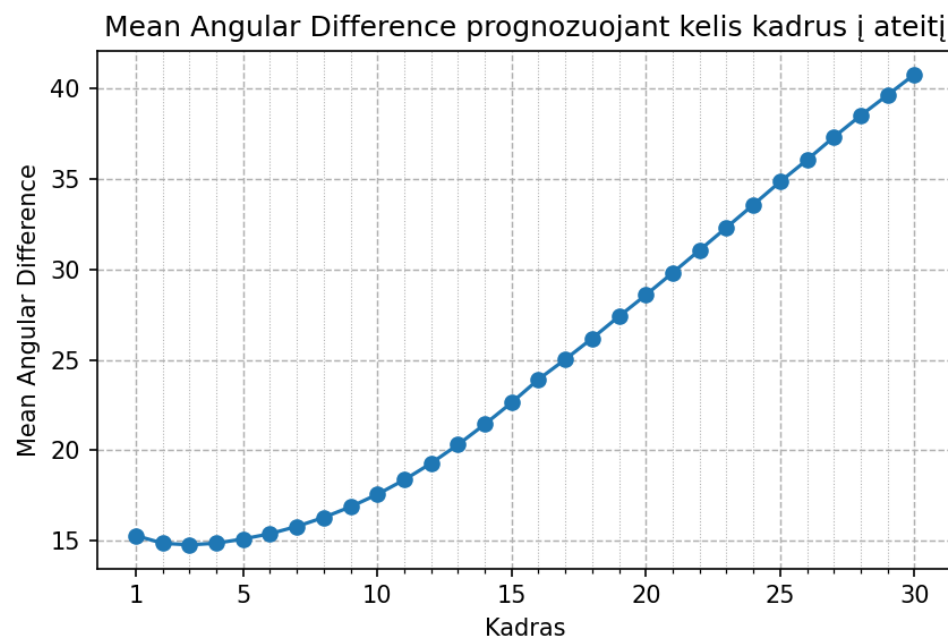


4.16 pav. Orientacijos metrikų rezultatai, kai naudojamas naujas modelis, treniruotas naudojant 5 kadrus

Kai treniruojant modelį naudojami 5 kadrai pasitelkiant autoregresiją pastebimi geresni rezultatai. Mažesnę nei 10 cm vidutinę prognozių tikslumą galima pasiekti prognozuojant 21 kadrą, panašų į anksčiau testuoto modelio tikslumą galima išgauti 5 kadrams. Orientacijos atžvilgiu modelis veikia žymiai geriau – panašus tikslumas, lyginant su įprastu modeliu, išlieka 5 kadrams. Tačiau, lyginant su modeliu, kurio treniravimo metu nenaudojama autoregresija pastebimos prastesnės metrikos pirmo kadro prognozės: vid. Euklido atstumas pirmame kadre pakyla nuo 1,156 cm iki 1,970 cm, vidutinis kampinis skirtumas pakyla nuo 10,27 laipsnio iki 15,29.



4.17 pav. Pozicijos metrikų rezultatai, kai naudojamas naujas modelis, treniruotas naudojant 10 kadrų



4.18 pav. Orientacijos metrikų rezultatai, kai naudojamas naujas modelis, treniruotas naudojant 10 kadrų

Kai treniruojant modelį naudojami 10 kadrų pasitelkiant autoregresiją kelių kadrų prognozės atveju pastebimi geriausi rezultatai. Mažesnę nei 10 cm vidutinę prognozių tikslumą galima pasiekti prognozuojant daugiau nei 30 kadrų, panašų į anksčiau testuoto modelio tikslumą galima išgauti 9 kadrų. Orientacijos atžvilgiu panašus tikslumas, lyginant su įprastu modeliu, išlieka 7 kadrų. Tačiau, lyginant su modeliu, kurio treniravimo metu nenaudojama autoregresija pastebimos prastesnės metrikos pirmo kadro prognozėms: vid. Euklido atstumas pirmame kadre pakyla nuo 1,156 cm iki 1,729 cm, vidutinis kampinis skirtumas pakyla nuo 10,27 laipsnio iki 12,43.

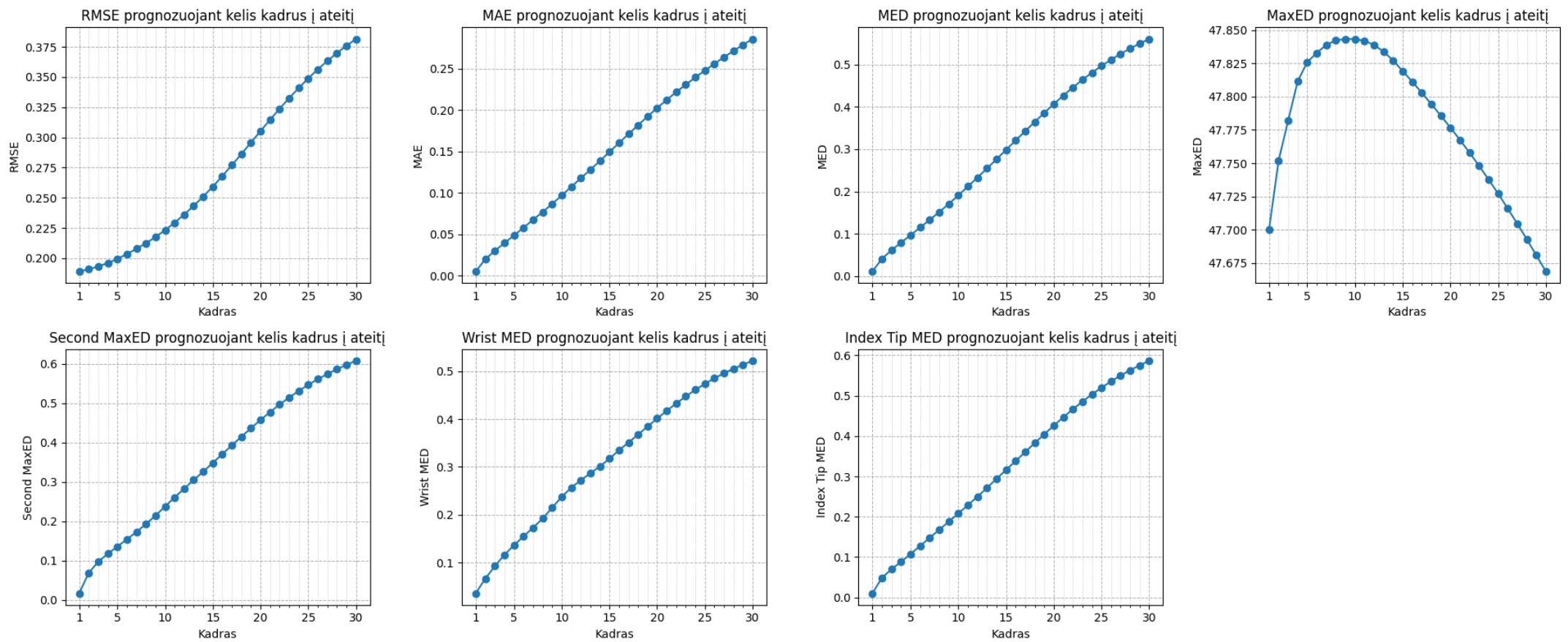
Atsižvelgiant į gautus rezultatus galima teigti, kad didinant regresijos kadrų kiekį gerėja ilgalaikių prognozių kokybė, tačiau prastėja pirmų kadrų prognozių tikslumas. Autoregresinio treniravimo modelių metrikose pastebima, kad pirmas prognozės kadras nėra pats geriausias – tai gali lemti suplanuoto mokymosi naudojimas.

4.8. Eksperimentai naudojant papildytą duomenų rinkinį

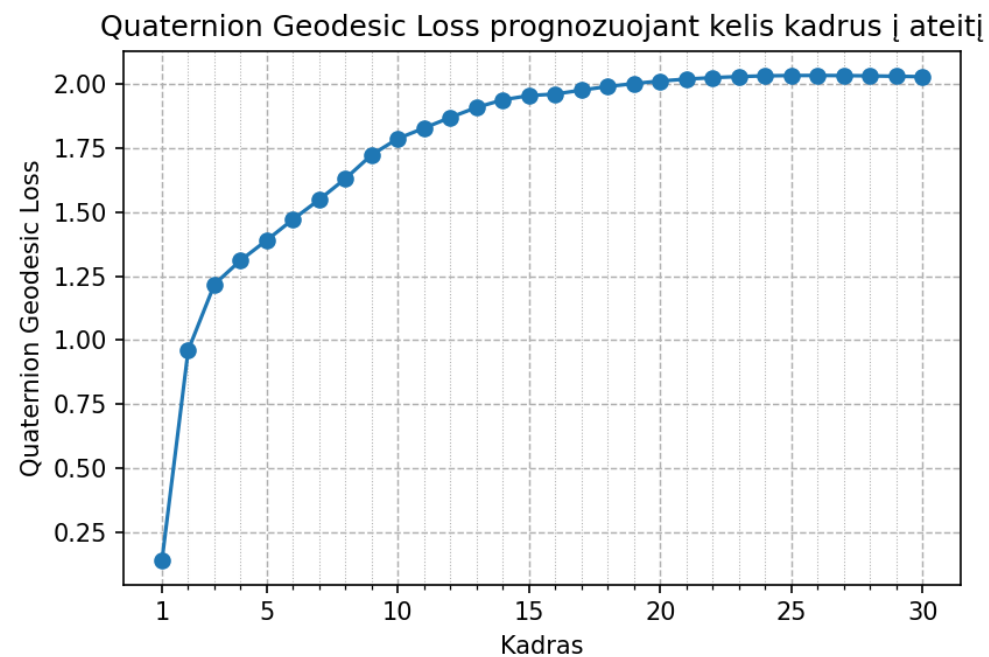
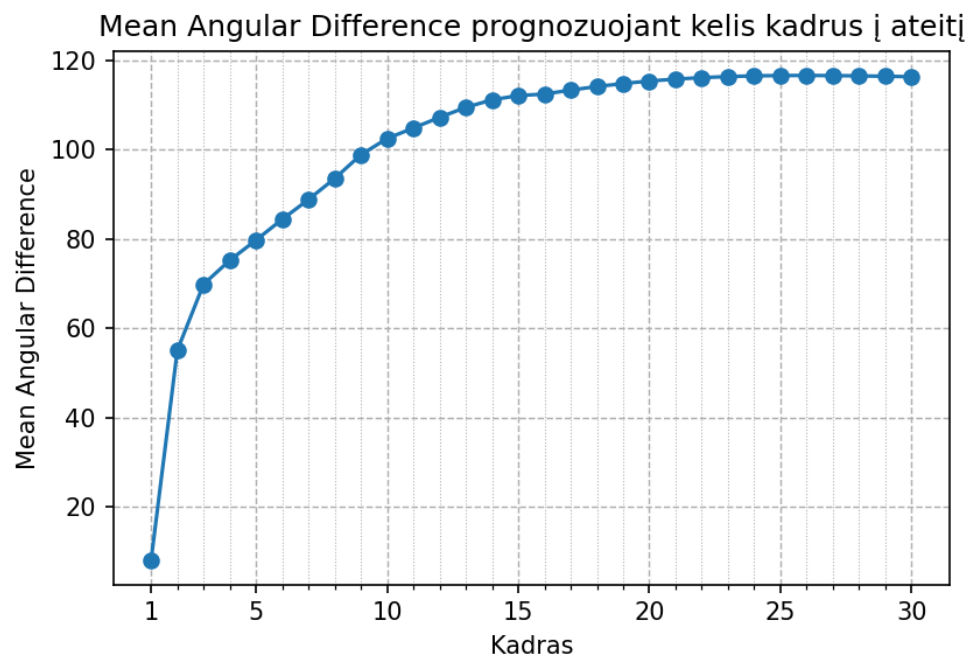
Aprašomi eksperimentai, kai naudojamas padidintas duomenų rinkinys, sudarytas iš rankos sąnarių sekimo pilno rinkinio (26000 kadrų) ir gestų sekimui paruošto rinkinio (80658 kadrų), kuris iš viso sujungia 106658 įvesties kadrus. Eksperimento metu gaunamos metrikos atspindi realesnę naudojimo aplinką, kadangi sekamuose rankų judesiuose užfiksuoti įvairūs judesiai, jų kombinacijos ir dažnai atliekami gestai: suspaustas kumštis, „ok“ gestas, ištiesto delno, rodymo pirštu, aukštyn pakelto arba nuleisto nykščio, pergalės arba „v“, mojavimo rankomis ir neutralios rankos padėties kadrai. Naudojama iki šiol rasta optimali parametrų kombinacija: 2 sluoksnių LSTM su 1024 paslėptų vienetų, kai naudojama įrenginio erdvė, naudojant pozicijos ir pasukimo įvestį, prognozuojant poziciją ir pasukimą, naudojant sujungtą vidutinio Euklido atstumo ir hierarchinę pirštų sąnarių atstumo klaidos funkcijas, kampinio kvaterniono klaidos funkcija pasukimui kai pasitelkiamas specifinių sąnarių svoris ir svorių nykimas, lygus 0,001. Naudojamas didesnis duomenų kiekis, lygus 1024, siekiant sutrumpinti treniravimo laiką. Rezultatai pateikiami lentelėje ir grafikuose (žr. 4.20 lentelė, 4.19-4.22 pav.).

4.20 lentelė. Kelių kadrų prognozių eksperimentų su papildytu duomenų rinkiniu rezultatai

Nr.	Prognozių kadrų kiekis	Metrikų kadras	RMSE	MAE	MED	MaxED	Antras MaxED	Riešo MED	Smiliaus galo MED	Vidutinis kampinis skirtumas	Kvaterniono geodezinė klaida	Treniravimo laikas (s)	Prognozės vid. laikas (s)
64.	1 (įprastas modelis)	1	0.18901	0.00516	0.01114	47.70012	0.01678	0.03641	0.01052	8.03591	0.14025	1320,37	0.003197
		5	0.1994	0.04866	0.09692	47.82594	0.01678	0.13652	0.10783	79.72396	1.39145		
		10	0.22319	0.0969	0.19092	47.84336	0.01678	0.23753	0.20814	102.47435	1.78852		
		30	0.38146	0.28587	0.56057	47.66865	0.01678	0.52151	0.58637	116.32423	2.03024		
65.	5	1	0.18989	0.00697	0.01471	47.69542	0.02083	0.08719	0.01384	11.52911	0.20122	3887.58	0,003844
		5	0.18993	0.00775	0.01625	47.69849	0.02083	0.08956	0.01555	13.72174	0.23949		
		10	0.19089	0.01645	0.03362	47.73457	0.02083	0.1046	0.03471	33.49157	0.58454		
		30	0.20042	0.05391	0.10653	47.75155	0.02083	0.1677	0.10692	103.38601	1.80443		

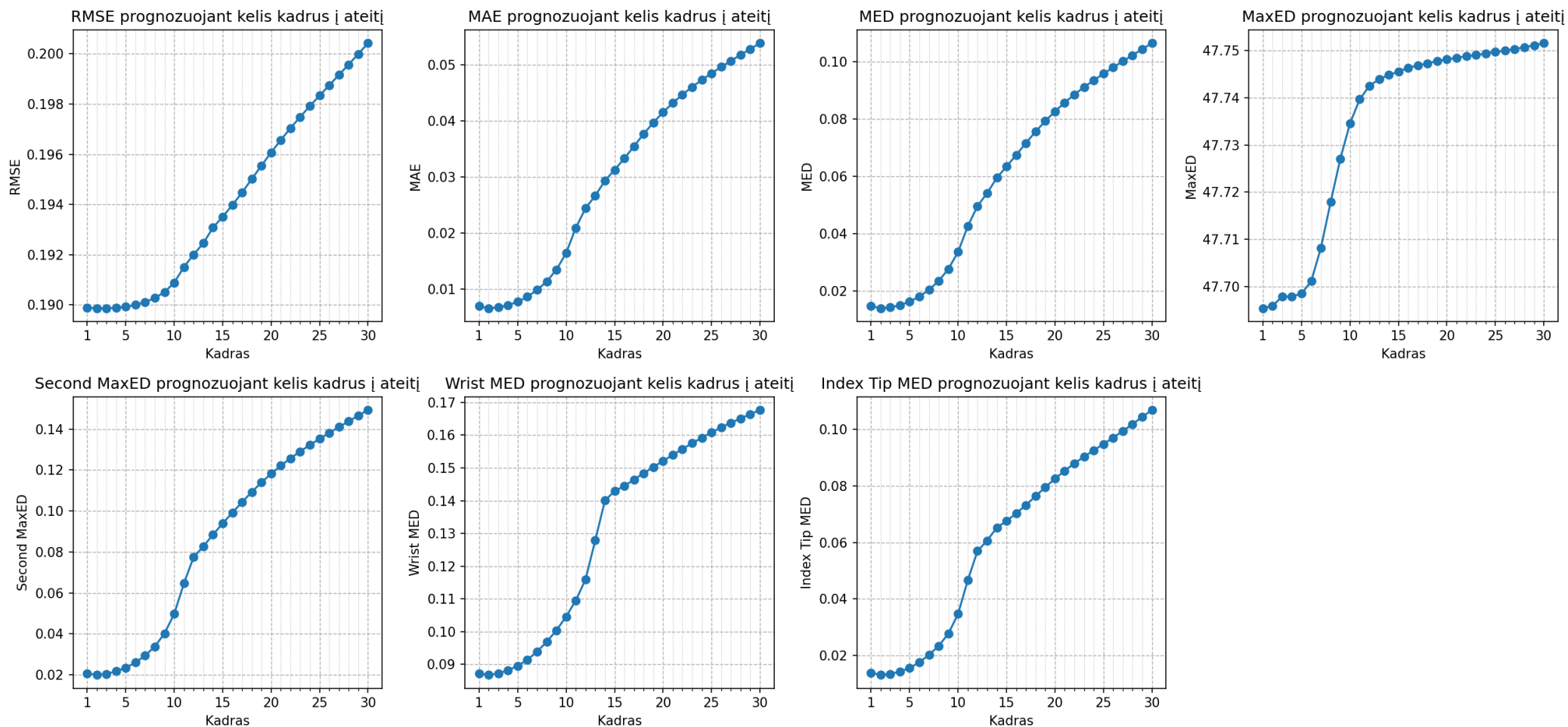


4.19 pav. Pozicijos metrikų rezultatai su papildytų duomenų rinkiniu, kai naudojamas įprastas modelis

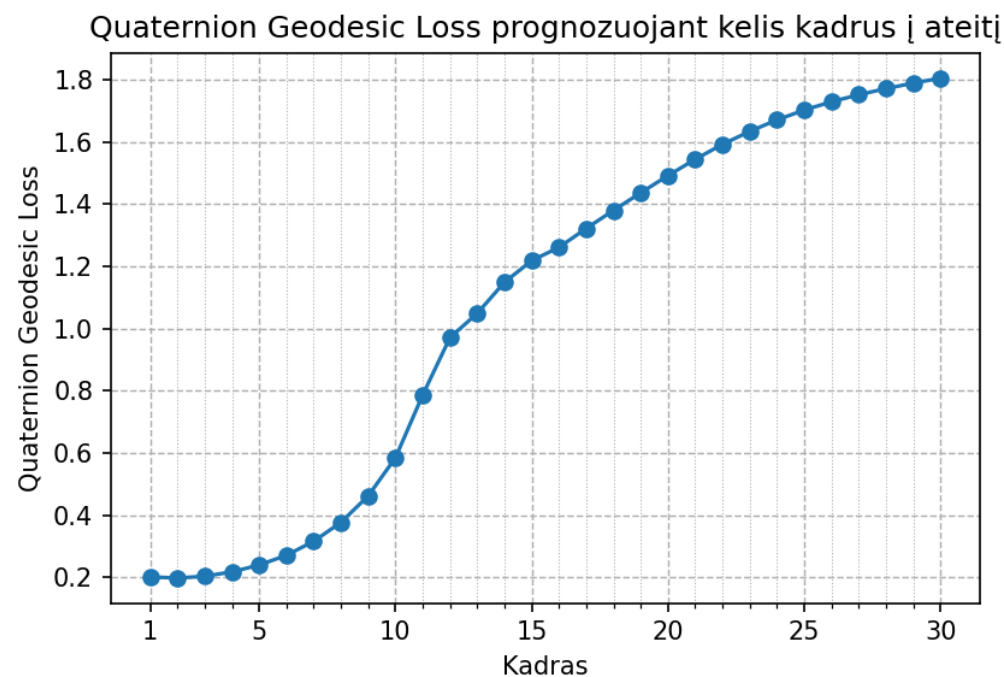
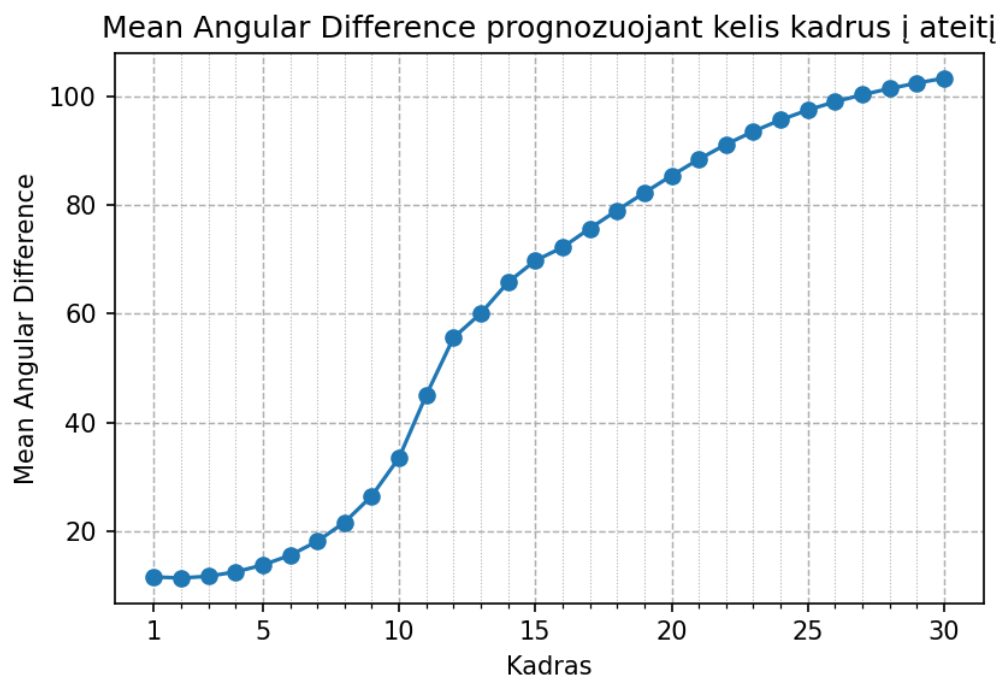


4.20 pav. Orientacijos metrikų rezultatai su papildytų duomenų rinkiniu, kai naudojamas įprastas modelis

Įprasto modelio atveju naudojant papildyta duomenų rinkinį pastebimos geresnės metrikos, nei naudojant mažesnę duomenų rinkinį. Galima daryti prielaidą, kad didesnis duomenų kiekis leidžia optimaliai išmokti sąsajas tarp rankos pirštų sąnarių ir sumažinti prognozių paklaidas.



4.21 pav. Pozicijos metrikų rezultatai su papildytų duomenų rinkiniu, kai naudojamas autoregresinis modelis, treniruotas naudojant 5 kadrus



4.22 pav. Pasukimo metrikų rezultatai su papildytų duomenų rinkiniu, kai naudojamas autoregresinis modelis, treniruotas naudojant 5 kadrus

Su papildytu duomenų rinkiniu naudojant modelį su 5 kadrų autoregresiniu treniravimu taip pat pastebimi geresni rezultatai, lyginant su mažesnio duomenų rinkinio rezultatais.

Bendru atveju gaunamos naujos mažiausios vieno kadro prognozių pozicijos ir orientacijos paklaidos: vidutinis Euklido atstumas vieno kadro prognozių atveju yra 1.11 cm, antras maksimalus Euklido atstumas tarp sąnarių yra 1.67 cm, vidutinis kampinis skirtumas yra 8,03 laipsniai.

5. Išvados

1. Atliekant technologijų analizę pastebėta, kad "Meta Quest 2" sistema, nors ir palaiko rankų sekimą be papildomų valdiklių, susiduriama su problemomis, kai naudojami papildomi įrenginiai ant rankų (pvz., haptinės pirštinės), uždengiama dalis rankos arba nėra optimalaus apšvietimo. Tai vienas iš uždavinių kuriuos galima spręsti naudojant įgyvendintą sistemą, kurioje prognozuojama rankos pirštų sąnariu erdvinė informacija kai nutrūksta sekimas.
2. Atliekant egzistuojančių sprendimų analizę pastebėta, kad dauguma darbų, susijusių su sąnarių pozicijų prognozėmis, orientuojasi į viso kūno ar galūnių (pvz., rankų ar kojų) judesių prognozavimą, tačiau neapima tikslaus pirštų sąnarių pozicijos ir pasukimo prognozavimo. Dažnai apsiribojama pozicijų prognozėmis, nevertinant orientacijos – tai yra būtina sąlyga siekiant įgyvendinti rankų sekimo sistemą virtualiojoje realybėje. Daugelis nagrinėtų darbų remiasi specializuota ar papildoma įranga (pvz., *OptiTrack* sistema arba kelių kamerų sprendimais), o ne integruotais VR sekimo metodais. Šie skirtumai leidžia teigti, kad įgyvendinta sistema gali spręsti išskylančias duomenų praradimo problemas naudojant pigesnę techninę įrangą.
3. Atliekant egzistuojančių sprendimų analizę pastebėta, kad dirbtinio intelekto algoritmais paremtas gestų aptikimas yra patikimas, tačiau reikalauja tinkamo algoritmo pasirinkimo ir didelio duomenų rinkinio. Plačiausiai naudojami konvoliuciniai neuroniniai tinklai (*CNN*) ir rekurentiniai neuroniniai tinklai (*RNN*).
4. Siekiant įgyvendinti komunikaciją tarp *C#* ir *Python* posistemų realiu laiku, efektyvu naudoti „Pythonnet“ paketą. Šis paketas leidžia užtikrinti ypač žemą delsos laiką, mažesnę nei 1 milisekundė atliekant abipusę komunikaciją.
5. Eksperimentų metu pastebėta, kad įgyvendintas papildomas funkcionalumas – galimybė sujungti keletą klaidos funkcijų, skirti specifinių sąnarių svorius ir skirtingus svorius pagal prognozuojamą ypatybę, kombinuojamas su modifikuotomis GRU ir LSTM architektūromis, leidžia pasiekti didesnio tikslumo.
6. Sukurta sistema, naudojant pasiūlytus metodus, gali pasiekti geriausią sąnarių pozicijos prognozės tikslumą naudojant duomenis įrenginio erdvėje, kaip įvestį naudojant ankstesnių 15 kadrų poziciją ir pasukimą, išvedant poziciją kartu su pasukimu, su LSTM 2 sluoksnių 1024 neuronų architektūra, naudojant 256 duomenų kiekį, sujungtą vidutinio Euklido atstumo ir hierarchinio pirštų atstumo klaidos funkcijas, kvaterniono kampinę klaidą, naudojant sąnarių svorius ir svorių nykimą, lygų 0,001. Tai užtikrina RMSE lygią 0,0162 m, MAE lygią 0,00567 m, vidutinį Euklido atstumą lygų 1,11 cm, užtikrinant 6,38 ms greitaveiką tarp prognozių abiem rankoms.
7. Sukurta sistema, naudojant pasiūlytus metodus, gali pasiekti geriausią sąnarių pasukimo prognozės tikslumą naudojant duomenis įrenginio erdvėje, kaip įvestį naudojant ankstesnių 15 kadrų poziciją ir pasukimą, išvedant tik pasukimą, su LSTM 2 sluoksnių 1024 neuronų architektūra, naudojant 256 duomenų kiekį, pasitelkiant kvaterniono kampinę klaidą, naudojant sąnarių svorius ir nenaudojant svorių nykimo. Ši parametrų kombinacija leidžia pasiekti 5,2 laipsnių paklaidą, užtikrinant 6,38 ms greitaveiką tarp prognozių abiem rankoms.
8. Surinktas papildytas duomenų rinkinys padėjo pasiekti geresnius prognozių rezultatus, lyginant su mažesniu duomenų rinkiniu. Įvairesni ir daugiau judesių užfiksuojantys duomenys leidžia modeliui geriau apibendrinti (angl. *generalize*) išmokstamą informaciją.
9. Įgyvendintas autoregresijos panaudojimas treniruojant modelį su papildomais metodais leidžia tiksliau prognozuoti daugiau kadrų į ateitį, lyginant su vieno kadro prognozėmis paruoštais

modeliais. Pasiekiamos vidutinės Euklido pozicijų paklaidos yra 1,47 cm, 1,62 cm, 3,36 cm ir 10,65 cm prognozuojant 1, 5, 10 ir 30 kadrų (16,5, 82,5, 165 ir 495 milisekundžių) į ateitį atitinkamai.

10. Darbo rezultatai pristatyti 30-oje IVUS 2025 (angl. *Information Society and University Studies*) tarptautinėje konferencijoje „Predicting Spatial Data of Virtual Reality Hand Tracking Joints with Recurrent Neural Networks“ publikacijoje.

Literatūros sąrašas

1. *HTC VIVE Standalone and PC VR Headsets, VR Glasses, AR, and MR* [interaktyvus]. [žiūrėta 2024-01-13]. Prieiga per: <https://www.vive.com/us/product/>.
2. *HTC VIVE Hand tracking* [interaktyvus]. [žiūrėta 2024-01-13]. Prieiga per: https://www.vive.com/ca/support/flow/category_howto/hand-tracking.html.
3. *Varjo for developers. Hand Tracking* [interaktyvus]. 2023. [žiūrėta 2024-01-13]. Prieiga per: <https://developer.varjo.com/docs/get-started/hand-tracking>.
4. *About | Ultraleap* [interaktyvus]. [žiūrėta 2024-01-13]. Prieiga per: <https://www.ultraleap.com/company/about/>.
5. *Ultraleap launches second generation of iconic hand tracking camera – Leap Motion Controller 2 | Ultraleap* [interaktyvus]. 2023 [žiūrėta 2024-01-13]. Prieiga per: <https://www.ultraleap.com/company/news/press-release/leap-motion-controller-2/>.
6. *Hand Tracking Module | Pimax* [interaktyvus]. [žiūrėta 2024-01-13]. Prieiga per: <https://pimax.com/product/hand-tracking-module/>.
7. SWEENEY, E. Oculus: Virtual reality company's complete history and device development. *Business Insider* [interaktyvus]. [žiūrėta 2024-01-13]. Prieiga per: <https://www.businessinsider.com/what-is-oculus>.
8. *Steam Hardware & Software Survey* [interaktyvus]. [žiūrėta 2024-01-13]. Prieiga per: <https://store.steampowered.com/hwsurvey/Steam-Hardware-Software-Survey-Welcome-to-Steam>.
9. *Getting started with Hand Tracking on Meta Quest headsets | Meta Store* [interaktyvus]. [žiūrėta 2024-01-13]. Prieiga per: <https://www.meta.com/help/quest/articles/headsets-and-accessories/controllers-and-hand-tracking/hand-tracking/>.
10. *Gesture Recognition: Use Cases, Technologies, and Datasets* [interaktyvus]. *Datagen* [žiūrėta 2024-01-14]. Prieiga per: <https://datagen.tech/guides/pose-estimation/gesture-recognition/>.
11. OUDAH, M. ir kt. Hand Gesture Recognition Based on Computer Vision: A Review of Techniques. *Journal of Imaging*. Basel: Multidisciplinary Digital Publishing Institute, 2020, 6(8), 73. ISSN 2313-433X.
12. LIU, X., PAN, H. The Path of Film and Television Animation Creation Using Virtual Reality Technology under the Artificial Intelligence. *Scientific Programming*. London: Hindawi, 2022, 2022, 1-8. ISSN 1058-9244.
13. *Hackster.io. Sensor glove for sign language translation* [interaktyvus]. [žiūrėta 2024-01-14]. Prieiga per: <https://www.hackster.io/ayooluwa98/sensor-glove-for-sign-language-translation-fe6980>.
14. ZHU, Y. ir kt. A Novel Wearable Soft Glove for Hand Rehabilitation and Assistive Grasping. *Sensors*. Basel: Multidisciplinary Digital Publishing Institute, 2022, 22(16), 6294 . ISSN 1424-8220.
15. SHIN, Seitaro, NAKANISHI, Yusuke, YAMAMOTO, Kohei, YOSHITAKE, Yasuhide, ir MIURA, Takahiro. A Sensor Glove for the Interaction with a Nursing-Care Assistive Robot. *2019 IEEE International Conference on Cyborg and Bionic Systems (CBS)* [interaktyvus]. New York: IEEE, 2019, 303–308 [žiūrėta 2024-01-14]. Prieiga per: doi:10.1109/CBS46900.2019.8780354.

16. High-Accuracy Sensor Glove in ROS & Gazebo [interaktyvus]. 2021. [žiūrėta 2024-01-14]. Prieiga per: <https://cps.unileoben.ac.at/high-accuracy-sensor-glove-ros-gazebo/>.
17. *Haptic Glove Market Size In 2023 : Forecasting Share and Scope for 2023-2030* [interaktyvus]. 2023 [žiūrėta 2024-01-14]. Prieiga per: <https://www.linkedin.com/pulse/haptic-glove-market-size-2023-forecasting/>.
18. *SenseGlove. VR Gloves and How They Work* [interaktyvus]. 2022. [žiūrėta 2024-01-14]. Prieiga per: <https://www.senseglove.com/a-guide-to-vr-gloves/>.
19. *Unbound XR. Top 6 Virtual Reality (VR) Handschoenen* [interaktyvus]. [žiūrėta 2024-01-14]. Prieiga per: <https://unboundxr.eu/top-vr-gloves>.
20. LEON, Jovanny, GUTIERREZ, Jairo. Real Time Hand Gesture Recognition by Skin Color Detection for American Sign Language. *2020 IEEE 17th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET)* [interaktyvus]. New York: IEEE, 2020, 82–86 [žiūrėta 2024-01-14]. Prieiga per: doi:10.1109/HONET50430.2020.9322817.
21. DAWOD, Ahmad Yahya, ABDULLAH, Junaidi, ir ALAM, Md. Jahangir. Hand Feature Detection from Skin Color Model with Complex Background [interaktyvus]. [s.l.], 2010 [žiūrėta 2024-01-14]. Prieiga per: doi:10.5176/978-981-08-7656-2ATAI2010-29.
22. NANNI, L. ir kt. A Standardized Approach for Skin Detection: Analysis of the Literature and Case Studies. *Journal of Imaging*. Basel: Multidisciplinary Digital Publishing Institute, 2023, 9(2), 35. ISSN 2313-433X.
23. LI, Y. ir kt. Gesture interaction in virtual reality. *Virtual Reality & Intelligent Hardware*. Beijing: Elsevier, 2019, 1(1), 84–112. ISSN 2096-5796.
24. PAVLOVIC, V.I. ir kt. Visual interpretation of hand gestures for human-computer interaction: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. New York: IEEE, 1997, 19(7), 677–695. ISSN 1939-3539.
25. ZHAO, L., LI, S. Object Detection Algorithm Based on Improved YOLOv3. *Electronics*. Basel: Multidisciplinary Digital Publishing Institute, 2020, 9(3), 537. ISSN 2079-9292.
26. EWE, E.L.R. ir kt. Hand Gesture Recognition via Lightweight VGG16 and Ensemble Classifier. *Applied Sciences*. Basel: Multidisciplinary Digital Publishing Institute, 2022, 12(15), 7643. ISSN 2076-3417.
27. LU, Z. ir kt. One-Shot Learning Hand Gesture Recognition Based on Lightweight 3D Convolutional Neural Networks for Portable Applications on Mobile Systems. *IEEE Access*. New York: IEEE, 2019, 7, 131732–131748. ISSN 2169-3536.
28. LIAO, S. ir kt. Occlusion gesture recognition based on improved SSD. *Concurrency and Computation: Practice and Experience*. Chichester: John Wiley & Sons, Ltd., 2021, 33(6), e6063. ISSN 1532-0634.
29. MAZHAR, O. ir kt. A Deep Learning Framework for Recognizing Both Static and Dynamic Gestures. *Sensors*. Basel: Multidisciplinary Digital Publishing Institute, 2021, 21(6), 2227. ISSN 1424-8220.

30. WONG, W.K. ir kt. Multi-Features Capacitive Hand Gesture Recognition Sensor: A Machine Learning Approach. *IEEE Sensors Journal*. New York: IEEE, 2021, 21(6), 8441–8450. ISSN 1558-1748.
31. JI, S. ir kt. 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. New York: IEEE, 2013, 35(1), 221–231. ISSN 1939-3539.
32. KARPATY, A. ir kt. Large-Scale Video Classification with Convolutional Neural Networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* [interaktyvus]. New York: IEEE, 2014, 1725–1732 [žiūrėta 2024-01-15]. Prieiga per: <https://ieeexplore.ieee.org/document/6909619>.
33. NEGIN, F., RODRIGUEZ, P., KOPERSKI, M., KERBOUA, A., GONZÁLEZ, J., BOURGEOIS, J., CHAPOULIE, E., ROBERT, P., ir BREMOND, F. PRAXIS: Towards automatic cognitive assessment using gesture recognition. *Expert Systems with Applications*. Amsterdam: Elsevier, 2018, 106, 21–35. ISSN 0957-4174. Prieiga per: doi:10.1016/j.eswa.2018.03.063.
34. MAZHAR, O. ir kt. A real-time human-robot interaction framework with robust background invariant hand gesture detection. *Robotics and Computer-Integrated Manufacturing*. Oxford: Elsevier, 2019, 60, 34–48. ISSN 0736-5845.
35. HAKIM, N.L. ir kt. Dynamic Hand Gesture Recognition Using 3DCNN and LSTM with FSM Context-Aware Model. *Sensors*. Basel: Multidisciplinary Digital Publishing Institute, 2019, 19(24), 5429. ISSN 1424-8220.
36. TANG, J., WANG, J., ir HU, J.-F. Predicting human poses via recurrent attention network. *Visual Intelligence*. Berlin: Springer Nature, 2023, 1(1), 18. ISSN 2731-9008. Prieiga per: doi:10.1007/s44267-023-00020-z.
37. ZHANG, Y. ir kt. Incorporating Physics Principles for Precise Human Motion Prediction. *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* [interaktyvus]. Waikoloa, HI, USA: IEEE, 2024, 6152–6162 [žiūrėta 2024-11-16]. Prieiga per: <https://ieeexplore.ieee.org/document/10484216>.
38. ZHANG, Binbin, ZHOU, Xuefeng, LIU, Xiangyu, ir XU, Lingyun. Dynamic Model Informed Human Motion Prediction Based on Unscented Kalman Filter. *IEEE Transactions on Human-Machine Systems* [interaktyvus]. New York: IEEE, 2022, 52(4), 725–733 [žiūrėta 2024-11-16]. ISSN 2168-2291. Prieiga per: <https://ieeexplore.ieee.org/document/9790329>.
39. CUI, Q., LIU, W., WANG, C., ir WONG, K.-Y.K. Learning Dynamic Relationships for 3D Human Motion Prediction. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* [interaktyvus]. Seattle, WA, USA: IEEE, 2020, 6518–6526 [žiūrėta 2024-11-16]. Prieiga per: <https://ieeexplore.ieee.org/document/9157765/>.
40. GAMAGE, N.M., STEINICKE, F., ir JERALD, J. So Predictable! Continuous 3D Hand Trajectory Prediction in Virtual Reality. *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST 2021)* [interaktyvus]. Virtual Event, USA: ACM, 2021, 332–343 [žiūrėta 2025-04-29]. Prieiga per: <https://dl.acm.org/doi/10.1145/3472749.3474753>.
41. *Use Fast Motion Mode | Meta Horizon OS Developers* [interaktyvus]. [žiūrėta 2025-01-13]. Prieiga per: <https://developers.meta.com/horizon/documentation/unity/fast-motion-mode/>.

42. *Requirements to use Meta Quest Link | Meta Store* [interaktyvus]. [žiūrėta 2024-06-16]. Prieiga per: <https://www.meta.com/help/quest/articles/headsets-and-accessories/oculus-link/requirements-quest-link/>.
43. *PyTorch. Start Locally* [interaktyvus]. [žiūrėta 2024-06-16]. Prieiga per: <https://pytorch.org/get-started/locally/>.
44. *Puget Systems. Hardware Recommendations* [interaktyvus]. [žiūrėta 2024-06-16]. Prieiga per: <https://www.pugetsystems.com/solutions/ai-and-hpc-workstations/machine-learning-ai/hardware-recommendations/>.
45. *GeeksforGeeks. Introduction to Recurrent Neural Network* [interaktyvus]. 2018. [žiūrėta 2024-06-18]. Prieiga per: <https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/>.
46. 46. TRIPATHI, A. What is the main difference between RNN and LSTM | NLP | RNN vs LSTM [interaktyvus]. 2021. [žiūrėta 2024-06-18]. Prieiga per: <https://ashutoshtripathi.com/2021/07/02/what-is-the-main-difference-between-rnn-and-lstm-nlp-rnn-vs-lstm/>.
47. *Python Scripting | Python Scripting | 6.0.1* [interaktyvus]. [žiūrėta 2024-06-17]. Prieiga per: <https://docs.unity3d.com/Packages/com.unity.scripting.python@6.0/manual/index.html>.
48. *gRPC* [interaktyvus]. [žiūrėta 2024-06-17]. Prieiga per: <https://grpc.io/>.

Priedai

1 priedas. Dalyvavimas IVUS 2025 (angl. *30th International Conference Information Society and University Studies*) tarptautinėje konferencijoje

Tarptautinėje konferencijoje pristatyta publikacija pavadinimu „Predicting Spatial Data of Virtual Reality Hand Tracking Joints with Recurrent Neural Networks“, paremta baigiamojo magistro darbo rezultatais.

DOKŠAS, Gytis ir BLAŽAUSKAS, Tomas. Predicting Spatial Data of Virtual Reality Hand Tracking Joints with Recurrent Neural Networks. Pristatyta IVUS 2025: 30th International Conference Information Society and University Studies. Kaunas, 2025.

2 priedas. Straipsnis ALTA 2024 (angl. *Advanced Learning Technologies and Applications*) tarptautinės konferencijos leidinyje

Išleistas straipsnis tarptautinės konferencijos leidinyje pavadinimu „Artificial Intelligence Growth in the Lithuanian Education Field“

ŠAFRANAVIČIŪTĖ, V. ir DOKŠAS G. Artificial Intelligence Growth in the Lithuanian Education Field. *ALTA 2024 – Advanced Learning Technologies and Applications: Sustainability and Partnership In Education* [interaktyvus]. Kaunas, 2024. 77-87. ISSN 2335-2140 [žiūrėta 2025-04-29]. Prieiga per internetą: <https://ndma.lt/alta2024/wp-content/uploads/2025/03/ALTA2024Proceeding.pdf>.