



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

Augustas Mikulėnas

**Mašininiu mokymusi grindžiamas naudotojo sąsajos kūrimas specialiųjų poreikių
turintiems naudotojams**

**Machine Learning Based Approach User Interface development for Users With
Special Needs**

Baigiamasis magistro darbas

Informacinių sistemų programų inžinerijos studijų programa

valstybinis kodas 6211BX017

Informatikos inžinerijos studijų kryptis

Vadovas Dr. Asta Slotkienė

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

TVIRTINU

Katedros vedėjas

(Parašas)

(Vardas, pavardė)

(Data)

Augustas Mikulėnas

**Mašininiu mokymusi grindžiamos naudotojo sąsajos kūrimas specialiųjų poreikių
turintiems naudotojams**

**Machine Learning Based Approach User Interface development for Users With
Special Needs**

Baigiamasis magistro darbas

Informacinių sistemų programų inžinerijos studijų programa

valstybinis kodas 6211BX017

Informatikos inžinerijos studijų kryptis

Vadovas

Dr. Asta Slotkienė

(Moksl. laipsnis, vardas, pavardė)

(Parašas)

(Data)

Lietuvių kalbos konsultantas Dr. Vaida Buivydienė

(Moksl. laipsnis, vardas, pavardė)

(Parašas)

(Data)

Vilnius, 2022

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINŲ SISTEMŲ KATEDRA

Informatikos inžinerija

Informacinių sistemų programų
inžinerija

TVIRTINU
Katedros vedėjas

(Parašas)

Prof. dr. Nikolaj Goranin

(Vardas, pavardė)

(Data)

**BAIGIAMOJO MAGISTRO DARBO
UŽDUOTIS**

.....Nr.

Vilnius

Studentui (ei) Augustas Mikulėnas

(Vardas, pavardė)

Baigiamojo darbo tema: Mašininiu mokymusi grindžiamas naudotojo sąsajos kūrimas specialiųjų poreikių turintiems naudotojams

Machine Learning Based Approach User Interface development for Users

With Special Needs

patvirtinta 20.. m. d. dekanų potvarkiu Nr.

Baigiamojo darbo užbaigimo terminas 2022 m. birželio 9 d.

BAIGIAMOJO DARBO UŽDUOTIS:

1. Išanalizuoti ASS turinčių naudotojų patirtį ir vartotojo sąsajų panaudojamumo ypatumus bei išanalizuoti naudotojų sąsajos kūrimo aspektus, moksliniuose tyrimuose, kuriuose yra integruojama PECS sistema;
2. Išnagrinėti mašininio mokymosi algoritmus, tinkamus dalykinei sričiai;
3. Ištirti skatinamojo mokymosi aplinkų tinkamumą tiriamai naudotojų grupės patirčiai ir panaudojamumui gerinti;
4. Pasiūlyti naudotojo sąsajos apsimokymo algoritmą.

Vadovas

.....
(Parašas)

Dr. Asta Slotkienė

.....
(Moksl. laipsnis/pedag.vardas, vardas, pavardė)

Užduotį gavau

.....
(Parašas)

.....
(Vardas, pavardė)

.....
(Data)

Padėka

Norėčiau padėkoti viso darbo metu mane konsultavusiems VŠĮ Vilniaus universiteto ligoninės Santaros klinikų Vaiko raidos centro specialistams už suteiktą vertingą informaciją, kuri leido identifikuoti autizmo spektro sutrikimą turinčių asmenų specifiką ir nustatyti šių asmenų specialiuosius poreikius. Specialistų suteikta specifinė vertinga informacija bei metodinė pagalba, vykdant darbo metu atliekamą tyrimą, leido nustatyti optimalų vykdytame tyrime naudotų kortelių ir kategorijų kiekį, galimas naudotojų reakcijas į korteles bei išgryninti labiausiai autizmo spektro sutrikimą turintiems vaikams skaitmenizuotose sistemose priimtinius informacijos pateikimo aspektus, keliamus reikalavimus tokioms sistemoms.

Taip pat dėkoju savo darbo vadovei Dr. Astai Slotkienei už nuoširdžią pagalbą bei suteiktus aiškius patarimus viso darbo metu.

Turinys

Padėka.....	4
Paveikslų sąrašas.....	7
Lentelių sąrašas.....	9
Santrumpos	10
Anotacija.....	11
Summary.....	12
Įvadas	13
1. Naudotojo sąsajos ir jo patirties sintezė.....	16
1.1. ASS turinčio asmens naudotojo patirties analizė.....	17
1.2. PECS sistemos naudojamos moksliniuose tyrimuose	20
1.3. Kortelės naudojamos PECS sistemos realizacijose	21
1.3.1. My PECS svetainėje siūlomos kortelės.....	21
1.3.2. Autismag svetainėje siūlomos kortelės	21
1.3.3. „Card Talk“ programėlėje siūlomos kortelės	22
1.3.4. „SymboTalk – AAC Talker“ programėlėje siūlomos kortelės	23
1.4. Skyriaus išvados	24
2. Mašininio mokymosi algoritmų analizė.....	25
2.1. Dirbtinio intelekto algoritmų apžvalga.....	25
2.2. Mašininio mokymosi algoritmų apžvalga	27
2.2.1. Neprižiūrimo mokymosi (angl. <i>Unsupervised learning</i>) algoritmų analizė	29
2.2.2. Skatinamojo mokymosi (angl. <i>Reinforcement learning</i>) algoritmų analizė	32
2.3. Mašininio mokymosi taikymo ASS turintiems naudotojams mokslinių tyrimų analizė.....	35
2.4. Skyriaus išvados	38
3. Naudotojo sąsajos projektavimo metodo kūrimas ir tyrimas.....	39
3.1. Projektavimo metodo kontekstas ir reikalavimai	39
3.2. Projektavimo metodo etapai	39
3.2.1. Dalykinės srities duomenų analizė.....	40
3.2.2. Duomenų rinkinio sudarymas	41
3.3. Tyrimo aprašymas su skatinamojo mokymosi algoritmais	44
3.3.1. Tyrimų aplinkos parametrų aprašymas	45
3.3.2. Skatinamojo mokymosi aplinkų analizė	48

3.3.3.	Pirmasis eksperimentinis tyrimas.....	52
3.3.4.	Antrasis eksperimentinis tyrimas	59
3.3.5.	Atliktų tyrimų rezultatai ir išvados	62
3.4.	Projektavimo metodo individualizuoto algoritmo kūrimas	62
3.4.1.	Algoritmo kūrimo eiga	63
3.4.2.	Siūlomos apsimokymo aplinkos aprašymas.....	70
3.5.	Tyrimo rezultatų apibendrinimas.....	73
4.	Išvados	74
5.	Rekomendacijos ir papildomos tyrimų kryptys	75
	Literatūros sąrašas.....	76
6.	Priedai	80
1 priedas.	Pilnas atliktų tyrimų katalogas	80
2 priedas.	MyPECS svetainėje siūlomos kortelių kategorijos	81
3 priedas.	25-ojoje Lietuvos jaunųjų mokslininkų konferencijoje „Mokslas – Lietuvos ateitis“ šio darbo tema pristatyto pranešimo pažymėjimas	83
4 priedas.	25-ojoje Lietuvos jaunųjų mokslininkų konferencijoje „Mokslas – Lietuvos ateitis“ šio darbo tema pristatyto pranešimo sertifikatas.....	84
5 priedas.	Vaiko raidos centro vadovės dr. Jovitos Petruolytės atsiliepinimas	85

Paveikslų sąrašas

1 pav. 4 pavyzdinės PECS kortelės (My PECS, 2022).....	18
2 pav. Lietuvoje platinamas PECS kortelių rinkinys (Leidykla Presvika, 2022)	19
3 pav. Autismag svetainėje pateikiamos PECS kortelių kategorijos (Autismag, 2020).....	22
4 pav. SymboTalk programėlėje siūlomų kategorijų pavyzdys (autorius daryta nuotrauka iš „SymboTalk – AAC Talker“ programėlės, 2022)	23
5 pav. Dirbtinio intelekto kategorijos (sudaryta autoriaus, remiantis (Marizel & Louella, 2018) 25	
6 pav. Skatinamojo mokymosi agento sąveika su aplinka (sudaryta autoriaus remiantis (Li, Gu, Zeng, & Feng, 2020)	33
7 pav. Skatinamojo mokymosi algoritmų grupės bei jų suskirstymas (sudaryta autoriaus remiantis (Gupta & Chaurasiya, 2019)	33
8 pav. Projektavimo etapai (sudaryta autoriaus).....	39
9 pav. Duomenų rinkinio sudarymas (sudaryta autoriaus)	41
10 pav. Agento sąveikos bei Q lentelės pavyzdys (sudaryta autoriaus remiantis (Zhao, Lu, Yang, & Guo, 2020)	42
11 pav. Tyrimo metu simuliuojama situacija (sudaryta autoriaus).....	43
12 pav. Empirinio tyrimo su skatinamojo mokymosi algoritmais etapo BPMN diagrama (sudaryta autoriaus).....	44
13 pav. Epsilon parametro keitimo logika kode (autorius rašyto kodo fragmentas)	48
14 pav. Agento veiksmo pasirinkimas kai remiamasi tik išmoktais Q lentelės duomenimis kode (autorius rašyto kodo fragmentas).....	48
15 pav. Apdovanojimų vidurkio priklausomybė nuo epizodų kiekio (tikimybė=0.73, learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, decayRate=0.00005, epizodų skaičius=250000, maxSteps=400) (sudaryta autoriaus, kinatmųjų aprašai).....	54
16 pav. Funkcija generuoti žemėlapi <i>FrozenLake</i> aplinkos kode (autorius rašyto kodo fragmentas)	56
17 pav. Apdovanojimų vidurkis lyginant su epizodų skaičiumi (tikimybė=0.91, learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, decayRate=0.00005, epizodų skaičius=250000, maxSteps=400) (sudaryta autoriaus).....	56
18 pav. Atliktų agento žingsnių kiekis testuose su idealiomis sąlygomis apmokytu agentu (tikimybė=0.91, learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, decayRate=0.00005, epizodų skaičius=250000, maxSteps=400) (sudaryta autoriaus).....	57
19 pav. Agento žingsnių pasiskirstymas lyginant su epizodų skaičiumi (tikimybė=0.91, learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, decayRate=0.00005, epizodų skaičius=250000, maxSteps=400) (sudaryta autoriaus)	58

20 pav. Agento žingsnių pasiskirstymas atsitiktiniuose 100 testų lyginant su epizodų skaičiumi (tikimybė=0.91, learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, decayRate=0.00005, epizodų skaičius=250000, maxSteps=400) (sudaryta autoriaus).....	58
21 pav. Apdovanojimų vidurkis lyginant su epizodų skaičiumi (learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, decayRate=0.00005, epizodų skaičius=250000, maxSteps=400) (sudaryta autoriaus).....	61
22 pav. <i>CardFinder</i> algoritmo sukūrimo ir testavimo diagrama (sudaryta autoriaus)	63
23 pav. 1-ojo etapo žemėlapių pavyzdys (30 % tikimybė, kad kortelė bus neigiama) (sudaryta autoriaus).....	64
24 pav. Apdovanojimų vidurkis lyginant su epizodų skaičiumi (tikimybė=0.5, learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, decayRate=0.00005, epizodų skaičius=250000, maxSteps=400) (sudaryta autoriaus).....	65
25 pav. Apdovanojimų vidurkis lyginant su epizodų skaičiumi po pakeitimų (tikimybė=0.5, learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, decayRate=0.00005, epizodų skaičius=250000, maxSteps=400) (sudaryta autoriaus)	66
26 pav. 3 – ojo etapo judėjimo pagal metaduomenis pavyzdys (sudaryta autoriaus).....	67
27 pav. Kuriamos <i>CardFinder</i> aplinkos bei agento veiksmų diagrama (sudaryta autoriaus)	71

Lentelių sąrašas

1 lentelė. Mašininio mokymosi algoritmų savybių palyginimas (sudaryta autoriaus remiantis (Ayodele, 2010) (Chan, Chan, & Yeh, 2001)	27
2 lentelė. Rekomendacinių algoritmų grupės (sudaryta autoriaus remiantis (Lokesh, 2019) (Kuźelewska, 2020)	30
3 lentelė. Susijusių autorių darbų analizė (sudaryta autoriaus)	36
4 lentelė. Konfigūruojami Q naudos funkcijos bei aplinkos parametrai bei jų reikšmės (sudaryta autoriaus).....	46
5 lentelė. Įvairių OpenAI siūlomų aplinkų kategorijos (sudaryta autoriaus).....	49
6 lentelė. OpenAI siūlomų kategorijų pritaikomumas tyrimo uždaviniams spręsti (sudaryta autoriaus).....	50
7 lentelė <i>FrozenLake</i> aplinkos testai kai bandoma kuo labiau mažinti apmokymo epizodų kiekį (tikimybė=0.91, learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.01, decayRate=0.01, maxSteps=400) (sudaryta autoriaus).....	54
8 lentelė <i>FrozenLake</i> tyrimas, bandant kuo mažiau mažinti tikimybę, kad kortelė bus teigiama. (learningRate=0.8, gamma=0.9 maxEpsilon=1, minEpsilon=0.01, decayRate=0.01, maxEpisodes=1000, maxSteps=400) (sudaryta autoriaus).....	55
9 lentelė <i>Taxi</i> aplinkos apsimokymo duomenys padidinus galimų kortelių skaičių iki 81. (learningRate=0.8 gamma=0.9 maxEpsilon=1 maxEpisodes=250000) (sudaryta autoriaus)	61
10 lentelė. Dalis tyrimų rezultatų keičiant tikimybę, kad kortelė bus teigiama (learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, decayRate=0.00005, epizodų skaičius=250000, maxSteps=400) (sudaryta autoriaus).....	67
11 lentelė. Tyrimai atlikti varijuojant epizodų kiekiu (tikimybė=0.3, learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, decayRate=0.00005, maxSteps=400) (sudaryta autoriaus)	68
12 lentelė. Tyrimai atlikti varijuojant epizodų kiekį bei epsilon vertės decayRate (tikimybė=0.3, learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, maxSteps=400) (sudaryta autoriaus)	69
13 lentelė. Tyrimai kai buvo kritiškai mažinamas epizodų kiekis (tikimybė=0.3, learningRate=0.9, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, maxSteps=400) (sudaryta autoriaus).....	70

Santrumpos

ASS – autizmo spektro sutrikimas

PECS – paveikslėlių apsikeitimo komunikacijos sistema (angl. *Picture exchange communication system*)

Vaiko raidos centras – VšĮ Vilniaus universiteto ligoninės Santaros klinikų Vaiko raidos centras

Apple – technologijų kompanija „Apple Incorporated“

DI – dirbtinis intelektas

MM – mašininis mokymasis

SARSA – „State-Action-Reward-State-Action“ funkcija

iOS – technologijų kompanijos „Apple Incorporated“ operacinė sistema

DQN – gilieji Q tinklai (angl. *Deep Q Network*)

OpenAI – dirbtinio intelekto tyrimų ir diegimo įmonė „OpenAI“

„Google Play Store“ parduotuvė – produktų programėlių platinimo sistema „Google Play Store“

Android – atvirojo kodo operacinė sistema „Android“

BPMN – verslo procesų modeliavimas ir notacija (angl. *Business Process Model and Notation*)

My PECS svetainė – PECS komunikacijos paveikslėlių platinimo svetainė mypecs.com

OS – operacinė sistema

Anotacija

Baigiamajame magistro darbe siūlomas mašininis mokymusi grindžiamas naudotojo sąsajos projektavimo būdas autizmo spektro sutrikimą turintiems naudotojams, nagrinėjama kokia seka pateikti paveikslėlių apsisikeitimo komunikacijos sistemos (angl. *Picture exchange communication system*) (toliau – PECS) korteles skaitmenizuotose PECS sistemos realizacijose.

Analitinėje darbo dalyje analizuojami šaltiniai, nagrinėjantys tikslinės auditorijos, autizmo spektro sutrikimą turinčių asmenų, patirties specifiką, naudotojų sąsajas, nagrinėjami galimi PECS kortelių rinkiniai, naudojami atliktuose moksliniuose tyrimuose bei jau veikiančiose sistemose. Siekiant pasirinkti tinkamą algoritmą darbo tyrimui, analizuojami mašininio mokymosi algoritmai bei atliekama mašininio mokymosi algoritmų analizė. Algoritmai yra lyginami vienas su kitu, analizuojama jų specifika ir galimybė juos taikyti darbo tikslui pasiekti. Atliktos analizės pagrindu sudaromi reikalavimai, kuriuos turėtų atitikti tikslinei auditorijai, autizmo spektro sutrikimą turintiems asmenims, skirtų sistemų mašininio mokymosi algoritmais paremtos aplinkos.

Tiriamąjame darbo dalyje atliekamas mašininio mokymosi algoritmų tinkamumo palyginimas. Pagal apibrėžtus reikalavimus testuojamos kitų autorių kurtos skatinamuoju mašininis mokymusi paremtos aplinkos. Remiantis darbo metu atlikta analize ir tyrimais, siūlomas skatinamojo mašininio mokymosi algoritmais paremtas aplinkos prototipas *CardFinder*. Atlikti tyrimai su sukurta apsimokymo aplinka *CardFinder* leidžia detalizuoti parametrus, kurių taikymas leidžia pasiekti tinkamą apsimokymo tikslumą – tikslo pasiekimas. Pateikiama, kokie sprendimai bei kokie testai buvo atliekami siekiant patikrinti sukurtos aplinkos efektyvumą lyginant ją su kitų autorių sukurtais aplinkomis. Darbe siūlomas skatinamojo mokymosi algoritmas paremtas projektavimo metodu, įgalina kiekvienam specialių poreikių naudotojui užtikrinanti individualizuotą komunikaciją.

Raktiniai žodžiai: skatinamasis mašininis mokymasis, naudotojo sąsajos, autizmo spektro sutrikimas, PECS.

Summary

The thesis assesses the possibility of creating a user interface using machine learning algorithms in alternative communication systems meant for children with autism spectrum disorder.

In the analytical part of the thesis, the general practices of good user interface and the target audience specifics are examined. Afterwards, publicly available scientific studies and most popular current PECS communication systems are analyzed in order to obtain details about possible initial card datasets. Various machine learning algorithms are analyzed, compared and assessed in order to figure out which algorithms could be used to solve the problems that were identified before. Then requirements for reinforced learning algorithm environments, that would be used during testing, were identified.

During the experiments various other environments created by other scientists were analyzed and some that could be used to provide solutions for questions raised during analysis were identified. Every single environment chosen was thoroughly tested for its effectiveness on solving the issues raised in the analytical part of the thesis. Using the information obtained during analysis and testing a prototype environment called *CardFinder* was created and tested. The experiments done with the *CardFinder* environment allows to determine the best parameters in order to solve the issue. Solutions, design choices and tests done are described in order to compare the effectiveness of the created environment in comparison to the best performing environments created by other authors. The reinforced learning algorithm based design method provided in the thesis would allow the system using it to provide individualized communication capabilities to every single user.

Keywords: Artificial intelligence, Machine learning, Autizm spectrum disorder, PECS, Reinforcement learning.

Ivadas

Temos aktualumas. Komunikacija yra vienas svarbiausių žmonijos aspektų. Komunikuojant yra mokomasi, bendraujama bei kitaip keičiamasi informacija. Tačiau egzistuoja asmenų grupė, kuriems nuo pat mažens kyla sunkumų išreikšti net ir paprasčiausius jausmus bei norus. Tokia asmenų grupė yra autizmo spektro sutrikimą (toliau – ASS) turintys asmenys. ASS yra spektrinis raidos sutrikimas, kuris pasižymi netipine žmogaus socialine sąveika su kitais žmonėmis, komunikacijos sutrikimais bei dažnai netipiniu pasikartojančiu elgesiu (World Health Organization, 2019). Tai reiškia, kad ASS turintys asmenys, o ypač vaikai, su juos supančiu pasauliu komunikuoja ne žodžiais, o veiksmais, pavyzdžiui verkimu, neklausymu, laimės išreiškimu bei kitomis reakcijomis.

Norint padėti ASS turintiems asmenims lengviau įsilieti į visuomenę ir išmokti bendrauti su juos supančiais asmenimis buvo sukurtos alternatyviosios komunikacijos sistemos, kurios leidžia vaikui išmokti susikalbėti naudojant įvairius paveikslėlius, simbolius ar kitas alternatyvas. 2018 metais išleistame straipsnyje pateikiama sisteminė komunikacijai naudojamų priemonių, kurios buvo sukurtos 2007–2018 metais, analizė (Mahoney, Johnson, McCarthy, & White, 2018). Tyrimo duomenys parodė, kad didžioji dalis alternatyviųjų komunikacijos sistemų, pradėtų naudoti su nepilnamečiais vaikais, padėjo jiems lengviau komunikuoti su kitais asmenimis ir net patiems pradėti inicijuoti pokalbius.

Šiuo metu viena populiariausių ir efektyviausiai taikomų tokių sistemų yra paveikslėlių apsikeitimo komunikacijos sistema (angl. *Picture Exchange Communication System*) (toliau – PECS). PECS sistemą ypač tinka taikyti vaikams, turintiems ASS. Sistema jos naudotoją susikalbėti skatina tik jo paties iniciatyva (Vicker, 2002). Sistema leidžia visiškai nekomunikuojančiam arba tik iš dalies komunikuojančiam asmeniui bendrauti pasitelkiant paveikslėlius, kurie yra rodomi vienas kitam norint apsikeisti informacija (National Autism Resources, 2022). Tad bendravimas PECS sistemoje yra atliekamas bendraujantiems dalyviams rodant vienas kitam tam tikro pavidalo korteles ir taip keičiantis informacija.

Kaip rodo autorių Slobodin, Heffler ir Davidovitch atlikto tyrimo rezultatai, ASS turintiems asmenims yra ypač svarbios naujausiosios technologijos ir jie rodo didesnę susidomėjimą jomis bei vidutiniškai prie jų praleidžia daugiau laiko nei vidutinis to paties amžiaus tipiškai besivystantis žmogus (Slobodin, Heffler ir Davidovitch, 2019).

Atsižvelgiant į minėtus aspektus, ASS turinčio vaiko komunikacijai siūloma taikyti skaitmenizuotas PECS sistemas ir ją sudarančias korteles pateikti tik su tam iš anksto pritaikyta individualizuota tvarka.

Tyrimo objektas – interaktyvios naudotojo sąsajos kūrimas integruojant PECS sistemą ASS turintiems asmenims.

Darbo tikslas – pasiūlyti naudotojo sąsajos projektavimo metodą, specialiųjų poreikių turintiems naudotojams, taikant mašininio mokymosi algoritmus.

Darbo uždaviniai:

1. Išanalizuoti ASS turinčių naudotojų patirtį ir vartotojo sąsajų panaudojamumo ypatumus bei išanalizuoti naudotojų sąsajos kūrimo aspektus, moksliniuose tyrimuose, kuriuose yra integruojama PECS sistema;
2. Išnagrinėti mašininio mokymosi algoritmus, tinkamus dalykinei sričiai;
3. Ištirti skatinamojo mokymosi aplinkų tinkamumą tiriamai naudotojų grupės patirčiai ir panaudojamumui gerinti;
4. Pasiūlyti naudotojo sąsajos apsimokymo algoritmą.

Temos naujumas. Mašininio mokymosi metodų taikymas skirtingose srityse yra vis dar populiarus bei nuolat tiriamas. Tačiau sistemų, skirtų specialiųjų poreikių turinčių asmenų komunikacijai gerinti, naudotojo sąsaja nėra aktyviai tyrinėjama bei analizuojama. Dėl tyrimų stokos nėra aišku, kaip tikslinga pateikti PECS sistemos korteles skaitmenizuotose PECS sistemos realizacijose. Siūlomas projektavimo metodas paremtas mašininio mokymosi algoritmais ir taip kiekvienam naudotojui įgalina individualizuoti komunikaciją.

Tyrimo metodika: mašininio mokymosi algoritmai, dalykinės srities duomenų analizė pasitelkiant VšĮ Vilniaus Universiteto ligoninės Santaros klinikų Vaiko raidos centro (toliau – Vaiko raidos centras) specialistus, analitinei daliai atlikti buvo naudojama su tema susijusių mokslinių straipsnių literatūros analizė, tiriamojoje dalyje buvo atliekami empiriniai tyrimai su 2 jau egzistuojančiomis OpenAI kurtomis skatinamojo mokymosi aplinkomis bei autoriaus kurta aplinka *CardFinder*.

Mokslinė darbo vertė. ASS turintiems asmenims yra ypač svarbi pagal jų poreikius adaptuota ir individualizuota naudotojų sąsaja. Šiuo metu yra vykdomi tyrimai, kurie tiria skaitmenizuotas alternatyviųjų komunikacijų sistemas bei jų veiksmingumą, tačiau nėra tiriama kaip turėtų būti pateikiama PECS sistemų naudotojo sąsaja norint siekti kuo didesnio sistemos panaudojamumo. Dėl žmonių įvairovės statinės sistemos realizacijos nėra efektyvios ir vienas būdas didinti tokių sistemų panaudojamumą yra naudoti mašininio mokymosi algoritmus. Šis darbas tiria galimybes taikyti mašininį mokymąsi PECS sistemų realizacijose bandant simuliuoti realių naudotojų elgesį.

Praktinė darbo vertė – pasiūlyta ir įgyvendinta mašininio mokymosi aplinka, leidžianti efektyviau bei tikslingiau kurti naudotojo sąsają kuriant PECS sistemos programėles.

Darbo struktūra. Darbą sudaro 5 skyriai. Pirmajame darbo skyriuje yra nagrinėjamos gerosios naudotojų sąsajos savybės bei atliekama tikslinės auditorijos patirties specifikos analizė bei analizuojami galimi alternatyviosios komunikacijos PECS kortelių rinkiniai. Antrajame skyriuje yra analizuojami mašininio mokymosi algoritmai bei jų pritaikomumas tyrimo uždaviniams spręsti. Trečiajame skyriuje yra iškeliami reikalavimai testuojamoms aplinkoms bei atlikus analizės darbus atrenkamos 2 kitų autorių kurtos aplinkos, kurios bent dalinai atitinka šiuos reikalavimus. Tuomet su minėtomis aplinkomis atliekami

eksperimentai bei testuojamas jų efektyvumas. Apibendrinus gautus tyrimų rezultatus yra kuriamas darbo autoriaus skatinamojo mokymosi aplinkos prototipas, jis lyginamas su prieš tai testuotomis aplinkomis atliekant eksperimentus bei testus. Ketvirtajame skyriuje yra pateikiamos viso darbo išvados. Penktame skyriuje yra pateikiamos tolimesnių darbų rekomendacijos bei galimos tolimesnių tyrimų kryptys.

1. Naudotojo sąsajos ir jo patirties sintezė

Teisingai sukurta naudotojo sąsaja gali lemti viso projekto naudotojų sėkmingą sistemos įsisavinimą. Remiantis 2019 metų mobiliųjų programėlių naudojimo statistika, tik 32 % visų naudotojų parsisiųstą programėlę įsijungė daugiau nei 11 kartų. Ir tik 25 % visų parsisiųstų programėlių būna įjungiamos daugiau nei vieną kartą (Clement, 2019).

Naudotojai gali programėlių nenaudoti tiek dėl nepakankamo jų funkcionalumo, tiek dėl jaučiamo nedidelio nepatogumo, apie kurį nebūtinai žino programėlės kūrėjai. Netinkamai sukurta programėlė, kuri netenkina arba netinkamai tenkina naudotojų poreikius, tiesiog yra pamirštama bei nenaudojama. Todėl svarbu programėles kurti pagal tam tikrus gerus panaudojamumo standartus bei praktikas.

Norėdama padėti programėlių kūrėjams sukurti geresnį panaudojamumą platinamose programėlėse, technologijų kompanija „Apple Incorporated“ (toliau – Apple) savo kurtoms operacinės sistemos iOS (toliau – iOS) programėlėms taiko aukštus standartus. Programėlė, kurią norima įkelti į vienintelę oficialią Apple produktų programėlių platinimo sistemą „App Store“, yra tikrinama Apple darbuotojo, kuris vertina, ar sistema atitinka sukurtus standartus (Apple, 2020):

- Estetinį vientisumą. Jis parodo, ar programėlė yra nuspėjama bei nusako, kaip gerai programėlės išvaizda ir veiksmai atitinka jos funkcionalumą.
- Nuoseklumą. Šis vertinimas užtikrina, kad norėdamas atlikti tam tikrą funkciją naudotojas, naudodamasis sistema, žinotų, ko tikėtis ir greičiausiai jau iš anksto numatytų, kaip atlikti užduotį. Kitaip tariant, yra vertinama, ar sukurtos programėlės funkcionalumas yra įgyvendintas, kaip panašių jau egzistuojančių programų. Taip pat yra tikrinama, ar programėlė funkcionuoja taip, kaip iš jos tikimasi, ar tinkamai yra naudojami panašiose programėlėse naudojami standartizuoti ženklukai, teksto stiliai ir kiti vaizdo elementai.
- Tiesioginę komponentų manipuliaciją. Ji nusako, kad programėlė turėtų aiškiai ir iš karto atvaizduoti naudotojo padarytų veiksmų pasekmes. Pvz., pavertus naudojamo prietaiso ekraną, atitinkamai turėtų sureaguoti tuo metu ekrane vaizduojamas turinys.
- Grįžtamąjį ryšį. Jis nusako programėlės galimybę informuoti naudotoją apie jo atliktus veiksmus. Remiantis tam tikrais standartais privaloma naudotojui pranešti apie mygtukų paspaudimus, išsiųstas žinutes, rodyti ilgiau trunkančių operacijų progreso būklę bei naudojant kitus telefono išteklius nuolat informuoti naudotoją.
- Metaforų naudojimą. Jis nusako, kiek įvairūs programėlėje galimi veiksmai atitinka realų pasaulį bei jau nustatytas bendrąsias taisykles. Tai yra įjungtas jungiklis turėtų būti žalias, o išjungtas – raudonas ir t. t.
- Naudotojo kontrolę. Ji nusako, ar naudotojas pats valdo programėlę bei jos veikimą. Taip pat ar programėlė neperima kontrolės iš naudotojo jam to nesitikint. Pvz., ilgų krovimo procesų metu,

sistema turėtų naudotojui suteikti galimybę procesą atšaukti, o ne versti laukti kol procesas pasibaigs.

2020 metais sukurtos gairės iš tiesų yra Nielseno bei Schneidermano charakterizuotų kriterijų interpretacija bei papildymas. Nielseno ir Schneidermano apibrėžti 5 panaudojamumo kriterijai:

- Išmokstamumas.
- Efektyvumas.
- Klaidų prevencija.
- Pasitenkinimas.
- Įsimintinumas.

Remiantis aukščiau nurodytomis charakteristikomis yra panaikinama nežinia kuriant programėles ir jei šių kriterijų yra laikomasi, tikėtina, kad ir sukurta programėlė bus sėkmingai naudojama naudotojų. Tačiau kaip matome kurtose charakteristikose, kiekviena sistema privalo būti vertinama atskirai, nes svarbūs gali būti visai kiti aspektai. Dėl šių priežasčių yra itin svarbu tinkamai pateikti kiekvienos programėlės naudotojo sąsają, o tai atlikus netinkamai, sistema tiesiog gali būti nenašė ar tiesiog nenaudinga naudotojui. Todėl yra ypač svarbu išsiaiškinti problemas su kuriomis susiduria PECS sistemų naudotojai naudodamiesi skaitmenizuotomis PECS sistemomis ir atlikti ASS turinčio asmens patirties specifikos analizę.

1.1. ASS turinčio asmens naudotojo patirties analizė

ASS yra globaliai paplitęs sutrikimas, todėl su asmenų turinčių ASS komunikacijos problemomis susiduria visos valstybės. ASS yra spektrinis raidos sutrikimas, kuris pasižymi netipiniu žmogaus bendravimu su juos supančiu pasauliu (World Health Organization, 2019). Anot Vaiko raidos centro specialistų dažniausiai ASS turintys asmenys, o ypač vaikai, su juos supančiu pasauliu komunikuoja ne žodžiais, o veiksmais, pavyzdžiui, neigiamas emocijas gali išsakyti verkimu, neklausymu bei kitomis reakcijomis. Dėl šių priežasčių nuo pat ankstyvos vaikystės vaikai, turintys ASS, turi būti mokomi bendrauti ir kalbėti. Ypač svarbus ikimokyklinis amžius, kurio metu formuojasi pagrindiniai kalbos ir bendravimo įgūdžiai. Norint padėti šiems vaikams bendrauti lengviau yra naudojamos alternatyviosios komunikacijos vizualinių simbolių sistemos.

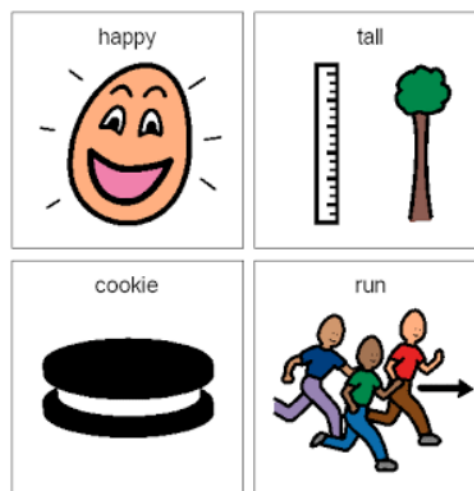
Dėl nuolatinio ASS turinčių asmenų skaičiaus augimo alternatyviosios komunikacijos sistemos tampa vis svarbesnės ASS turinčių asmenų gyvenime. Nustatyta, kad 2003 metais Kalifornijos valstijoje gimęs asmuo turi net 16,6 karto didesnę statistinę galimybę gimti su autizmo spektro sutrikimu lyginant su asmeniu, gimusiu ten pat 1992 metais (Keyes, et al., 2011). Panašios išvados padarytos ir vėlesniuose tyrimuose, nors pokytis yra nebe toks didelis. Kanadoje nustatyta, jog vidutiniškai ASS turinčių vaikų skaičius išaugo nuo 4,83 iš 1000 vaikų 2003 metais iki 17,66 iš 1000 vaikų 2015 metais. Dar naujesnėje,

2018 metų studijoje teigiama, jog 2000 metais autizmas buvo nustatomas 1 iš 150 vaikų, o 2018 metais jis yra nustatomas vienam iš 59 vaikų. Autoriai tokios statistikos augimą įvardija kaip „autizmo epidemiją“ (Baio, et al., 2018).

Ieškant užsienio šaltiniuose galima rasti įvairių alternatyviosios komunikacijos sistemų, kurių didžioji dalis yra anglų kalba. Visos populiariosios alternatyviosios komunikacijos sistemos ASS turintiems asmenims statistiškai bendrauti padeda panašiai (Mahoney, Johnson, McCarthy ir White, 2018). Tačiau dėl savo prieinamumo bei paprastumo populiariausia bei plačiausiai naudojama tokių priemonių yra PECS paveikslėlių apsikeitimo sistema. Dėl šių priežasčių būtent ji ir bus toliau analizuojama šiame darbe.

Šios sistemos realizacijų yra labai įvairių. Jų galima įsigyti internete iš populiariausių pardavimų svetainių. Produktai bei jų kaina kinta nuo nemokamų programėlių bei už kelis eurus .pdf formatu siūlomų įsigyti paveikslėlių rinkinukų iki už kelis šimtus eurų siūlomų paveikslėlių albumų ar kompiuterinių programų.

Šiuose rinkiniuose ar programose būna naudotojams pateikiamos PECS sistemos kortelės. Tipinę PECS kortelę sudaro keletas paprastų elementų – kortelės paveikslukas bei užrašas ką tas paveikslukas, o kartu ir visa kortelė, reiškia (žr. 1 pav.). Tokių kortelių rinkiniai ir yra PECS sistemos, kurių korteles naudodami ASS turintys asmenys lengviau bendrauja su aplinkiniu pasauliu.



1 pav. 4 pavyzdinės PECS kortelės (My PECS, 2022)

Vienas tokių rinkinių yra Lietuvoje naudojama Margaritos Jurevičienės ir Tomos Jokubaitienės kurta ugdymo priemonė „Bendraukime paveikslėliais“ (žr. 2 pav.). Tačiau tokie rinkiniai, anot Vaiko raidos centro specialistų, yra netobuli ir turi daug trūkumų.



2 pav. Lietuvoje platinamas PECS kortelių rinkinys (Leidykla Presvika, 2022)

Anot Vaiko raidos centro specialistų, komunikacijos sistemos yra reikalingiausios bei bendrauti skatina efektyviausiai kai ASS turintis asmuo yra ikimokyklinio amžiaus. Tačiau šio amžiaus vaikai, ypač turintys ASS sutrikimą, būna išrankūs ir gali kai kurių kortelių tiesiog nenaudoti nes jos jiems nepatinka. Kalbėdama apie kortelę ant kurios nupieštas futbolo kamuolys, o užrašyta „Kamuolys“, Vaiko raidos centro specialistė pabrėžė, kad „Autizmo spektro sutrikimą turintys vaikai yra išrankūs ir jei jiems nepatinka futbolas, jie tokios kortelės tiesiog nenaudos“. Tokiu atveju, kortelė su užrašu „Kamuolys“ tampa bevertė. Specialistė taip pat atkreipė dėmesį, jog ASS turintis vaikas gali kortelių nenaudoti netgi dėl tokių priežasčių kaip nupiešto paveiksluko spalva: „Jis (ASS turintis vaikas) mėgsta mėlynos spalvos kelnės, o rinkinyje esančioje kelnų kortelėje paveikslukas yra su žalios spalvos kelnėmis. Tokios kortelės vaikas irgi nenaudos“. Anot specialistės, būtent dėl tokio tipo priežasčių maždaug trečdalis nusipirkto rinkinio kortelių vaikui nepritaikomos ir lieka nė karto nepanaudotos. Optimalesnė skaitmenizuota PECS realizacijos sistema padėtų šias problemų išspręsti. Tokioje sistemoje būtų galima pateikti kiek norima skirtingų kortelių, tad sistemos naudotojai visada turėtų vaikui patinkančių alternatyvų. Taip pat, realizaciją įgyvendinus su apmokytu mašininio mokymosi algoritmu, sistema automatiškai galėtų nenaudojamas korteles pašalinti arba tiesiog pateikti žemiau sąraše.

Panašių problemų neišsprendžia ir skaitmenizuotos sistemos, pvz., mobiliųjų telefonų programėlės. Šiose sistemose irgi naudotojams yra pateikiamos sistemų kūrėjų sukurtos PECS kortelės ar jų rinkiniai. Tačiau paprastai jie nėra rūšiuojami pagal naudotojo kortelių naudojimą ar kitus aspektus, kuriais būtų galima užtikrinti geresnį sistemos efektyvumą. Sistema gali tapti naudotojui nepatraukli, jei naudotojui kiekvieną kartą norint pasiekti jo labiausiai naudojamas korteles ar kortelių kategorijas, pirmam būtų pateikiama trečdalis visiškai jam neaktualių kortelių ar kategorijų. Taip pat sistemos panaudojamumas tampa prastesnis įvedus galimybę naudotojui pačiam kurti norimas korteles, nes atsiradus daugiau kortelių, tikėtina, kad naudotojui reikės įdėti vis daugiau pastangų norint pasiekti korteles kurių

ieškoma. Tokį turinio perkėlimą, slėpimą bei prioretizavimą remiantis naudotojų ankstesniu sistemos naudojimu jau taiko populiarieji socialiniai tinklai, pvz., „Instagram“ (Carman, 2018). Tokių automatizuotų sistemų naudojimas kelis kartus pagerina naudotojų potyrius naudojantis sistema, nes jie greičiau mato jiems aktualų turinį.

Siekiant dar geriau suprasti PECS sistemų kortelių savybes bei rinkinius yra tikslinga išanalizuoti korteles, kurios buvo ar yra naudojamos moksliniuose darbuose bei jau egzistuojančiose, viešai prieinamose atviro kodo operacinės sistemos „Android“ (toliau – Android) programėlėse.

1.2. PECS sistemos naudojamos moksliniuose tyrimuose

Atliekant mokslinius tyrimus yra labai svarbu turėti kokybiškus pradinius duomenis. Anot mokslininkų, nekokybiški duomenys gali nulemti tyrimo rezultatus, kurie neatitinka realios situacijos, o po to, remiantis tokiais tyrimais ateityje, gali būti atliktos rimtos sprendimų klaidos (Cai & Zhu, 2015). Todėl galima manyti, kad įvairiuose moksliniuose tyrimuose, kuriuose yra naudojamos PECS kortelės, tačiau nėra tiesiogiai tiriama, kurios PECS kortelių kategorijos yra efektyviausios bei daugiausiai naudojamos, vis tiek bus kokybiškai nustatomos bei pateikiamos PECS kortelės bei kategorijos. Dažniausiai moksliniuose tyrimuose tokios kategorijos yra sudaromos arba bendradarbiaujant su specialistais, arba atliekant tikslinės auditorijos apklausas (Malandraki & Okalidou, 2007).

Atvejo studijoje, kurioje buvo tiriamas kurčio, ASS turinčio, 10 metų vaiko PECS sistemos naudojimas, buvo sudarytos vaikams aktualios kortelės, atitinkančios PECS reikalavimus (Malandraki & Okalidou, 2007). Nors tyrimo esmė buvo pačios PECS sistemos efektyvumas, tyrimo metu surinktos kortelės buvo parenkamos specialistų. Iš viso tyrime buvo pasirinktos 6 kategorijos: Maistas bei gėrimai, Žaislai, Pagrindiniai poreikiai, Mokyklos daiktai, Daiktai menams bei kūrybai ir Naudingi daiktai.

Kitoje 2 autorių N. Soomro ir S. Soomro atliktoje studijoje buvo kuriamas PECS kortelių programėlės prototipas Android OS telefonams, o po to tiriamas naudotojų pasitenkinimas naudojantis šiuo prototipu. Tyrimo pabaigoje darytoje apklausoje buvo matyti, jog kurtas programėlės prototipas pasiteisino, nes atsakymus suskirsčius į 5 grupes, nepatenkintųjų atitinkamai pagal grupes buvo 5 % pirmosios, 9 % antrosios, 6 % trečiosios, 11 % ketvirtosios ir 6 % penktosios grupės. Remiantis šiais rezultatais yra tikėtina, kad ir prototipe pateiktos kortelių kategorijos buvo pateikiamos tinkamai ir patenkino naudotojus. Šiame tyrime pateikiamos kategorijos buvo 8: Gyvūnai, Maistas, Spalvos, Formos, Vaisiai, Emocijos, Judesiai bei Daržovės (Soomro & Soomro, 2018).

Apibendrinant galima teigti, kad kitų autorių atliktuose tyrimuose naudojamų kortelių įvairovė yra išties didelė. Nebuvo pastebėta jokių kortelių ar jų kategorijų, kurios pasikartotų visuose tyrimuose. Visuose kitų autorių tyrimuose buvo remiamasi specialistų rekomendacijomis ir visi jie gavo teigiamus rezultatus, tad tikėtina, kad ir tyrimo duomenų rinkiniai, PECS kortelės, buvo teisingai surinktos. Norint patvirtinti

pastebėtus rezultatus yra tikslinga išanalizuoti korteles, kurios yra naudojamos kitų autorių kurtose PECS sistemos realizacijose.

1.3. Kortelės naudojamos PECS sistemos realizacijose

Nepastebėjus jokio ryšio tarp skirtinguose moksliniuose tyrimuose naudojamų kortelių ar jų kategorijų buvo nuspręsta analizuoti skaitmenizuotas PECS kortelių platinimo vietas. Apžvalgai buvo pasirinktos 2 svetainės, kuriose yra platinamos PECS standartus atitinkančios kortelės, bei 2 šiuo metu populiarios programėlės, platinamos produktų programėlių platinimo sistemoje „Google Play Store“ (toliau – „Google Play Store“ parduotuvė).

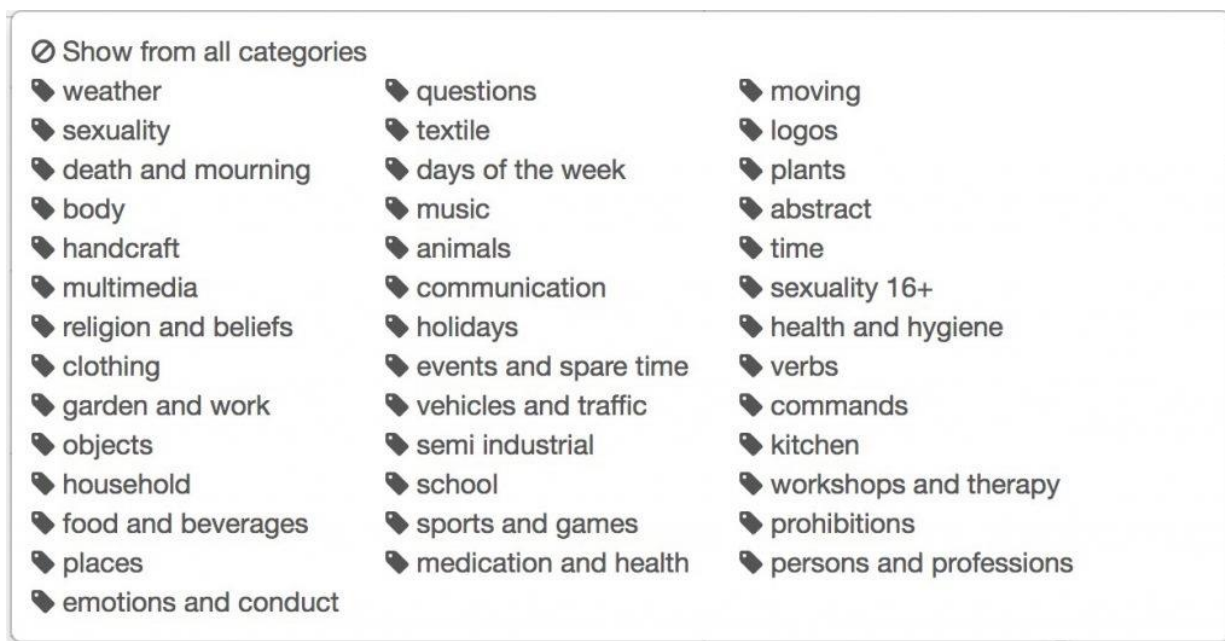
1.3.1. My PECS svetainėje siūlomos kortelės

Populiarioje PECS komunikacijos paveikslėlių platinimo svetainėje mypecs.com (toliau – My PECS svetainė) yra siūlomas didelis nemokamų paveikslėlių pasirinkimas, o visos siūlomos kortelės yra suskirstytos į kategorijas. Svetainė kaip pagrindines, ar labiausiai naudojamas, kategorijas išskiria 12 kategorijų: Veiklos, Gyvūnai, Bendrieji dalykai, Apranga, Komunikacija, Maistai, Namai, Žmonės, Vietos, Pagalba sau, Įrankiai bei Transportas. Tuomet, svetainėje siūlomos kortelių kategorijos yra skirstomos į dar mažesnes subkategorijas kurios yra pateikiamos šio darbo prieduose (žr. 2 priedas.) (My PECS, 2022).

Svetainėje siūlomos kortelės yra labai įvairiapusės. Jų gausoje tikrai būtų įmanoma surasti ir vaikams tinkamų kortelių. Tačiau taip pat yra siūlomos tokios kortelės, kaip „Karinis transportas“. Tad tikrai ne visos kortelės yra tinkamos naudoti tyrimo tikslinei auditorijai skirtose sistemose. Kita vertus, galima numanyti, kad svetainė yra labiau pritaikyta asmenims, norintiems kurti PECS korteles suaugusiems, arba tėvams, norintiems kurti korteles vaikams. Svetainėje yra siūlomas labai didelis kiekis kortelių, kurios nebūtinai yra svarbios ar naudingos kiekvienam galimos sistemos naudotojui. Tad tikėtina, kad sukūrus sistemą su kortelių paketu paremtu vien My PECS siūlomomis kategorijomis didžioji dalis kortelių būtų ištrinamos.

1.3.2. Autismag svetainėje siūlomos kortelės

Autismag, kaip ir My PECS svetainė, platina PECS korteles naudotojams. Svetainėje yra iš viso siūloma 40 kortelių kategorijų (žr. 3 pav.) (Autismag, 2020).



3 pav. Autismag svetainėje pateikiamos PECS kortelių kategorijos (Autismag, 2020)

Joje kaip ir prieš tai aptartoje My PECS svetainėje siūlomos kortelės yra labai įvairiapusiškos ir kategorijų pasirinkimas yra išties didelis, net egzistuoja tokios vaikams nepritaikytos kategorijos, kaip „seksuali veikla 16+“. Tad kaip ir My PECS svetainė, ši sistema yra labiau pritaikyta asmenims, norintiems kurti PECS korteles suaugusiems, arba tėvams, norintiems kurti korteles vaikams, atliekant kortelių paiešką. Tačiau dėl kategorijų bei kortelių gausos, kaip ir My PECS svetainėje, jei būtų sukurta sistema su kortelių paketu paremtu vien Autismag siūlomomis kategorijomis didžioji dalis kortelių būtų ištrinamos.

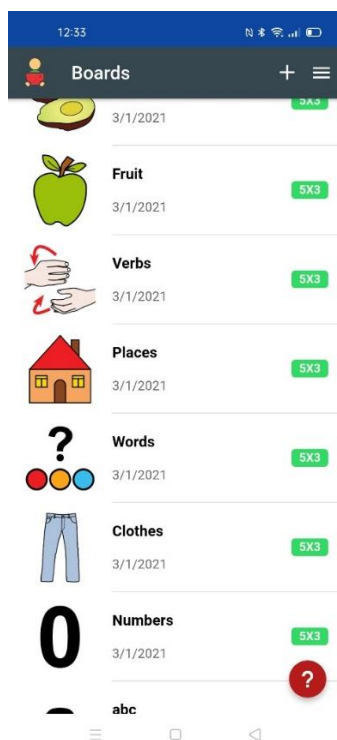
1.3.3. „Card Talk“ programėlėje siūlomos kortelės

„Card Talk“ programėlė yra šiuo metu viena populiariausių bei geriausiai vertinamų nemokamų PECS komunikacijos programėlių platinamų „Google Play Store“ parduotuvėje. Jos įvertinimas siekia 4,5 iš 5 žvaigždučių ir naudotojai ją yra atsisiuntę virš 50 tūkstančių kartų.

Šioje sistemoje yra siūloma 14 sistemos autorių sukurtų kategorijų: Norai, Įrankiai, Maistas, Žmonės, Gyvūnai, Kūnas, Apranga, Transportas, Laikas, Jausmai, Veiksmai, Santykiai ir Klausimai. Kiekvienoje kategorijoje yra siūloma apie keliolika su kategorija susijusių kortelių. Šios kategorijos, kaip ir pati programėlė, yra pritaikyta naudoti jaunesnio amžiaus naudotojams. Tad tokia programėlė iš esmės tinka ir tyrimo tikslinei auditorijai. Tačiau atsižvelgiant į tai, kad dalis kategorijų pateikia išties sudėtingas korteles, pvz., klausimų kategorijoje yra sudėtingų klausimų, kuriuos, tikėtina, kad naudotojas panaudos labai retai. Tokių kortelių šalinti ar kitaip deaktivuoti sistemoje negalima. Taip pat sistemos naudotojas turi galimybę programėlėje pats kurti korteles, tad kortelių kiekis teoriškai nėra ribojamas. Atsižvelgus į šias savybes galima teigti, kad „Card Talk“ programėlėje gali būti neribotas kiekis kortelių, kurios nėra niekaip rikiuojamos ar filtruojamos, tad tokiais atvejais pačios programėlės panaudojamumas nėra geras.

1.3.4. „SymboTalk – AAC Talker“ programėlėje siūlomos kortelės

„SymboTalk – AAC Talker“ programėlė (žr. 4 pav.), kuri yra platinama „Google Play Store“ parduotuvėje siūlo mažiau kategorijų nei „Card Talk“ programėlė. Šios sistemos įvertinimas siekia 4,3 iš 5 žvaigždučių ir naudotojai ją yra atsisiuntę virš 10 tūkstančių kartų.



4 pav. SymboTalk programėlėje siūlomų kategorijų pavyzdys (autorius daryta nuotrauka iš „SymboTalk – AAC Talker“ programėlės, 2022)

Programėlė siūlo tokias kategorijas: Pagrindiniai daiktai, Jausmai, Spalvos, Maistas, Žaislai, Daržovės, Vaisiai, Veiksmažodžiai, Vietos, Žodžiai, Apranga, Skaičiai, Abėcėlė ir Žmonės. Kiekvienoje kategorijoje yra siūloma keliolika kortelių susijusių su ta kategorija. Nors sistemoje siūloma gana daug kategorijų, nėra tiksliai aišku, kam sistemoje reikalinga kategorija Žodžiai, nes remiantis PECS praktikomis viena kortelė jau pati atitinka vieną veiksmą, daiktą ar žodį. Taip pat sistemoje nėra galimybės slėpti ar trinti kortelių, tad naudojant sistemą naudotojui gali būti rodomos jam asmeniškai neaktualios kortelės. Kita vertus, pačios pateikiamos kategorijos yra ganėtinai įvairiapusės ir pritaikomos naudoti vaikams, o tuo pačiu ir tyrimo tikslinei auditorijai.

1.4. Skyriaus išvados

Apibendrinant ištirtą naudotojo sąsają bei tikslinės auditorijos naudotojo patirties sintezę galima išskirti žemiau pateiktas išvadas:

Išanalizavus tikslinės auditorijos specifiką nustatyta, kad PECS sistemose yra svarbu kiekvienam naudotojui individualiai nustatyti kortelių kategorijų svarbą, kortelių pateikimo eiliškumą ir kai kurių kortelių naikinimą arba ignoravimą.

Išanalizavus literatūrą, mokslinius darbus, sukurtas populiariausias skaitmenizuotas PECS sistemos realizacijas ir konsultuojantis su Vaiko raidos centro specialistais buvo nustatyta, jog norint užtikrinti teigiamą tikslinės auditorijos naudotojų patirtį yra svarbu, kad skaitmenizuota PECS sistema būtų optimizuota ir pateiktų kiekvienam naudotojui svarbią informaciją kuo kokybiškiau, nes anot specialistų ASS turintys vaikai yra išrankūs ir gali nenaudoti arba greitai nustoti naudoti sistemą vien todėl, kad ji jiems pakankamai greitai nepateikia reikalingos informacijos.

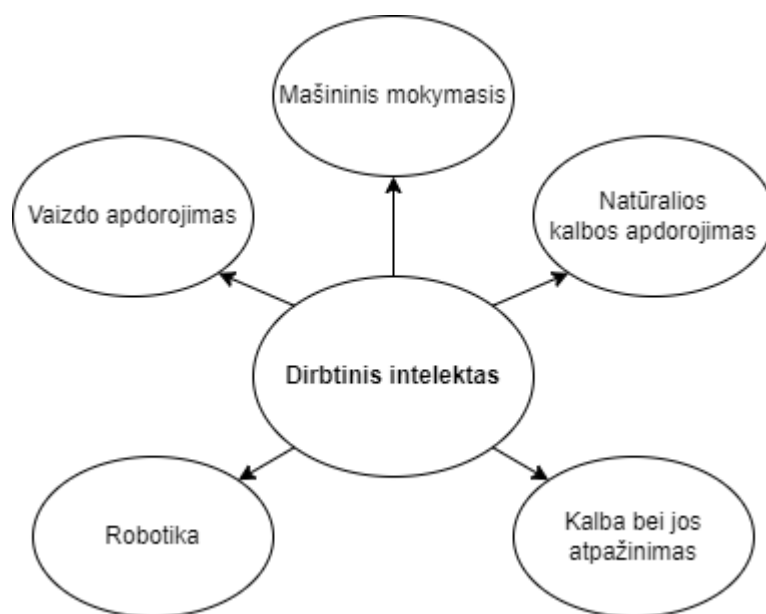
Išanalizavus kituose moksliniuose tyrimuose taikomas PECS korteles, taip pat PECS korteles naudojamas egzistuojančiose sistemose nebuvo atrasti jokie pastebimi kortelių, kurios yra naudojamos skirtinguose tirtuose objektuose, panašumai. Konsultuojantis su Vaiko raidos centro specialistais buvo nuspręsta, kad tyrimo metu yra optimalu taikyti 81 kortelę, suskirstytą į 9 skirtingas kategorijas. Tačiau atsižvelgiant į tai, kad kiekvienas vaikas yra unikalus bei egzistuoja PECS sistemos realizacijų, kuriose patys naudotojai gali kurti korteles, mašininio mokymu paremtos sistemos turi gerai veikti ir su gerokai didesniu ar mažesniu kiekiu kortelių.

2. Mašininio mokymosi algoritmų analizė

Mašininis mokymasis yra neatsiejama dirbtinio intelekto dalis. Šiame skyriuje analizuojamas dirbtinis intelektas, jo apibrėžimai, savybės bei galimas pritaikymas siekiant spręsti tyrimo iškeltus uždavinius.

2.1. Dirbtinio intelekto algoritmų apžvalga

Dirbtinis intelektas (toliau – DI) yra žmogaus intelekto imitavimas kompiuteriuose (Nilsson, 2009). Šių algoritmų esmė yra išmokyti kompiuterį kuo kokybiškiau mąstyti ir priimti tokius pat sprendimus kaip žmogus. Tai gebėdami atlikti kompiuteriai galėtų efektyviai žmonėms padėti daugelyje aktualių sričių. Tačiau norint išsiaiškinti ar tyrimo užduotims spręsti ši technologija yra tinkama būtina analizuoti DI kategorijas.



5 pav. Dirbtinio intelekto kategorijos (sudaryta autoriaus, remiantis (Marizel & Louella, 2018))

2 autorių Marizel ir Louella atliktas tyrimas apibrėžė 5 pagrindines DI kategorijas (žr. 5 pav.). Kiekviena jų yra analizuojama žemiau:

- Vaizdo apdorojimas – tai yra dirbtinio intelekto sritis leidžianti kompiuteriams bei kompiuterinėms sistemoms surinkti bei prasmingai apdoroti informaciją iš skaitmeninių vaizdų ar nuotraukų (Shapiro & Stockman, 2001). Šios kategorijos metodai darbe nebus analizuojami, nes šio darbo tyrimas nenaudos skaitmeninių vaizdų kaip įėjties duomenų, kurie teikia informaciją sprendimams atlikti.
- Robotika – tai yra dirbtinio intelekto sritis, kurioje yra analizuojamos bei kuriamos fizinės mašinos, kurios bando atlikti tam tikras užduotis (Murphy, 2019). Šios kategorijos metodai

darbe nebus analizuojami, nes šio darbo tyrimas nenaudos ar neanalizuos fizinių mašinų, robotų ar kitų panašių fizinių įrenginių.

- Kalba bei jos atpažinimas – tai yra dirbtinio intelekto sritis leidžianti kurti, interpretuoti bei kitaip apdoroti žmogaus sakomą kalbą (Marizel & Louella, 2018). Pvz., bandyti iš teksto generuoti kalbą bei atvirkščiai. Šios kategorijos metodai darbe nebus analizuojami, nes šio darbo tyrimas nenaudos ar neanalizuos teksto ar šnekamosios kalbos.
- Natūralios kalbos apdorojimas – tai yra dirbtinio intelekto sritis bandanti interpretuoti pačią kalbą (Chowdhary, 2020). T. y. bandanti generuoti sakinius, sistemizuoti tekstus, sakinių struktūras, generuoti vertimus bei kitas kalbos dalis. Šios kategorijos metodai darbe nebus analizuojami, nes šio darbo tyrimas nenaudos ar neanalizuos teksto ar šnekamosios kalbos.
- Mašininis mokymasis – dirbtinio intelekto sritis, leidžianti kompiuterinėms sistemoms mokytis iš pateiktų bei prisitaikyti prie naujų duomenų ar situacijos be jokio žmogaus įsikišimo (Jordan & Mitchell, 2015). Šio tipo algoritmų esmė yra racionalizuoti informaciją bei kuo tiksliau bei greičiau atlikti tikslo link vedančius skaičiavimus. Gerai apmokyti algoritmai nuolat teikia prioritetą sprendimams, kurie turi didžiausią tikimybę pasiekti teigiamą užduoties rezultatą.

Iš aukščiau pateiktos informacijos galima teigti, kad tyrimui geriausiai tinka naudoti mašininio mokymosi srities dirbtinį intelektą, nes šio tipo algoritmai leidžia užtikrinti optimalų užduoties sprendimą bei pritaikomumą tyrimo uždaviniams spręsti. Norint iš šios kategorijos pasirinkti tinkamą algoritmą darbo tyrimo uždaviniams spręsti reikia analizuoti mašininio mokymosi algoritmus.

2.2. Mašininio mokymosi algoritmų apžvalga

Mašininio mokymosi algoritmų pasirinkimas yra išties didelis. Norint šiuos algoritmus kiek struktūrizuoti buvo apibrėžtos 4 algoritmų grupės pagal jų savybes bei veikimo principą. Populiariausius ir dažniausiai naudojamus algoritmus galima būtų suskirstyti į šias grupes (sudaryta autoriaus remiantis (Ayodele, 2010)):

- Pilnai prižiūrimas mokymasis (angl. *Supervised learning*) – pati populiariausia mašininio mokymosi algoritmų grupė, kai algoritmas yra apmokomas su teisingus atsakymus turinčiais uždaviniais.
- Pusiau prižiūrimas mokymasis (angl. *Semi-supervised learning*) – tai yra tarpiniai algoritmai tarp pilnai prižiūrimų ir neprižiūrimų tipų, kai mokymų metu yra pateikiami tiek uždaviniai su teisingais atsakymais, tiek uždaviniai be atsakymų.
- Skatinamasis mokymasis (angl. *Reinforcement learning*) – yra tokie algoritmai, kurie mokymosi metu yra apdovanojami už teisingą bei baudžiami už klaidingą uždavinio sprendimą.
- Neprižiūrimas mokymasis (angl. *Unsupervised learning*) – yra tokie algoritmai, kurie yra apmokomi jiems nepateikiant jokių užuominų, koks atsakymas yra teisingas. Vienas populiariausių pavyzdžių yra „clustering“ tipo algoritmai, kurie skirsto tam tikrus daiktus pagal tam tikras reikšmes. Šie algoritmai ypač dažnai yra naudojami rekomendacijų sistemose.

Žemiau lentelėje pateikiamas esminių populiariausių mašininio mokymosi algoritmų savybių palyginimas :

1 lentelė. Mašininio mokymosi algoritmų savybių palyginimas (sudaryta autoriaus remiantis (Ayodele, 2010) (Chan, Chan, & Yeh, 2001))

	Pilnai prižiūrimas mokymasis	Pusiau prižiūrimas mokymasis	Skatinamasis mokymasis	Neprižiūrimas mokymasis
Pradiniai duomenys	Teisingus atsakymus ar sprendimus turintys duomenys	Tiek teisingus atsakymus turintys sužymėti duomenys, tiek generuoti nežymėti duomenys	Neapdoroti, nežymėti duomenys	Neapdoroti, nežymėti duomenys

	Pilnai prižiūrimas mokymasis	Pusiau prižiūrimas mokymasis	Skatinamasis mokymasis	Neprižiūrimas mokymasis
Rezultatų tikslumas bei pastovumas	Tikslus bei aiškus vienas rezultatas, kurio tikėjosi programuotojas	Tikslus vienas rezultatas, kurio tikėjosi programuotojas	Tikslus vienas rezultatas, kurio tikėjosi programuotojas prieš atlikdamas užduotį	Nėra aišku, koks rezultatas bus pasiektas, gali būti visai kitaip nei tikėtasi tyrimo pradžioje
Pagrindinis panaudojimas	Rizikos įvertinimas, pardavimų numatymas, paveikslėlių atpažinimas	Garso ir vaizdo manipuliavimas, duomenų kūrimas, natūralios kalbos atpažinimas	Darbų optimizavimas, parametrų numatymas bei automatizavimas	Pardavimų numatymas, paieškos bei atitikimų, panašumo nustatymas

Atsižvelgiant į tai, kad tyrimo aplinkoje kiekvienas naudotojas gali turėti skirtingus lūkesčius bei naudoti iš esmės skirtingas korteles, todėl pati sistema turėtų galėti prisitaikyti prie kiekvieno naudotojo individualiai. Taip pat šio darbo tyrimo tikslinė auditorija yra itin socialiai jautri, o anot Vaiko raidos centro specialistų, ASS turintys vaikai yra išrankūs ir gali nenaudoti arba greitai nustoti naudoti sistemą vien todėl, kad ji jiems pakankamai greitai nepateikia norimos informacijos. Dėl šios priežasties yra būtina atsižvelgti į įvairias tik šiai auditorijai reikšmingas problemas. Tai įvertinus nustatyta, kad nebe visi algoritmai yra tinkami taikyti būtent šiai tikslinei auditorijai. Remiantis aukščiau pateikta palyginimų lentelė galima nustatyti kiekvieno algoritmo tipo tinkamumą būtent šiam specifiniam uždaviniui spręsti (sudaryta autoriaus):

- Pilnai prižiūrimas mokymasis – šie algoritmai šiam tyrimui netinka, nes kiekvienas naudotojas gali norėti naudoti skirtingas kortelių kategorijas ar korteles, tad vieno teisingo atsakymo niekada neįmanoma apibrėžti. O teisingais atsakymais pažymėti uždaviniai yra būtini norint tinkamai apmokyti šį algoritmą.
- Pusiau prižiūrimas mokymasis – šie algoritmai šiam tyrimui netinka, nes kiekvienas naudotojas gali norėti naudoti skirtingas kortelių kategorijas ar korteles, tad vieno teisingo atsakymo niekada neįmanoma apibrėžti. O teisingais atsakymais pažymėti uždaviniai yra būtini norint tinkamai apmokyti šį algoritmą

- Skatinamasis mokymasis – šiame tyrime nėra nei teisingų nei klaidingų atsakymų, nes kiekvienas naudotojas gali naudoti skirtingas korteles. O šie algoritmai yra puikiai pritaikyti situacijoms, kai tarp skirtingų žmonių atsakymai gali labai skirtis. Taip pat šie algoritmai puikiai atlieka skaičiavimus ir gyvai prisitaiko prie kiekvienos situacijos net ir turint itin mažą duomenų kiekį.
- Neprižiūrimas mokymasis – šie algoritmai yra apmokomi iš anksto nepateikiant jokių užuominų koks atsakymas yra teisingas, o šiame tyrime teisingi atsakymai iš anksto nėra aiškūs. Tačiau kiekvienas žmogus yra skirtingas ir tyrime yra bandoma įvertinti galimą žmogaus emociją.

Atlikus algoritmų tinkamumo palyginimą nustatyta, kad anksčiau apibrėžtus šio tyrimo tikslus turėtų būti galima pilnai pasiekti naudojant skatinamojo mokymosi (angl. *Reinforcement learning*) arba neprižiūrimo mokymosi (angl. *Unsupervised learning*) algoritmus. Žemiau yra pateikta kiekvieno šių algoritmų tinkamumo tyrimo situacijoje analizė.

2.2.1. Neprižiūrimo mokymosi (angl. *Unsupervised learning*) algoritmų analizė

Labiausiai tinkantys tyrimo užduočiai spręsti neprižiūrimo mokymosi algoritmai yra rekomendaciniai. Rekomendacijos tokių algoritmų sistemose yra pateikiamos remiantis mašininio mokymosi algoritmais. Realizavus tyrimo sistemą su šio tipo algoritmu, skaičiavimų metu atitinkamai būtų bandoma spėti, kurias korteles naudotojas yra linkęs labiau naudoti ir tokioms kortelėms būtų suteikiamas geresnis įvertinimas. Galiausiai pagal šį įvertinimą bei kitus spėjimus būtų bandoma išrūšiuoti visas korteles sąrašė. Tokiems algoritmams yra duodami dideli įeities duomenų rinkiniai. Tuomet algoritmai yra jais apmokomi ir galiausiai atitinkamai nuo jau specifinio naudotojo pasirinkimų yra generuojami rekomendacijų rezultatai (Ghahramani, 2003).

Populiariausius rekomendacinius variklius galima būtų suskirstyti į keletą grupių pagal jų turinio filtravimo principus:

- Filtravimas pagal turinį (angl. *Content-based filtering*) – šis filtravimo būdas yra pagrįstas objektų aprašymais arba tam tikrais objekto metaduomenimis. Sistema randa objektų panašumus pagal jų kontekstą. Tuomet yra atsižvelgiama į ankstesnę sistemos naudojimo istoriją, ieškant objektų, kurie galimai patiks kiekvienam naudotojui. Pvz., priskyrus keliems filmams veiksmo žanrą ir pastebėjus, kad naudotojui patinka veiksmo filmai, jam toliau būtų rekomenduojama pateikti daugiau veiksmo žanro filmų (Pal, Parhi, & Aggarwal, 2017).
- Kolektyvinis filtravimas (angl. *Collaborative filtering*) – šis filtravimo būdas yra pagrįstas naudotojo elgesiu sistemoje. Ypač svarbų vaidmenį atlieka naudotojų veiksmų istorija. Pavyzdžiui, jei naudotojui „1“ patinka atlikėjai „A“, „B“ ir „C“, o naudotojui „2“ patinka „A“, „B“ ir „D“, tuomet jų požiūris, greičiausiai, yra panašus. Tai pastebėjęs algoritmas pateiktų

rezultatus, kad yra didelė tikimybė, jog naudotojui „1“ patiks atlikėjas „D“, o naudotojui „2“ – atlikėjas „C“. Kolektyvinis filtravimas dar yra skirstomas į 2 skirtingas kategorijas pagal tai kuris objektas yra lyginamas su kuriuo (Doshi, 2019):

- Naudotojas–Naudotojas kolektyvinis filtravimas (angl. *User-User collaborative filtering*) – šio tipo skaičiavimuose yra lyginami naudotojų vektoriai. Kiekvienas naudotojo vektorius apima visus objektus su kuriais atliko veiksmus naudotojas bei kiekvieno konkretaus objekto įvertinimą pagal atliktus veiksmus. Tuomet naudotojai yra lyginami vienas su kitu ir apskaičiuojamas panašumas tarp jų (Pujahari & Padmanabhan, 2015). Galiausiai atsižvelgiant į panašumą tarp naudotojų yra siūlomos objektų rekomendacijos.
- Objektas–Objektas kolektyvinis filtravimas (angl. *Item-Item collaborative filtering*) – šio tipo skaičiavimuose, priešingai nei prieš tai aptartame filtravime, yra lyginami ne naudotojai, o patys objektai su kuriais veiksmus atlieka naudotojai (Pujahari & Padmanabhan, 2015). T. y. stebima su kuriais objektais kurie naudotojai atliko veiksmus, tuomet pagal tai yra ieškoma panašumo tarp objektų. Apskaičiavus panašumus yra siūlomos objektų rekomendacijos .

Visos aukščiau aptartos rekomendacijų algoritmų grupės turi savitų pranašumų bei trūkumų. Šie yra pateikiami žemiau esančioje lentelėje (žr. 2 lentelė).

2 lentelė. Rekomendacinių algoritmų grupės (sudaryta autoriaus remiantis (Lokesh, 2019) (Kuzelewska, 2020)

	Filtravimas pagal turinį	Kolektyvinis filtravimas	
		Naudotojas– Naudotojas	Objektas–Objektas
Pranašumai	Naudotojas, kaip rekomendacijas gaus objektus, kurie yra jam patinkančio tipo. Skaičiavimams atlikti yra reikalingi tik to objekto duomenys.	Naudotojui yra dažnai siūlomi nauji objektai, kurie galimai jam patiks. Atliekami skaičiavimai dažniausiai yra paprastesni nei filtruojant pagal turinį.	

	Filtravimas pagal turinį	Kolektyvinis filtravimas	
		Naudotojas– Naudotojas	Objektas–Objektas
Trūkumai	<p>Naudotojui yra sunku išbandyti naujo tipo objektus, nes jam jie nėra rekomenduojami.</p> <p>Jei naudotojo arba objektų matrica pakinta, panašumo matricą reikia skaičiuoti iš naujo.</p> <p>Yra reikalingas pakankamai didelis kiekis naudotojo naudojimo duomenų norint pateikti teigiamus rezultatus.</p> <p>Kuriant naujus objektus kiekvienam objektui yra būtina priskirti struktūrizuotus metaduomenis, kurie jau yra naudojami panašiuose objektuose.</p>	<p>Norint atlikti tikslius skaičiavimus yra reikalingi visų naudotojų objektų įvertinimo istorija.</p> <p>Naujo objekto negalima rekomenduoti, jei joks naudotojas jo neįvertino.</p> <p>Yra reikalingas didelis pradinių duomenų kiekis norint, kad algoritmas veiktų teisingai.</p>	<p>Norint atlikti tikslius skaičiavimus yra reikalinga turėti visų naudotojų per tam tikrą laiką atliktus veiksmus su objektu.</p> <p>Yra reikalingas didelis pradinių duomenų kiekis norint, kad algoritmas veiktų teisingai.</p>

Apibendrinant aukščiau pateiktos lentelės duomenis galima įvertinti kiekvienos algoritmų grupės tinkamumą šio darbo tyrimui. Kiekvienos grupės tinkamumas yra pateikiamas žemiau (sudaryta autoriaus):

- Filtravimas pagal turinį - projekte šios grupės algoritmai dalinai atitiktų išsikeltus reikalavimus, nes apmokytas algoritmas pateiktų kiekvienam naudotojui galimai patinkančias rekomendacijas. Tačiau neatsižvelgtų į paties naudotojo išskirtinius pomėgius, o tiesiog siūlytų korteles pagal iš anksto apibrėžtus metaduomenis. Taip pat, kol vartotojo naudojimo duomenų būtų nedaug, jam pateikiami rekomendacijų rezultatai galėtų labai skirtis nuo tų, kuriuos galimai nori matyti naudotojas. Kita vertus, šios grupės algoritmų skaičiavimams atlikti yra reikalingi tik objekto metaduomenys ir nėra reikalinga didelė kitų vartotojų sistemos naudojimo istorija. Tačiau kuriant naujus objektus kiekvienam objektui yra būtina priskirti metaduomenis, ką naudotojas darydamas pats nebūtinai gali daryti tinkamai.
- Kolektyvinis filtravimas (Naudotojas – Naudotojas bei Objektas – Objektas) – šių grupių algoritmai dalinai atitinka išsikeltus reikalavimus, nes apmokytas algoritmas pateiktų

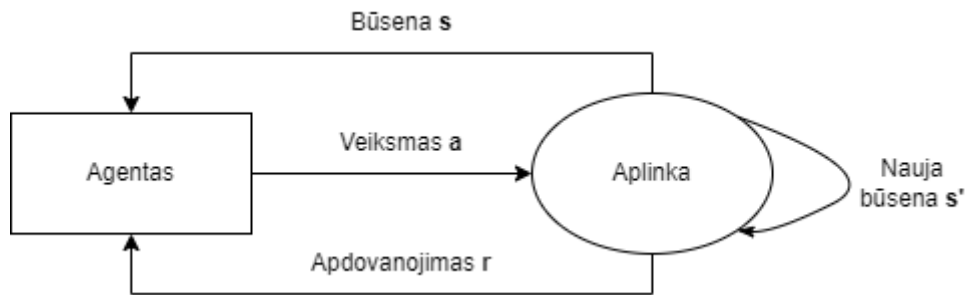
kiekvienam naudotojui galimai patinkančias rekomendacijas. Tačiau tiksliais skaičiavimams atlikti yra reikalinga kitų naudotojų sistemos naudojimo istorija, o naudotojų duomenų sistemoje apmokymo metu bus itin nedaug, tad šie algoritmai tyrimo situacijoje yra netinkami.

Remiantis aukščiau pateikta informacija tyrimui labiausiai tinkamas algoritmas būtų iš „Filtravimo pagal turinį“ grupės. Tačiau būtina atsižvelgti į tai, kad tyrimas yra atliekamas nišinėje, socialiai jautrių asmenų auditorijoje, o vienas iš esminių tyrimo aspektų yra tai, kad sistema turi tiksliai prisitaikyti prie kiekvieno naudotojo individualių poreikių. Atsižvelgiant į vieną iš šios grupės trūkumų matoma, kad tokioje sistemoje naudotojui būtų sunku pamatyti naujo tipo korteles, nes jam jos tiesiog nebūtų rekomenduojamos. O naudotojas gali turėti individualių pomėgių ir norėti matyti kortelę iš, anot algoritmo, netinkamų kortelių kategorijų.

Taip pat tyrimui yra reikalingas algoritmo sprendimas, kuris veiktų gerai su ypač mažu pradinių duomenų kiekiu, tačiau vis tiek sėkmingai bei tinkamai prisitaikytų prie kiekvieno sistemos naudotojo. Todėl, vien pritaikius rekomendacinį algoritmą visų naudotojų surinktiems duomenims, problema būtų išspręsta tik dalinai. O dėl kiekvieno ASS turinčio vaiko individualumo yra reikalingas sprendimas, kuris tuo pačiu metu pats apsimokytų pagal kiekvieną naudotoją. Būtent tokias situacijas geriausiai sprendžia skatinamojo mašininio mokymosi algoritmai.

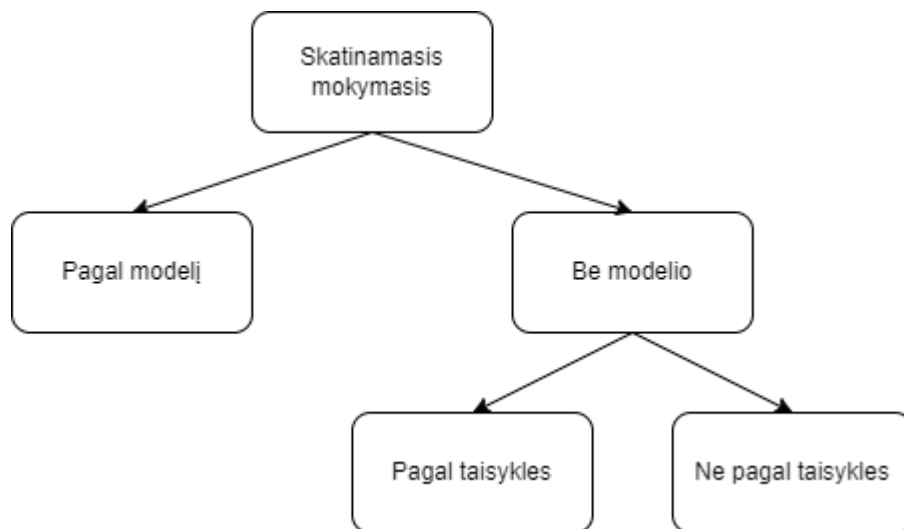
2.2.2. Skatinamojo mokymosi (angl. *Reinforcement learning*) algoritmų analizė

Skatinamasis mokymasis (angl. *Reinforcement learning*) – mašininio mokymosi algoritmų tipas. Reikšmingas skirtumas tarp skatinamojo mokymosi ir kitų mašininio mokymosi algoritmų yra tas, kad skatinamojo mokymosi sistemose nėra aiškių išankstinių mokymosi duomenų. Visi duomenys yra generuojami sistemos agentui sąveikaujant su tiksliai neapibrėžta ar potencialiai sudėtinga tyrimo aplinka (Ding, Zhao, Xu, Sun, & Jia, 2019). Šie algoritmai yra pagrįsti agento mašinos būsenomis bei jų kitimu, kai sistema bando įvertinti kiekvieno aplinkoje galimo atlikti veiksmo efektyvumą tam tikroje būsenoje (Wenlin & Jingkai, 2020). Skatinamojo mokymosi agentas veikia tyrimo aplinkoje tyrinėjimo principu, kai agentas pasirenka veiksmą, kurį nori atlikti tyrimo aplinkoje. Tuomet atlikus minėtą veiksmą agento būseną pakinta iš pradinės būsenos (s) į galutinę būseną (s'). Atitinkamai pagal atlikto veiksmo rezultatą agentas yra teigiamai arba neigiamai apdovanojamas. Galiausiai agentas renkasi naują veiksmą pagal jau naują pasiektą būseną (s') bei gautą apdovanojimą (r). Šis procesas yra pavaizduotas paveikslėlyje žemiau (žr. 6 pav.).



6 pav. Skatinamojo mokymosi agento sąveika su aplinka (sudaryta autoriaus remiantis (Li, Gu, Zeng, & Feng, 2020))

Skatinamojo mokymosi algoritmus pagal savybes bei veikimo principą galima skirstyti į 2 grupes: be modelio (angl. *Model-free*) ir pagal modelį (angl. *Model-based*). Be modelio grupę dar galima skirstyti į pogrupius: pagal taisykles (angl. *Policy*) ir ne pagal taisykles (angl. *Off-Policy*) (Kung-Hsiang, 2018). Kaip šios grupės susijusios tarpusavyje pavaizduota paveikslėlyje žemiau (žr. 7 pav.).



7 pav. Skatinamojo mokymosi algoritmų grupės bei jų suskirstymas (sudaryta autoriaus remiantis (Gupta & Chaurasiya, 2019))

Modeliais paremti algoritmai bando simuliuoti testuojamos aplinkos dinamiką. Tai yra modelis bando atitinkamai išmokti kaip dažnai arba kokia tikimybe turėtų kažkokių veiksmu iš vienos agento mašinos būsenos pereiti prie kitos. Tinkamai apmokytas modelis simuliacijos metu remiasi šiomis tikimybėmis atlikdamas visas užduotis. Tačiau tokio pobūdžio algoritmai pasidaro gerokai mažiau efektyvūs, kai didėja būsenų arba pačių veiksmų skaičius (Kung-Hsiang, 2018).

Algoritmai paremti nemodelinėmis sistemomis mokosi darydami atsitiktinius veiksmus, net ir tokius, kurie gali turėti neigiamų padarinių. Tuomet modelis padaręs klaidą bando šio veiksmo vengti. Tokie algoritmai taupo vietą bei kompiuterio resursus, nes nėra būtina turėti visų būsenų bei visų galimų

veiksmų duomenų (Kung-Hsiang, 2018). Kadangi realioje tyrimo situacijoje kortelių kiekis gali būti kintantis, gali būti situacijų, kai algoritmai be modelio dėl savo savybių tampa gerokai efektyvesni, nei algoritmai pagal modelį. Todėl toliau bus analizuojami tik algoritmai be modelio.

Pagal taisykles (angl. *policies*) paremti algoritmai, skaičiuodami kurį veiksmą aktoriui šiuo metu geriau daryti, remiasi tik dabartine nustatyta taisykle, o algoritmai paremti ne pagal taisykles principu tą patį žingsnį skaičiuoja pagal veiksmą a^* , kuris yra gautas iš kitos, paprastai prieš tai buvusios, taisyklės. Pvz., tradiciniame Q naudos funkcijos algoritme yra taikoma godi (angl. *greedy*) taisyklė, kai yra tikimasi, kad agentas pastoviai rinksis taisyklę, kuri jį greičiausiai ves arčiau tinkamiausio rezultato (Baird & Moore, 1999). Tai iš esmės reiškia, kad algoritmai, kurie yra paremti taisyklėmis, pasieks teisingą rezultatą atlikus mokymą ir tuomet toliau bus naudojamas tas pats taisyklių komplektas. Tokio tipo algoritmai paprastai greičiau apsimoko bei daugiau dėmesio skiria į atliktas klaidas, tad yra pranašesni tyrimuose, kuriuose apmokymo žingsnis yra brangus arba yra svarbu išvengti kuo daugiau klaidų apmokymo etape. Pvz., apmokant robotus, nes klaidos gali būti brangios. Kita vertus ne pagal taisykles tipo algoritmai yra pranašesni tais atvejais, kai apmokymo periodo efektyvumas mums nėra toks svarbus. O po kurio laiko mums reikės pilnai pasitikėti apmokymo metu gautais rezultatais ir atlikti kuo mažiau atsitiktinių žingsnių (Degris, White, & Sutton, 2013), pavyzdžiui, situacijose, kai yra žaidžiami keli apmokomieji treniruočių žaidimai, kuriuos galime pralaimėti nieko nepraradę, tačiau po apmokymo bus žaidžiamas turnyras, kurio svarba lyginant su treniruotėmis yra ypač svarbi.

Norint toliau tikslingai analizuoti mašininio mokymosi algoritmus yra būtina aptarti galimus funkcijų pasirinkimus.

Q naudos funkcija – Q naudos funkcijos mokymosi algoritmai yra ne pagal taisykles (angl. *Off-Policy*) tipo algoritmai. Pagrindinė Q naudos funkcijos idėja yra atlikti daug iteracijų bei pagal gautus rezultatus keisti Q lentelės vertes, kuriomis vadovaujantis vėliau galima atlikti tam tikras užduotis. Šių algoritmų tikslas yra kiek įmanoma maksimizuoti Q naudos funkcijos vertę ir tomis vertėmis pildyti Q lentelę. Q naudos funkcija yra pateikiama žemiau (žr. 1 formulė).

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right] \quad (1)$$

α šioje funkcijoje nurodo, kaip greitai yra artėjama prie tikslo arba mokymosi spartą, o γ simbolizuoja koeficientą, kuriuo nustatoma, kaip stipriai algoritmas turėtų atsižvelgti į galimą prizą ateityje.

State-Action-Reward-State-Action (toliau – SARSA) funkcija – SARSA funkcijos algoritmas yra labai panašus į Q naudos funkcijos algoritmą. Pagrindinis skirtumas tarp jų yra tai, kad SARSA yra algoritmas paremtas taisyklėmis (angl. *on-policy*). O tai reiškia, kad algoritmo sprendimai yra atliekami pagal tam tikras iš anksto išmoktas taisykles (angl. *policies*). SARSA algoritmo funkcija pateikiama žemiau (žr. 2 formulė)

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (2)$$

Šioje funkcijoje matoma, kad priešingai nei įprastoje Q naudos funkcijoje veiksmo pasirinkimas iš Q lentelės yra atliekamas 2 kartus, tad tikėtina, kad naudojantis šia funkcija itin sudėtingus skaičiavimus atlikti galima kiek tiksliau, tačiau visi skaičiavimai naudos daugiau kompiuterio išteklių. Todėl SARSA gali būti perteklinė būtent dėl šios priežasties.

Gilieji Q tinklai (angl. Deep Q Network) (toliau – DQN) – dviejų prieš tai aptartų algoritmų bendras trūkumas yra tai, kad jie abu yra paremti dinaminio programavimu ir Q lentelė yra atnaujinama tik sutikus tam tikrą situaciją bei atlikus atitinkamus skaičiavimus. Tai reiškia, kad jei agentas kažkada atsidurtų būsenoje, kurioje jis dar nėra buvęs, jis elementariai nežinotų, kurį veiksmą atlikti yra geriau. Norint geriau spręsti šią problemą, buvo sugalvota atsisakyti dvimatės Q lentelės matricos ir viskam pritaikyti neuroninius tinklus. Būtent DQN naudoja neuroninius tinklus bei jų apmokymą, norint suteikti agentui daugiau informacijos jam dar nebūtose būsenose. Kita vertus šio darbo tikslinė auditorija yra itin jautri, o kitų autorių tyrimų tiriančių šio darbo problemą nebuvo rasta. Norint užtikrinti darbo tikslumą visose tyrimo etapuose buvo pasirinkta DQN šiame darbe nenaudoti.

Palyginus mašininio mokymosi algoritmus nustatyta, kad darbe išsikeltam tyrimo tikslui pasiekti tinkamiausias yra Q naudos funkcijos tipo algoritmas, nes ypač svarbu, kad jau apmokyta sistema teisingai atliktų sprendimus, vengdama įvairių variacijų bei neapibrėžtumo.

2.3. Mašininio mokymosi taikymo ASS turintiems naudotojams mokslinių tyrimų analizė

Tyrimai naudojantys skatinamąjį mokymąsi yra atliekami jau daug metų. Būtent šie algoritmai padeda išspręsti uždavinius, kurie gali būti sprendžiami agentui veikiant dinaminėse, nuolat kintančiose aplinkose, todėl ir tyrimai susiję su šio tipo algoritmais paprastai yra vykdomi situacijose, kuriose po testavimo bei apmokymo realiai egzistuojanti situacija nėra pilnai aiški ar gali kisti (Qiang & Zhongli, 2011). Paprastai tyrimai naudojantis skatinamąjį mokymąsi bando spręsti informacijos filtravimo, kelio radimo ar kitus uždavinius, kai yra žinomas palyginus apibrėžtas norimas rezultatas, tačiau nėra tiksliai žinoma kaip tas rezultatas bus pasiekiamas. Šie algoritmai ypač tinka taikyti šio tyrimo tikslinei auditorijai, nes dėl savo savybių nėra tiksliai aišku kokiam vaikui kokie dalykai gali patikti (Matson, Mahan, Hess, Fodstad, & Neal, 2010). Su šio darbo tema susijusių autorių darbų buvo galima rasti 6. Šių darbų santrauka pateikta lentelėje žemiau (žr. 3 lentelė).

3 lentelė. Susijusių autorių darbų analizė (sudaryta autoriaus)

Autoriai	Tyrimo objektas	Rezultatas	Taikyti metodai
A. H. Khabbaz, A. A. Pouyan, M. Fateh, V. Abolghasemi (Khabbaz, Pouyan, Fateh, & Abolghasemi, 2019)	Rimtas žaidimas (angl. <i>Serious game</i>), skirtas įvertinti vaikų, turinčių ASS, socialinius gebėjimus.	Darbo eksperimentai rodo, kad rimti žaidimai, kurių lygiai yra paremti skatinamojo mokymosi SARSA algoritmu gali būti naudingi asmenims, turintiems ASS, ypač kai žaidimas žaidžiamas ilgesnį laiką.	Skatinamojo mokymosi SARSA algoritmu buvo kurta eksperimento logika.
Minjia Li, Xue Li, Lun Xie, Jing Liu, Feifei Wang & Zhiliang Wang (Li, et al., 2019)	Pagalbinė terapinė sistema, skirta ASS turintiems asmenims.	Tyrimo metu kurtos pagalbinės terapijos skatinamuoju mokymusi pagrįstas prognozavimo modelis pasiekė teigiamus rezultatus. Šie rezultatai parodo prognozavimo modelio, pagrįsto skatinamuoju mokymusi, veiksmingumą ir parodo pagalbinės terapinės sistemos veiksmingumą vaikams turintiems ASS.	Skatinamasis mokymasis paremtas neuroniniais tinklais bei vektorių regresija.
Wei Li, Xiaoli Qiu, Yang Li, Jing Ji, Xinxin Liu, Shuanzhu Li (Li, et al., 2021)	Mašininiu mokymusi paremta alternatyvaus bendravimo sistema.	Rezultatai rodo, kad tyrimo metu kurta mašininiu mokymusi paremta sistema buvo daug suprantamesnė ASS turintiems tyrimo dalyviams, lyginant su tradicinėmis alternatyviojo bendravimo sistemomis.	DQN paremtas skatinamojo mokymosi algoritmas

Autoriai	Tyrimo objektas	Rezultatas	Taikyti metodai
Ognjen (Oggi) Rudovic , Meiru Zhang, Björn Schuller, Rosalind W. Picard (Rudovic, Zhang, Schuller, & Picard, 2019)	Daugiamodalinis žmogaus elgesio duomenų apdorojimas.	Tyrimo metu parodytas daugiamodalinės Q naudos funkcijos efektyvumas. Taip pat buvo ištirta, kad naudojant daugiamodalinę modelio apmokymo logiką paspartėja pačio modelio veikimas bei efektyvumas.	Daugiamodalinės Q naudos funkcijos mokymasis.
Torki Altameem, Mohammed Amoon, Ayman Altameem (Altameem, Amoon, & Altameem, 2020)	Dirbtiniu intelektu paremtų robotų, skirtų bendravimui su protinę negalia turinčiais asmenimis, apmokymas.	Pasiūlytas skatinamuoju mokymusi paremtas modelis eksperimentų metu pateisino savo veiksmingumą ir parodė 98.42 % tikslumą savo spėjimuose.	DQN paremtas algoritmas.
Hui Huang, Xiaojun Hei, Yayu Gao, Chengwei Zhang (Huang, Hei, Gao, & Zhang, 2020)	Mašininio mokymusi paremtas „WeChat“ sistemai kurtas mokymosi įrankis ASS turintiems vaikams.	Dėl tinkamai taikytų mašininio mokymosi algoritmų aspektų buvo geriau išlaikytas ASS turinčių vaikų dėmesys. Taip pat pati sistema buvo efektyvesnė, nes galėjo siūlyti kiekvienam naudotojui personalizuotas užduotis.	Mašininio mokymosi algoritmai: Logistic Regression, Random Forest.

Išanalizavus su šio darbo tema susijusių autorių darbus buvo nustatyta, jog norint užtikrinti teigiamą tikslinės auditorijos naudotojų patirtį yra svarbu, kad skaitmenizuota PECS sistema būtų optimizuota ir kiek įmanoma efektyvesnė. O realizacijos paremtos skatinamuoju mokymusi yra tinkama priemonė šiam tikslui pasiekti.

2.4. Skyriaus išvados

Analizės metu taip pat nustatyta, jog norint užtikrinti teigiamą sistemos panaudojamumą yra svarbu, kad algoritmas veiktų gerai su ypač mažu pradiniu duomenų kiekiu, tačiau vis tiek sėkmingai bei tinkamai prisitaikytų prie kiekvieno sistemos naudotojo. Tyrimas yra atliekamas specifinė auditorijoje – ASS turinčių vaikų naudojamose PECS sistemose, tad pradinių duomenų apie naudotojų elgesį yra ypač mažai.

Išanalizavus dirbtinio intelekto mašininio mokymosi algoritmus nustatyta, kad tyrimo tikslinei auditorijai skirtus algoritmus geriausia įgyvendinti naudojant skatinamojo mokymosi tipo mašininį mokymąsi.

Išanalizavus ir palyginus mašininio mokymosi algoritmus nustatyta, kad išsikeltam tikslui pasiekti tinkamiausias yra Q naudos funkcijos tipo algoritmas, nes ypač svarbu, kad jau apmokyta sistema teisingai atliktų sprendimus, vengdama įvairių variacijų bei neapibrėžtumų.

3. Naudotojo sąsajos projektavimo metodo kūrimas ir tyrimas

Skyriuje pateikiamas darbe siūlomas projektavimo metodas, skirtas pagerinti skaitmenizuotų PECS sistemų panaudojamumą. Skyriuje pateikiami 2 esminiai darbe atliekami etapai: tyrimai su kitų autorių kurtomis aplinkomis bei pačio darbo autoriaus siūlomos aplinkos kūrimas bei testavimas. Visų pirma iškeliami reikalavimai, kuriuos siūlomas metodas turėtų įvykdyti ir pateikiamos viso tyrimo bei dalinių tyrimo etapų verslo procesų modeliavimo ir notacijos (angl. *Business Process Model and Notation*) (toliau – BPMN) diagramos bei jų apibūdinimai.

3.1. Projektavimo metodo kontekstas ir reikalavimai

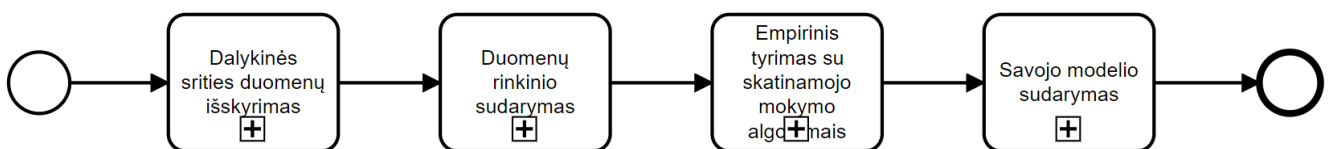
Analitinėje dalyje buvo išanalizuotas ir siūlomo metodo kontekstas, kuris yra ypač svarbus tyrimo tikslinei auditorijai. Dėl tikslinės auditorijos naudotojų specifinių savybių yra svarbu, kad algoritmas veiktų tinkamai su ypač mažu pradiniu duomenų kiekiu, tačiau vis tiek sėkmingai bei tinkamai prisitaikytų prie kiekvieno sistemos naudotojo.

Apibendrinant analitinę dalį siūlomam projektavimo metodui galima išsikelti šiuos reikalavimus:

- PECS sistema gali būti sudaryta iš skirtingo kortelių kiekio.
- Dėl tyrimo tikslinės auditorijos specifinių savybių bei algoritmų specifiškumo reikia sąryšio tarp visų PECS kortelių.
- Dėl tikslinės auditorijos specifikos tyrimui atlikti neįmanoma turėti iš anksto sužymėto duomenų rinkinio.
- Dėl naudotojų individualumo bei tikslinės auditorijos specifikos nėra įmanoma sukaupti patirtinių sužymėtų duomenų.
- Vaiko raidos centro specialistų teigimu komunikacijai yra aktualios trys reakcijos į korteles

3.2. Projektavimo metodo etapai

Siekiant pasiūlyti naudotojo sąsajos projektavimo metodą nustatyta, kad tikslui pasiekti reikėjo atlikti 4 etapus. Kiekvieno jų rezultatas yra panaudojamas kitai veiklai kaip įėjties duomenys. Etapai pateikiami pasitelkiant BPMN diagramos notaciją (žr. 8 pav.).



8 pav. Projektavimo etapai (sudaryta autoriaus)

Visų 8 paveikslėlyje pavaizduotų etapų aprašymai pateikiami žemiau:

- Dalykinės srities duomenų išskyrimas – šios veiklos metu buvo nustatomi specialūs bei išskirtiniai tyrimo tikslinės auditorijos aspektai bei kuriama sistemos specifika.
- Duomenų rinkinio sudarymas – šios veiklos metu yra sudaromas tyrimo metu naudojamas duomenų rinkinys, nustatomas kortelių ir jų tarpusavio ryšys.
- Empirinis tyrimas su skatinamojo mokymosi algoritmais – šios veiklos metu aprašomi veiksmai, tyrimai bei kita veikla atlikta su skatinamojo mokymosi algoritmais bei atliekama jų analizė.
- Savojo modelio sudarymas – šioje veikloje yra aprašomas tyrimo metu pasiūloma aplinka, jos savybės, įvairūs parametrai bei kiti algoritmui svarbūs aspektai.

3.2.1. Dalykinės srities duomenų analizė

Šio etapo metu buvo nustatomi visi tikslinės auditorijos specifikos aspektai, analizuotos PECS sistemos. Dalykinės srities duomenų analizės metu gauta informacija, išvados detaliai aprašomos šio darbo 1 skyriuje. Apibendrinant paminėtina, kad programėlės turi būti kuriamos pagal tam tikrus gerus panaudojamumo standartus bei praktikas, kadangi naudotojai gali programėlių nenaudoti dėl nepakankamo jų funkcionalumo, nepatogumo. Todėl buvo ypač svarbu išsiaiškinti problemas su kuriomis susiduria PECS sistemų naudotojai naudodamiesi skaitmenizuotomis PECS sistemomis ir atlikti ASS turinčio asmens patirties specifikos analizę.

Pastebėta, kad egzistuojančiose skaitmenizuotose PECS sistemose naudotojams yra pateikiamos sistemų kūrėjų sukurtos PECS kortelės ar jų rinkiniai, kurie nėra rūšiuojami pagal naudotojo kortelių naudojimą ar kitus aspektus, o tai yra labai neefektyvu.

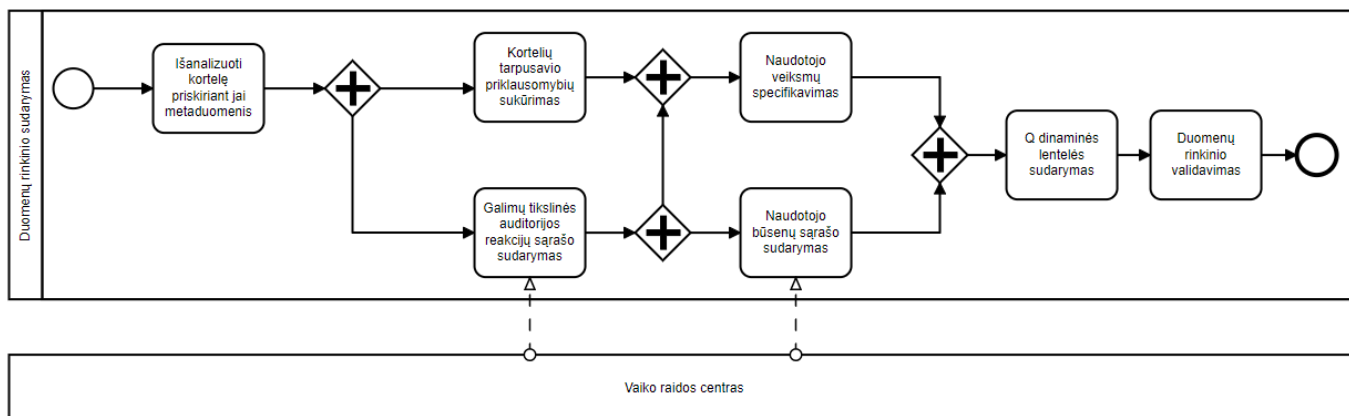
Išanalizavus atliktus mokslinius tyrimus nustatyta, kad atliktuose tyrimuose naudojamų kortelių įvairovė yra didelė ir nebuvo pastebėta jokių kortelių ar jų kategorijų, kurios pasikartotų visuose tyrimuose. Tyrimuose buvo remiamasi specialistų rekomendacijomis ir visi jie gavo teigiamus rezultatus, tad tikėtina, kad ir tyrimo duomenų rinkiniai, PECS kortelės, buvo teisingai surinktos. Norint patvirtinti pastebėtus rezultatus buvo išanalizuotos kortelės, kurios yra naudojamos kitų autorių kurtose PECS sistemos realizacijose. Apžvalgai buvo pasirinktos 2 svetainės, kuriose platinamos PECS standartus atitinkančios kortelės, bei 2 šiuo metu populiarios programėlės, platinamos produktų programėlių platinimo sistemoje.

Atsižvelgus į kituose moksliniuose tyrimuose taikomas korteles, taip pat išanalizavus korteles naudojamas egzistuojančiose sistemose, nebuvo atrasti jokie pastebimi kortelių, kurios yra naudojamos skirtinguose tirtuose objektuose, panašumai. Kiekvienoje sistemoje buvo naudojamas vis kitas kiekis kortelių.

Konsultuojantis su Vaiko raidos centro specialistais buvo nuspręsta, kad tyrimo metu yra optimalu taikyti 81 kortelę, suskirstytą į 9 skirtingas kategorijas. Tačiau atsižvelgiant į tai, kad kiekvienas vaikas yra unikalus bei egzistuoja PECS sistemos realizacijų, kuriose patys naudotojai gali kurti korteles, mašininio mokymu paremtos sistemos turi gerai veikti ir su gerokai didesniu ar mažesniu kiekiu kortelių.

3.2.2. Duomenų rinkinio sudarymas

Šio etapo metu buvo sudaromas duomenų rinkinys naudojamas tyrime. Duomenų surinkimui pasitelkta Vaiko raidos centro specialistų pagalba, kurie galėjo pasiūlyti informacijos apie galimas tikslinės auditorijos reakcijas bei kitus algoritmui svarbius aspektus. Struktūrizuojant šį etapą detaliau galima būtų suskirstyti jį į 7 mažesnius etapus. Žemiau pateikiama detalesnė šių etapų diagrama (žr. 9 pav.).



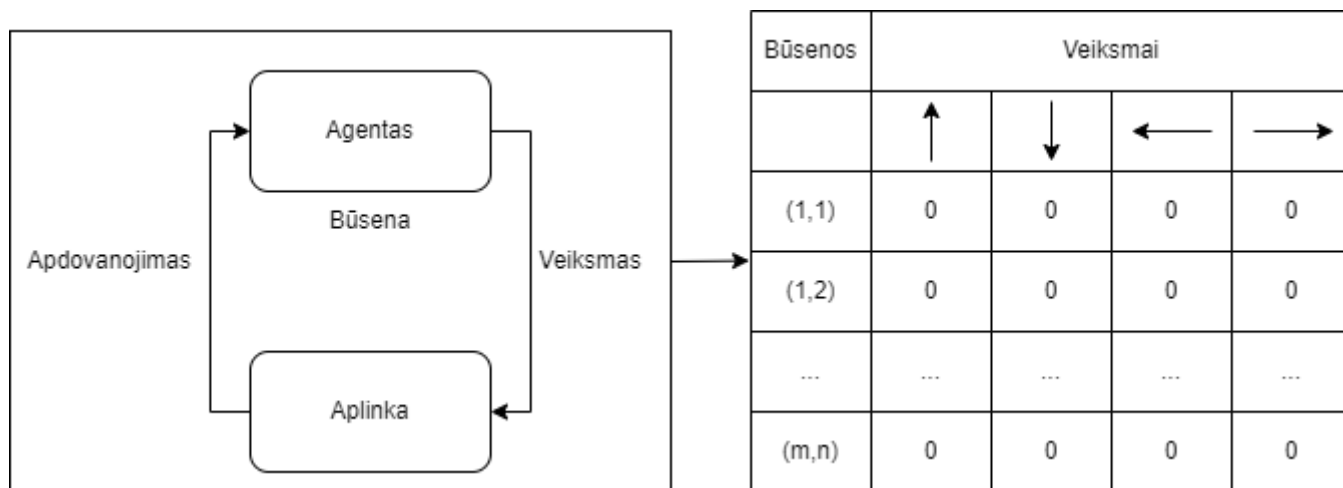
9 pav. Duomenų rinkinio sudarymas (sudaryta autoriaus)

Nagrinėjamo duomenų rinkinio sudarymo etapo detalizavimas

- Išanalizuoti korteles priskiriant joms metaduomenis – šios veiklos metu buvo analizuojamos kortelės bei jų savybės bei galimas jų tarpusavio ryšių kūrimas.
- Kortelių tarpusavio priklausomybių sukūrimas – šios veiklos metu buvo kuriami ryšiai tarp kortelių bei jos siejamos tarpusavyje, norint užtikrinti kokybišką agento perėjimą tarp jų. Tai užtikrina galimybę agentui be trikdžių judėti apsibrėžtoje aplinkoje, t. y. suteikia agentui galimybę rodyti korteles ir naviguoti tarp jų.
- Galimų tikslinės auditorijos reakcijų sąrašo sudarymas – šios veiklos metu buvo konsultuojamasi su Vaiko raidos centro specialistais siekiant išsiaiškinti, kokias apibrėžiamas reakcijas gali jausti ASS turintis vaikas. Tikslas yra sudaryti baigtinį jų sąrašą.
- Naudotojo veiksmų specifikavimas – šios veiklos metu buvo specifikuojama, kokius veiksmus gali sistemoje atlikti naudotojas veikdamas sistemos ribose.

- Naudotojo būsenų sąrašo sudarymas – šios veiklos metu galimos naudotojų reakcijos buvo performuotos į galimas naudotojų būsenas norint geriau jas identifikuoti bei vėliau aprašyti skatinamojo mokymosi algoritmu.
- Q dinaminės lentelės sudarymas – šios veiklos metu buvo naudojama visa iki šiol sutelkta informacija bei ji buvo konvertuota į skatinamiesiems mokymo algoritmams suprantamą formatą – Q lentelę.
- Duomenų rinkinio validavimas – šios veiklos metu sudaryta Q lentelė bei jos struktūra buvo validuojama ir vertinama ar ją bus galima naudoti tyrimams. Sudaryta Q lentelė bei jos struktūra buvo pradžioje taikyta kitų autorių kurtoms aplinkoms apmokyti bei žiūrėta ar gaunami rezultatai yra teigiami.

Atliktos literatūros analizės metu nebuvo rasta nei vieno mokslinio straipsnio, kuris analizuotų, kurios PECS kortelės yra svarbesnės už kitas. Tačiau pavyko rasti keletą mokslinių tyrimų (Malandraki & Okalidou, 2007) bei (Soomro & Soomro, 2018), kuriuose buvo naudojamos tam tikros kortelių kategorijos, nors tyrimo esmė ir nebuvo ištirti kurios kategorijos yra svarbiausios. Tas pats buvo pastebėta ir analizuojant jau egzistuojančias programėles. Tad visos tirtos egzistuojančios PECS komunikacijos programėlės bei kitų autorių atlikti moksliniai tyrimai naudoja kategorijas, kurios buvo sukurtos remiantis specialistų rekomendacijomis bei pagalba, galutinių sistemų naudotojų nuomone arba pačių programos kūrėjų nuožiūra. Šių kortelių įvairovė bei skirtumai rodo, kad tyrimo metu yra svarbu testuoti algoritmus su kintančiais kiekiais kortelių, tačiau pasitarus su Vaiko raidos centro specialistais nustatyta, kad didžioji dalis testavimo turėtų būti atliekama su 81 kortele suskirstyta į 9 kategorijas.



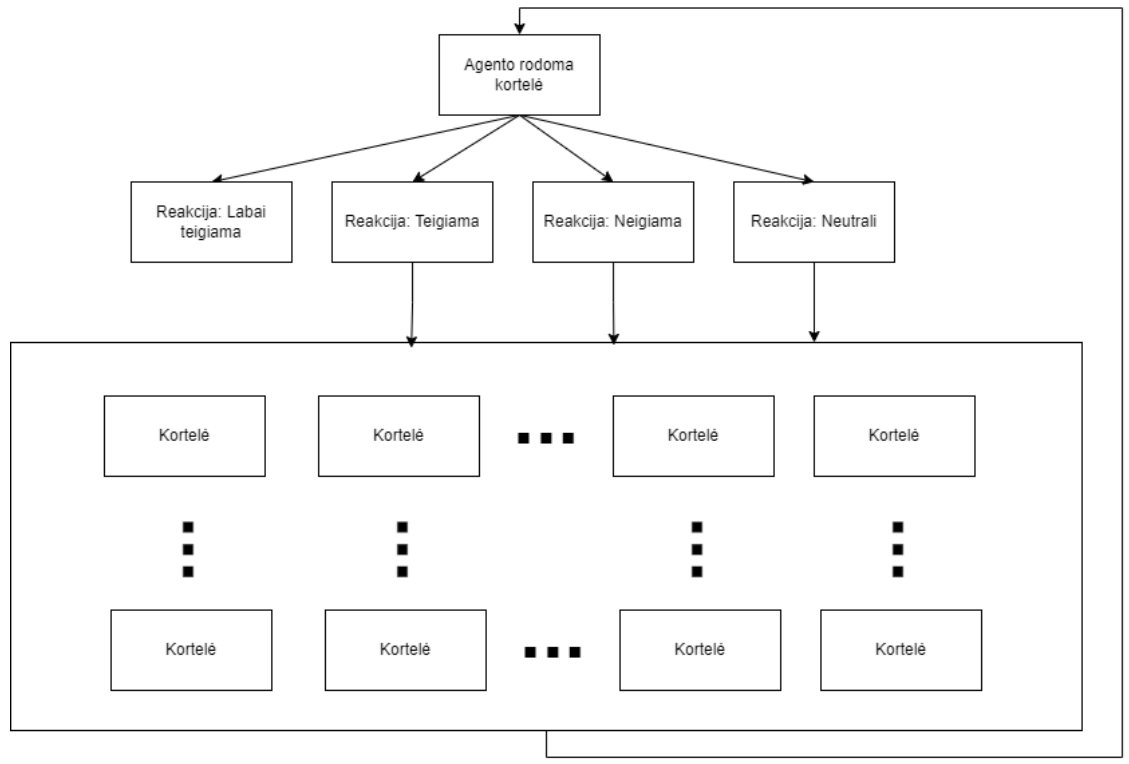
10 pav. Agento sąveikos bei Q lentelės pavyzdys (sudaryta autoriaus remiantis (Zhao, Lu, Yang, & Guo, 2020))

Norint tinkamai taikyti Q naudos funkcija paremtą skatinamąjį mokymąsi yra būtina tinkamai apibrėžti Q lentelės turinį bei galimas agento sąveikas su lentele. Q lentelės ašis sudaro būsenos, kurias gali

pasiekti agento mašina bei galimi agento veiksmai, kuriais jis keičia minėtas būsenas. Tuomet visi lentelės langeliai yra užpildomi naudojantis Q naudos funkcija agentui sąveikaujant aplinkoje bei gaunant apdovanojimus ar baudas (žr. 10 pav.).

3.2.2.1. *Q dinaminės lentelės būsenų išskyrimo aprašymas*

Norint tiksliai taikyti skatinamąjį mokymąsi tyrimuose yra svarbu apibrėžti aiškias būsenas į kurias gali patekti dirbantis algoritmas.



11 pav. Tyrimo metu simuliuojama situacija (sudaryta autoriaus)

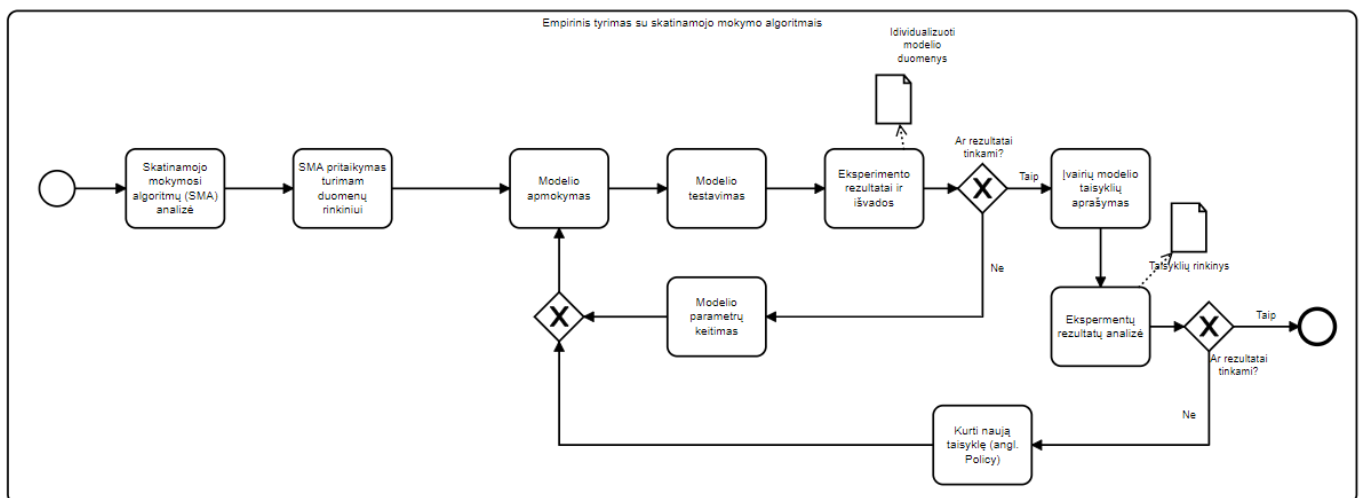
Tyrimo metu yra bandoma simuliuoti situaciją, kai kiekviena kortelė gali būti parodoma ekrane ir į ją atitinkamai naudotojas gali reaguoti tam tikromis neigiamomis arba teigiamomis reakcijomis. Tuomet yra rodoma kita kortelė iki kol randama ta kortelė kuri labai patinka naudotojui (žr. 11 pav.). Tad visų atliekamų tyrimų metu kiekviena modelio agento pasiekiamą būseną atitinka pateiktos kortelės rodyimą, o kiekviena sistemos aplinkoje apibrėžta kortelė yra viena galima agento mašinos būseną.

Taikant skatinamąjį mokymąsi yra svarbu aiškiai apibrėžti agento baudas bei apdovanojimus, kurie yra nustatomi pagal tam tikrus iš anksto aplinkoje numatytus kriterijus. Agentas tyrimo aplinkoje turi būti apdovanojamas bei baudžiamas už tam tikrą pasiektą būseną, kuri tyrimo aplinkoje yra imituojama tam tikra galima reakcija į kortelę. Norint kuo tiksliau simuliuoti galimas naudotojo reakcijas į rodomas korteles buvo konsultuojamasi su Vaiko raidos centro specialistais, kurie išsakė, kad atitinkamai į tam tikrą kortelę ASS turintis vaikas ženklais ar kitais veiksmais gali reaguoti Neigiamai, Teigiamai bei itin retais atvejais

Labai teigiamai. Taip pat tikslinės auditorijos analizės metu nustatyta, jog algoritmas turi prisitaikyti prie kiekvieno naudotojo individualiai, o skirtingiems naudotojams gali patikti visiškai skirtingi dalykai, tad simuliacijoje yra tikslinga galimas reakcijas į tam tikras korteles realizuoti tam tikromis tikimybėmis.

3.3. Tyrimo aprašymas su skatinamojo mokymosi algoritmais

Šiame skyriuje aprašomi etapai su skatinamojo mokymo algoritmais bei atliekama jų analizė. Ši etapą būtų galima išskirstyti į 10 mažesnių veiksmų. Detalesnė diagrama yra pateikiama žemiau (žr. 12 pav.).



12 pav. Empirinio tyrimo su skatinamojo mokymosi algoritmais etapo BPMN diagrama (sudaryta autoriaus)

Tyrimo darbo etapas vykdytas 2 etapais:

1. Apibrėžtų skatinamojo mokymo aplinkų taikymas varijuojant parametrais ir taisyklėmis – etapas, kurio metu buvo analizuojami algoritmai bei aiškinamasi jų teigiamos bei neigiamos savybės.
2. Individualizuoto modelio ASS vaikams sudarymas – šio etapo metu iš informacijos gautos iš pirmojo etapo buvo formuojamas šio darbo pasiūlomasis modelis bei jo struktūra.

Specifikuojant šio etapo diagramą smulkiau galima apibrėžti 10 skirtingų mažesnių etapų:

- Skatinamojo mokymosi algoritmų analizė sudarytam duomenų rinkiniui – šios veiklos metu buvo nustatoma ar jau egzistuoja algoritmų, kurie būtų tinkami teisingai apdoroti norimą duomenų rinkinį.
- Skatinamojo mokymosi algoritmų paleidimas – šios veiklos metu buvo paleidžiama bei atitinkamai konfigūruojama sistema, norint kuo tiksliau ištirti algoritmų veikimą.

- Skatinamojo mokymosi algoritmų pritaikymas turimam duomenų rinkiniui – šios veiklos metu algoritmas buvo atitinkamai pritaikytas, kad juo naudojantis būtų galima analizuoti rezultatus gautus naudojant tyrimo duomenų rinkinį.
- Modelio apmokymas – šios veiklos metu naudojant algoritmų funkcijas modeliai buvo apmokyti ir paruošti tolimesniems tyrimo testams.
- Modelio testavimas – šios veiklos metu su apmokytu modeliu buvo atliekami empiriniai testai.
- Eksperimento rezultatai bei išvados – šios veiklos metu buvo apibrėžti gauti testavimo bei apmokymo rezultatai ir iš jų išvedamos atitinkamos išvados.
- Modelio parametrų keitimas – šios veiklos metu buvo keičiami modelio parametrai (pvz., mokymosi sparta, atsitiktinio ėjimo tikimybė ir t. t.).
- Įvairių modelio taisyklių aprašymas – šio proceso metu buvo aprašomos identifikuotos modelio taisyklės.
- Tyrimo rezultatų analizė – šios veiklos metu buvo apibrėžiami gauti rezultatai bei analizuojama, ar gauti rezultatai tenkina tyrimą.
- Naujos modelio taisyklės kūrimas – šio proceso metu buvo keičiamos esminės modelio taisyklės kitiems testavimo bei apmokymo etapams.

3.3.1. Tyrimų aplinkos parametrų aprašymas

Norint tikslingai atlikti tyrimus ir gauti prasmingus rezultatus yra svarbu tyrimams pasirinkti tinkamą aplinką, o jos ieškant yra svarbu atsižvelgti į galimas taikyti programavimo kalbas. Anot tyrimo atlikto 2019 metais, kuris lygina programavimo kalbas, norint atlikti mašininio mokymosi užduotis, Python programavimo kalba yra viena pranašiausių kalbų vykdant veiksmus susijusius su mašininio mokymusi. Anot autorių I. Stančin ir A. Jović, Python programavimo kalba turi daug bibliotekų, skirtų būtent duomenų analizei bei darbui su duomenimis (Stančin & Jović, 2019). Taip pat Python programavimo kalba siūlo bibliotekas, kuriomis galima gana lengvai braižyti grafikus bei vizualiai atvaizduojamą informaciją (Nagpal & Gabrani, 2019). Būtent dėl šių priežasčių darbe atliekamiems tyrimams bus naudojama python programavimo kalba, o siekiant lengvesnio duomenų vizualizavimo diagramomis ir paprasto diagramų eksportavimo iš aplinkos buvo pasirinkta naudoti „Jupyter lab“ programavimo aplinką.

Norint išlaikyti tyrimų vientisumą bei dėl analitinėje dalyje įvardintų argumentų, visiems testuojamiems modeliams buvo naudojama Q naudos funkcija:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right] \quad (3)$$

Šioje funkcijoje s atspindi aplinkos algoritmo būseną, kurią gali pasiekti agentas, o a atspindi agento veiksmą, kuris apibūdina judėjimą tarp šių būsenų. Tuomet atitinkamai modelyje bus keičiami Q naudos funkcijos bei aplinkos parametrai (žr. 4 lentelė). Visi kiti Q naudos funkcijos akronimai yra aprašyti 2.2.2 Skatinamojo mokymosi (angl. *Reinforcement learning*) algoritmų analizė poskyryje.

4 lentelė. Konfigūruojami Q naudos funkcijos bei aplinkos parametrai bei jų reikšmės (sudaryta autoriaus)

Parametras	Atitikmuo kode	Reikšmė	Ribos
Apsimokymo dažnis (atitikmuo formulėje – α) (toliau – learningRate)	learning_rate	Šis parametras leidžia nustatyti, kaip greitai algoritmas mokosi. T. y., kaip greitai algoritmas bandys atnaujinti jau įrašytus Q lentelės duomenis naujais, paskaičiuotais šio epizodo metu. kuo ši reikšmė didesnė, tuo didesnė ir algoritmo priimamų naujų žinių sparta	dažniausiai nustatomos [0;1]
Nuolaidos dydis (atitikmuo formulėje – γ) (toliau – gamma)	gamma	Šis parametras leidžia nustatyti, kaip stipriai besimokindamas algoritmas turėtų atsižvelgti į galimus agento ateities prizus. T. y. nustačius aukštą šio parametro vertę, algoritmas turėtų labiau stengtis išmokti atlikti veiksmus, kurie leistų agentui ateities žingsniuose gauti didesnę prizą. Toks agentas būtų mažiau jautrus į gaunamą baudą ar prizą, bet daugiau koncentruotųsi į galimą prizą ateityje.	dažniausiai nustatomos [0;1]

Parametras	Atitikmuo kode	Reikšmė	Ribos
Tyrinėjimo tikimybė (toliau – epsilon)	epsilon	Šis parametras naudojamas ne Q naudos funkcijoje, o pačioje agento veiksmų pasirinkimų logikoje apsimokymo metu. Apmokant skatinamojo algoritmo agentus yra norima, kad agentas aplinkos gyvenimo ciklo pradžioje bandytų kuo daugiau atlikti atsitiktinių veiksmų norint surinkti kuo daugiau informacijos, o pabaigoje, priešingai, atsitiktinių veiksmų beveik arba visiškai neatlikinėtų. Taip norima užtikrinti, kad agentas tikslingai mokytųsi bei įsisavintų informaciją. Todėl kuo daugiau agentas yra ištyrinėjęs, tuo jis daugiau žino bei tuo mažiau reikia tokio tyrinėjimo.	[0;1]
Maksimali tyrinėjimo tikimybės reikšmė (toliau – maxEpsilon)	max_epsilon	Maksimali tyrinėjimo tikimybės reikšmė, nuo kurios yra pradedamas tyrimas.	[0;1]
Minimali tyrinėjimo tikimybės reikšmė (toliau – minEpsilon)	min_epsilon	Minimali tyrinėjimo tikimybės reikšmė, kurią pasiekus tyrinėjimo tikimybė parametras nebėra mažinamas toliau.	[0;1]
Tyrinėjimo tikimybės mažinimo dydis (toliau – decayRate)	decay_rate	Duomuo nustatantis kaip greitai yra mažinama tyrinėjimo tikimybės reikšmė.	[0;1]
Maksimalus iteracijų kiekis	total_episodes	Šiuo parametru yra nustatoma kiek mokymosi iteracijų algoritmas bus mokinamas. Kiekvieno epizodo metu aplinka, bet ne pati Q lentelė yra nustatoma iš naujo bei agentas joje atlieka veiksmus. Tuomet pasiekus epizodo pabaigą tikrinama ar epizodų skaičius dar nepasiekė total_episodes vertės ir jei ji nepasiekta yra inicijuojamas kitas epizodas.	[1;∞)

Parametras	Atitikmuo kode	Reikšmė	Ribos
Maksimalus žingsnių per vieną iteraciją kiekis (toliau – maxSteps)	max_steps	Maksimalus žingsnių kiekis kurį gali padaryti agentas kiekviename epizode. Jei šis žingsnių kiekis yra pasiekiamas tuomet skaitoma, kad agentas atlikti užduoties nesugebėjo.	$[1;\infty)$

Epsilon parametrui yra nustatoma pradinė vertė, kuri yra nuolat mažinama nustatyta decayRate sparta, kol pasiekiamas nustatyta minimali vertė. Šis mažinimas yra dažniausiai atliekamas eksponentiškai, norint kuo greičiau pereiti prie tikslingo agento mokymosi iš jau bent minimaliai užpildytos Q lentelės. Epsilon perskaičiavimo formulė yra pateikiama paveikslėlyje žemiau (žr. 13 pav.).

```
epsilon = min_epsilon + (max_epsilon - min_epsilon)*np.exp(-decay_rate*episode)
```

13 pav. Epsilon parametro keitimo logika kode (autorius rašyto kodo fragmentas)

Pilnai apmokius modelį tam tikrą kiekį epizodų, nuspręsta leisti atsitiktinių 100 testų, kuriuose agentas remiasi tik išmokta informacija bei neatlieka jokių paieškos veiksmų. Testuose buvo naudojama žemiau pateikta funkcija (žr. 14 pav.).

```
action = np.argmax(qtable[state,:])
```

14 pav. Agento veiksmo pasirinkimas kai remiamasi tik išmoktais Q lentelės duomenimis kode (autorius rašyto kodo fragmentas)

Ši funkcija leidžia užtikrinti, kad agentas testų metu rinkdamasis iš galimų veiksmų remtųsi tiktais išmoktais duomenimis. T. y. agento kodas nuolat parenka veiksmą, kuris teiks daugiausiai apdovanojimo kiekviename žingsnyje. Tuomet buvo analizuojama, kiek testų iš 100 buvo teisingai baigiami bei kaip greitai tai įvyko.

3.3.2. Skatinamojo mokymosi aplinkų analizė

Tyrimams atlikti reikėjo visų pirma išanalizuoti jau egzistuojančias, kitų autorių sukurtas skatinamojo mokymosi aplinkas ir pasirinkti jų tinkamumą sprendžiamam darbe uždaviniui. Šiam tikslui buvo analizuojamos vienos populiariausių skatinamojo mokymosi platformų kūrėjo dirbtinio intelekto

tyrimų ir diegimo įmonės „OpenAI“ (toliau – OpenAI) kurtos aplinkos. Visos jos yra skirtos specifiniams uždaviniams spręsti naudojant skatinamąjį mokymąsi. Tad buvo būtina įvertinti kiekvienos jų tinkamumą tyrimui. Visos OpenAI aplinkos yra suskirstytos į 7 kategorijas (žr. 5 lentelė).

5 lentelė. Įvairių OpenAI siūlomų aplinkų kategorijos (sudaryta autoriaus)

Kategorija	Apibūdinimas
„Algorithms“	Šioje kategorijoje yra sudėtos aplinkos, leidžiančios agentui atlikti įvairius veiksmus su simbolių eilute ir imituoti kompiuterinius skaičiavimus. Agentas gali atlikti tokius veiksmus kaip: sudėti, atimti, kopijavimą bei kitus paprastesnius veiksmus.
„Atari“	Šioje kategorijoje sudėtos aplinkos leidžia imituoti kompiuterinius žaidimus Atari kompiuteryje. T. y. agentas iš esmės mokosi laimėti ar pasiekti kuo didesnę rezultatą tam tikruose žaidimuose.
„Box2D“	Šioje kategorijoje sudėtos aplinkos leidžia imituoti objekto judėjimą dvimatėje erdvėje. T. y. agentas bando išmokyti vaikščioti, išvengti kliūčių, šokinėti, važiuoti ar kitaip judėti 2 dimensijų aplinkoje kuo efektyviau.
„Classic control“	Šioje kategorijoje sudėtos aplinkos, kurios imituoja tradicines mašininio mokymosi situacijas, aprašytas įvairioje literatūroje. Pvz., automobilio važiavimas iš duobės naudojant inerciją.
„MuJoCo“	Šioje kategorijoje sudėtos aplinkos iš esmės yra 3 dimensijų fizikos simulatoriai, kuriuose agentas bando atlikti tam tikras užduotis.
„Robotics“	Šioje kategorijoje sudėtos aplinkos bando simuliuoti įvairius tikslo siekiančius roboto veiksmus, pvz., kaladėlės paėmimas, kamuoliuko stūmimas ir t. t.

Kategorija	Apibūdinimas
„Toy text“	Šioje kategorijoje sudėtos aplinkos bando imituoti tam tikras įvairias supaprastintas gyvenimo situacijas, pvz., spėjimo žaidimą, žaidimą „BlackJack“, taksi judėjimą per miestą ir t. t.

Remiantis 3.1 Projektavimo metodo kontekstas ir reikalavimai poskyryje aprašytais reikalavimais aplinkų kategorijoms buvo vertinamas kiekvienos kategorijos pritaikomumas bei galimybė spręsti išsikeltus darbo uždavinius.

6 lentelė. OpenAI siūlomų kategorijų pritaikomumas tyrimo uždaviniams spręsti (sudaryta autoriaus)

Kategorija	Pritaikomumas
„Algorithms“	Šios kategorijos aplinkos yra visiškai nepritaikomos tyrimo uždaviniams spręsti, nes šioje kategorijoje siūlomos aplinkose veikiantys agentai bando spręsti aritmetinius veiksmus ir niekaip nebūtų įmanoma šiose aplinkose sukurti kortelių bei jų tarpusavio ryšių atitikmens. Tad išsikeltų reikalavimų ši kategorija neatitinka.
„Atari“	Šios kategorijos aplinkos yra visiškai nepritaikomos tyrimo uždaviniams spręsti, nes šioje kategorijoje siūlomos aplinkose veikiantys agentai imituoti kompiuterinius žaidimus Atari kompiuteryje, tad aplinkos bando spręsti labai specifines, iš anksto nustatytas tik žaidimų aplinkoje egzistuojančias užduotis ir niekaip nebūtų įmanoma sukurti kortelių bei jų tarpusavio ryšių atitikmens. Tad išsikeltų reikalavimų ši kategorija neatitinka.
„Box2D“	Šios kategorijos aplinkos yra visiškai nepritaikomos tyrimo uždaviniams spręsti, nes šioje kategorijoje siūlomos aplinkos iš esmės yra 2 dimensijų fizikos simuliatoriai, kuriuose agentas valdo kažkokį veikėją ir bando įveikti kliūtis. Tad niekaip nebūtų įmanoma sukurti kortelių bei jų tarpusavio ryšių atitikmens, todėl išsikeltų reikalavimų ši kategorija neatitinka.
„Classic control“	Šios kategorijos aplinkos yra visiškai nepritaikomos tyrimo uždaviniams spręsti, nes šioje kategorijoje siūlomos aplinkos sprendžia paprastus klasikiniuose tyrimuose aprašytus uždavinius, o sukurti kortelių bei jų tarpusavio ryšių atitikmens niekaip nebūtų įmanoma, todėl išsikeltų reikalavimų ši kategorija neatitinka.

Kategorija	Pritaikomumas
„MuJoCo“	Šios kategorijos aplinkos yra visiškai nepritaikomos tyrimo uždaviniams spręsti, nes šioje kategorijoje siūlomos aplinkos iš esmės yra 3 dimensijų fizikos simulatoriai, kuriuose agentas valdo kažkokį veikėją ir bando įveikti kliūtis. Tad niekaip nebūtų įmanoma sukurti kortelių bei jų tarpusavio ryšių atitikmens ar simuliuoti galimų reakcijų į korteles, todėl išsikeltų reikalavimų ši kategorija neatitinka.
„Robotics“	Šios kategorijos aplinkos yra visiškai nepritaikomos tyrimo uždaviniams spręsti, nes šioje kategorijoje siūlomos aplinkos iš dalies irgi yra 3 dimensijų fizikos simulatoriai, kai yra simuliuojami tam tikri fizinio roboto veiksmai. Tad niekaip nebūtų įmanoma sukurti kortelių bei jų tarpusavio ryšių atitikmens ar simuliuoti galimų reakcijų į korteles, todėl išsikeltų reikalavimų ši kategorija neatitinka.
„Toy text“	Dalis šioje kategorijoje siūlomų aplinkų gali atitikti reikalavimus. Šios kategorijos aplinkos simuliuoja labai įvairiapuses situacijas, iš kurių dalis leidžia sukurti arba išvesti kortelių atitikmenis jau siūlomų aplinkų elementams, taip pat kurti sąryšius tarp jų.

Iš visų kategorijų tyrimo reikalavimus bent dalinai atitinka tik aplinkos iš „Toy text“ kategorijos. Tačiau yra būtina įvertinti kiekvieną aplinką atskirai, nes siūlomos aplinkos yra labai skirtingos. „Toy text“ kategorijoje iš viso aplinkų yra 8 (sudaryta autoriaus):

- BlackJack – imituojamas kortų žaidimas „BlackJack“. Ši aplinka visiškai neatitinka tyrimo reikalavimų, nes yra imituojamas su kortelių paieška nesusijęs žaidimas.
- FrozenLake – agentas bando užšalusiu ežeru pereiti nuo pradžios taško iki finišo neįkrisdamas į vandenį. Teoriškai šioje aplinkoje galima kelti lygiagrečią tyrimo situacijai. Tokioje realizacijoje agentas ieškotų kelio nuo kaž kurios pradinės kortelės iki kortelės į kurią reaguojama teigiamai, bandant išvengti kortelių, į kurias reaguojama neigiamai. Tad ši aplinka bent dalinai atitinka tyrimo reikalavimus.
- GuessingGame – agentui reikia atspėti skaičių kuo arčiau atsitiktinai pasirinkto skaičiaus per kuo mažiau ėjimų. Ši aplinka visiškai neatitinka tyrimo reikalavimų, nes nors ir būtų įmanoma korteles sudėti į vieną eilę ir taip agentui jomis naviguoti. Tokia sistema turi iš anksto žinoti kuri kortelė yra teigiama, o šios informacijos žinoti tyrimo situacijoje dažnai neįmanoma.
- HotterColder – agentui reikia atspėti skaičių kuo arčiau atsitiktinai pasirinkto skaičiaus per kuo mažiau ėjimų ir jam nuolat yra sakoma ar spėjimas buvo geresnis ar blogesnis už praeitą. Ši

aplinka visiškai neatitinka tyrimo reikalavimų, nes nors ir būtų įmanoma kortelės sudėti į vieną eilę ir taip agentui jomis naviguoti. Tokia sistema turi iš anksto žinoti kuri kortelė yra teigiama, o šios informacijos žinoti tyrimo situacijoje dažnai neįmanoma.

- NChain – agentas bando judėti linijine grandine. Teoriškai ši aplinka dalinai atitiktų tyrimo reikalavimus sudėjus kortelės į eilę ir taip leidžiant agentui jomis naviguoti, tačiau joje agentas juda vienmatėje erdvėje, tad beveik nebūtų įmanoma išvengti neigiamų kortelių, o ir pats agento judėjimas būtų labai neefektyvus.
- Roulette – agentas bando išmokti laiminčią strategiją žaidime „Ruletė“. Ši aplinka visiškai neatitinka tyrimo reikalavimų, nes yra imituojamas su kortelėmis visiškai nesusijęs žaidimas.
- Taxi – agentas imituoja judėjimą mieste paimdamas ir išleisdamas keleivį tam tikrose vietose. Teoriškai šioje aplinkoje galima kelti lygiagrečią tyrimo situacijai, kai agentas rastų kortelę, ieškotų kelio nuo jos iki tam tikros kitos kortelės. Tad ši aplinka taip pat bent dalinai atitinka tyrimo reikalavimus.

Išanalizavus siūlomas OpenAI aplinkas pasirinkta toliau tyrinėti labiausiai tinkamas tyrimo situacijai *Taxi* bei *FrozenLake* aplinkas, nes šios abi aplinkos bent dalinai atitinka tyrimo reikalavimus, teoriškai abiejose aplinkose galima kelti lygiagretes tyrimo situacijai, kai *Taxi* aplinkoje agentas rastų kortelę, tuomet ieškotų kelio nuo jos iki tam tikros kitos kortelės, o *FrozenLake* aplinkoje agentas ieškotų kelio nuo kaž kurios pradinės kortelės iki kortelės į kurią reaguojama teigiamai, bandant išvengti kortelių, į kurias reaguojama neigiamai. Tačiau norint visapusiškai įvertinti abiejų aplinkų tinkamumą yra būtina su jomis individualiai daryti eksperimentinius tyrimus.

3.3.3. Pirmasis eksperimentinis tyrimas

Pirmajam eksperimentiniam tyrimui buvo pasirinktas OpenAI kurtas modelis pavadinimu *FrozenLake*. Tai yra žaidimas, kuris atitinkamai pateikus duomenis dalinai atitinka darbe vykdomo tyrimo reikalavimus. Žaidimo metu agentas bando nuo vienos atsitiktinės vietos nueiti iki kitos ir tai bandoma atlikti neįkrentant į skyles, kuriose agentas žūsta. Tad sudėjus kortelės į atitinkamą žemėlapiu formatą, t. y. pateikus jas taip, kad kiekviena eilutė būtų tam tikra kategorija, o kiekvienas stulpelis – tam tikra kortelė toje kategorijoje, galima vesti paralelų tyrimo išsikeltiems uždaviniams. Tokiu atveju, kiekviena atsitiktinė žemėlapiu vieta yra kortelė ant kurios lipdamas agentas ją rodo, o žaidimo esmė – nuo vienos rodomos kortelės nueiti iki kitos kortelės, į kurią vaikas reaguoja teigiamai, pakeliui neparodžius kortelės į kurias vaikas reaguoja neigiamai. Apibrėžto žaidimo *FrozenLake* atsitiktinio žaidimo būsenų paketas, kuris yra pateiktas asmenims suprantamu formatu – žemėlapiu yra pateikiamas žemiau:

SFFF

FHFH

FFFH

HFFG

Šiame žemėlapyje yra naudojama 4 simbolių anotacija:

S – žymi pradžios vietą, t. y. kortelę, kuri yra rodoma pradžioje. Tai laukelis ant kurio agentas gali lipti.

F – užšalęs paviršius, tarpinė neutrali kortelė einant link tikslo, dėl kurios vaikas nereaguoja nei neigiamai nei teigiamai. Tai laukelis ant kurio agentas gali lipti.

H – skylė lede, kortelė, kuri ASS turinčiam vaikui sukelia neigiamą reakciją. Tai laukelis ant kurio užlipęs agentas numiršta (pralošia žaidimą).

G – tikslas. Kortelė, kurią nori matyti vaikas ir kuri sukelia teigiamas emocijas. Užlipęs ant šios kortelės agentas laimi žaidimą.

FrozenLake žaidimo agentas geba žemėlapiu vaikščioti į 4 puses. Šiuos veiksmus taip pat galima būtų lyginti su tyrime atliekamais veiksmais:

- Veiksmas KAIRĖN – rodyti praėjusią kortelę toje pačioje kategorijoje.
- Veiksmas APAČION – rodyti kortelę iš kitos kategorijos.
- Veiksmas DEŠINĖN – rodyti kitą kortelę toje pačioje kategorijoje.
- Veiksmas VIRŠUN – rodyti kortelę iš praėjusios kategorijos.

Tyrimo atlikimui buvo sukurta sistema, kuri atitinkamai buvo apmokoma pasitelkiant Q naudos funkciją, o smulkiau apie Q naudos funkciją rašoma 3.3.1 Tyrimų aplinkos parametrų aprašymas poskyryje.

Pradžioje buvo atlikti tyrimai su 16 kortelių (t. y. 16 galimų Q lentelės būsenų) ir gauti rezultatai buvo teigiami. Gerai apmokytas agentas visada rasdavo kelią iki tikslo taikant 91 % tikimybę, kad laukelis nebus skylė (t. y. į kortelę bus reaguojama neigiamai). Paleidus keletą testų pastebėta, jog leidžiant 250000 epizodų apmokymams net 99 % visų epizodų dalis pasiekė tikslą, tad epizodų skaičių galima gerokai sumažinti (žr. 15 pav.).



15 pav. Apdovanojimų vidurkio priklausomybė nuo epizodų kiekio (tikimybė=0.73, learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, decayRate=0.00005, epizodų skaičius=250000, maxSteps=400) (sudaryta autoriaus, kinatmųjų aprašai)

Tuomet buvo atlikti tyrimai su 10000 epizodų nekeičiant tikimybės, jog laukelis nebus skylė. Pastebėta, kad net ir šiuo atveju net 92.04 % iš visų epizodų apmokymo metu pasiekė tikslą, tad buvo atlikti tyrimai su dar mažesniu kiekiu epizodų. Tik sumažinus epizodų kiekį iki maždaug 50, buvo pastebėta, jog apmokytas agentas pradeda nebe visada pasiekti tikslą testuose, kuriuose yra remiamasi tik apsimokiusia Q lentele (žr. 7 lentelė).

7 lentelė *FrozenLake* aplinkos testai kai bandoma kuo labiau mažinti apmokymo epizodų kiekį (tikimybė=0.91, learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.01, decayRate=0.01, maxSteps=400) (sudaryta autoriaus)

Epizodų skaičius	Epizodų, pasiekusių tikslą, dalis, [0;1]	Teisingai rezultatą rado iš 100 testų
250000	0.99	100
10000	0.92	100
1000	0.78	100
100	0.36	100
50	1	100
25	0.08	6
25	0.32	42
25	1	100
25	0.12	24

Epizodų skaičius	Epizodų, pasiekusių tikslą, dalis, [0;1]	Teisingai rezultatą rado iš 100 testų
50	1	100
50	0.24	37
50	0.3	100
50	0.24	100

Atliekant tyrimą su 16 būsenų pastebėta, jog modelis pradeda netinkamai veikti jei tikimybė, kad laukelis nebus skylė, yra ypač maža. Kadangi agentas gali judėti tik į 4 puses, o kiekvieną kartą agentui užlipus ant skylės epizodas yra nutraukiamas, gali egzistuoti situacija, kai Tikslas langelis gali būti nepasiekiamas. Pvz., žemėlapyje:

SFFF

FFFF

HHHH

FFFG

Aukščiau pateiktame žemėlapyje agentas tiesiog fiziškai negalėtų pasiekti tikslo, nes atsižvelgiant į agento judėjimo galimybes, visi keliai jam yra uždaryti. Teoriškai toks žemėlapis 16 būsenų aplinkoje gali būti sugeneruojamas net ir su $\sim 28\%$ tikimybe, kad laukelis nebus užšalęs. Būtent dėl šios priežasties *FrozenLake* modelio autoriai realizavimo aplinkos pergeneravimo funkcionalumą. Kiekvieną kartą prieš pradedant apmokymus aplinkos sistema tikrina ar agentas fiziškai gali pasiekti tikslą ir, jei šio testo rezultatas yra neigiamas, iš naujo generuojamas žemėlapis. Tokia situacija neatitinka realaus pasaulio situacijų, nes realiai naudotojui naudojant kortelių sistemą yra norima, kad sistema net ir pateikusi neigiamą kortelę naudotojui ir toliau teiktų korteles, o pats kortelių išdėstymas kistų kuo mažiau. Taip pat realiame pasaulyje gali būti situacijų, kai naudotojui neigiamas emocijas gali sukelti beveik visos kortelės, tad yra svarbu, kad agentas galėtų net ir užlipęs ant neigiamos kortelės tęsti darbą ir toliau ieškoti užduoties tikslo – teigiamos kortelės. Tai pagrindžia žemiau pavaizduota lentelė, kurioje yra matoma, kad kuo labiau mažinama tikimybė, kad kortelė nebus teigiama, tuo prasčiau tikslą pasiekia *FrozenLake* agentas (žr. 8 lentelę).

8 lentelė *FrozenLake* tyrimas, bandant kuo mažiau mažinti tikimybę, kad kortelė bus teigiama. (learningRate=0.8, gamma=0.9 maxEpsilon=1, minEpsilon=0.01, decayRate=0.01, maxEpisodes=1000, maxSteps=400) (sudaryta autoriaus)

Tikimybė, kad kortelė nebus teigiama	Epizodų, pasiekusių tikslą, dalis, [0;1]
50 %	0
50 %	0.63

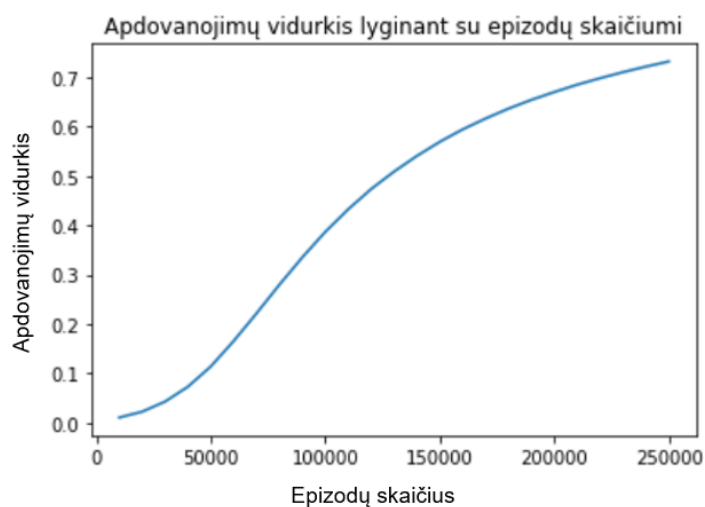
Tikimybė, kad kortelė nebus teigiama	Epizodų, pasiekusių tikslą, dalis, [0;1]
40 %	0
30 %	0
15 %	0
10 %	0
5 %	tyrimas nutrauktas po 15 minučių modeliui sėkmingai nesugeneravus žemėlapių

Norint tikslingiau simuliuoti realią pasaulio situaciją buvo toliau pasirinkta atlikti tyrimus su 81 kortele, suskirstyta į 9 kategorijas. Tuomet atitinkamai praplėstas žaidimo generuojamas žemėlapis, kad kurtų 9 x 9 atitikmens žemėlapi (žr. 16 pav.).

```
def generate_random_map(size=9, p=0.85):
    """Generuoja atsitiktinį tinkamą žemėlapi
    :param size: kiekvienos žemėlapi kraštinės dydis
    :param p: tikimybė, kad langelis yra užšalęs ir ant jo galima agentui lipti
    """
```

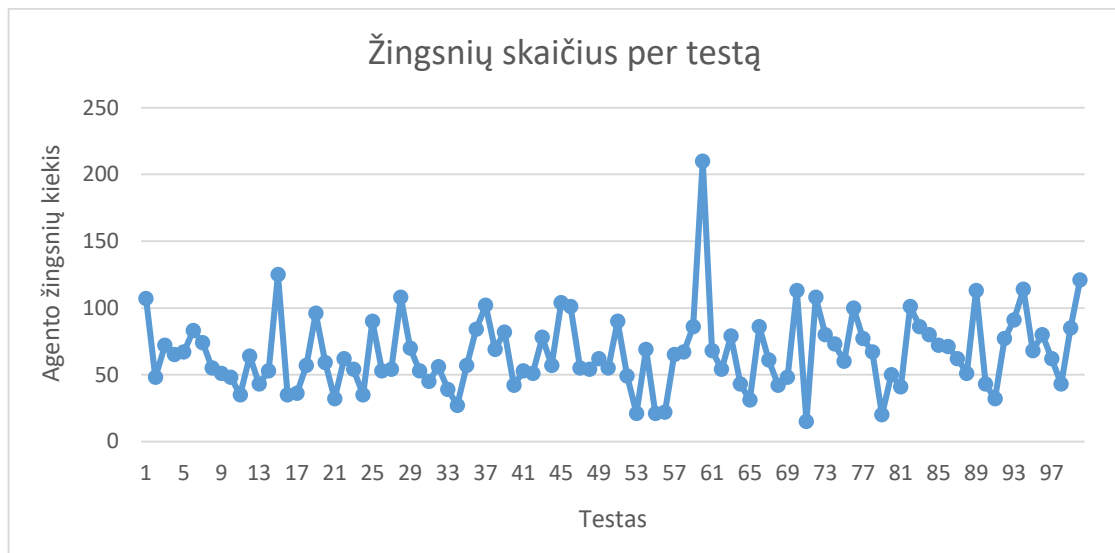
16 pav. Funkcija generuoti žemėlapi *FrozenLake* aplinkos kode (autorius rašyto kodo fragmentas)

Q apsimokymo funkcijos rezultatas per tam tikrą epizodų skaičių pateikiamas grafike žemiau (žr. 17 pav.). Taip pat iš diagramos galima matyti, kad rezultato kilimas maždaug nuo 150000 epizodų nebėra eksponentinis, o sulėtėja, tad ir mokyti modelį ilgiau nėra didelės prasmės.



17 pav. Apdovanojimų vidurkis lyginant su epizodų skaičiumi (tikimybė=0.91, learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, decayRate=0.00005, epizodų skaičius=250000, maxSteps=400) (sudaryta autoriaus)

Atliekant testus su 81 kortele pastebėta, kad apsimokęs modelis, simuliacijose, kuriose tikslą agentui yra įmanoma pasiekti, rezultatą grąžina teisingą. Tad galima teigti, kad ir tyrimo situacijoje vaikui būtų nerodomos kortelės, kurios jam itin nepatinka. Tačiau toks modelis dėl savo struktūros bei dalinamų taškų sistemos tai atlieta gana neefektyviai. Nors apmokytas modelis iš 100 testinių situacijų teisingai rado kelią 96-iose, vidutiniškai egzistuojant 81 kortelei, kiekvieno testo teisingam rezultatui pasiekti prireikė net 66.25 žingsnio (žr. 18 pav.).



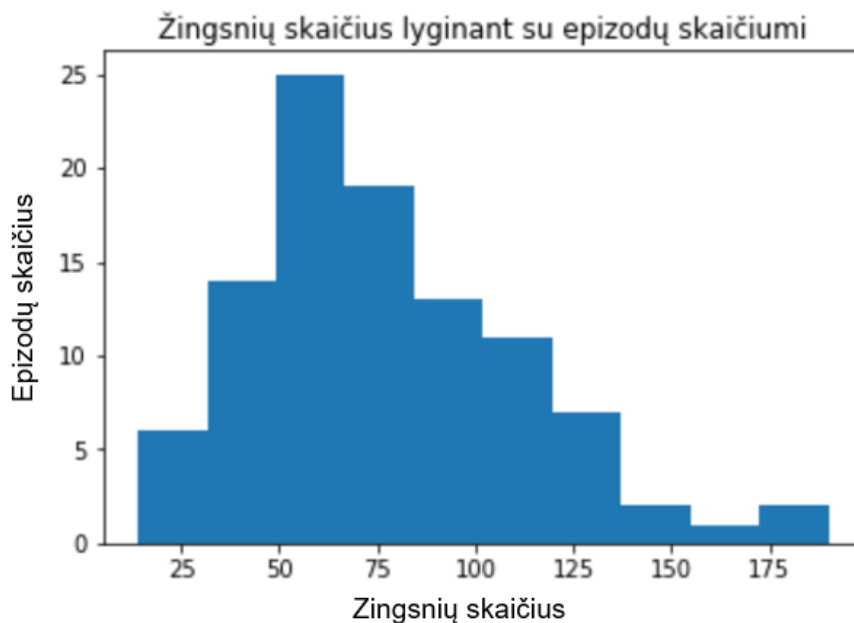
18 pav. Atliktų agento žingsnių kiekis testuose su idealiomis sąlygomis apmokytu agentu (tikimybė=0.91, learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, decayRate=0.00005, epizodų skaičius=250000, maxSteps=400) (sudaryta autoriaus)

Tai reiškia, kad testuojant su 81 kortele, testų metu buvo panaudota per didžiąją dalį visų kortelių arba tos pačios kortelės dažnu atveju buvo parodomos kelis kartus. Tad nors kortelių išdėstymas žemėlapyje bei 4 kryptių agento vaikščiojimas yra pakankama realizacija norint gauti teigiamą rezultatą. Specifiškai tyrimui kuriamame modelyje reikėtų agentą apdovanoti bei bausti ir už nueitą kiekį žingsnių. Tai pagrindžia ir apsimokymo metu padaromas žingsnių kiekis. Tyrime, kuriame tikimybė, kad laukelis nebus skylė – 91 %, apmokant modelį didžiojoje dalyje epizodų agentas atliko net apie 70 žingsnių. Žingsnių skaičius lyginant su epizodų skaičiumi minėtame tyrime yra pavaizduotas diagramoje žemiau (žr. 19 pav.).



19 pav. Agento žingsnių pasiskirstymas lyginant su epizodų skaičiumi (tikimybė=0.91, learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, decayRate=0.00005, epizodų skaičius=250000, maxSteps=400) (sudaryta autoriaus)

Testavimo metu agentas parodė dar prastesnį rezultatą ir vidutiniškai atliko net 77.21 žingsnio nors ir sėkmingai pasiekė tikslą 98 iš 100 kartų (žr. 20 pav.).



20 pav. Agento žingsnių pasiskirstymas atsitiktiniuose 100 testų lyginant su epizodų skaičiumi (tikimybė=0.91, learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, decayRate=0.00005, epizodų skaičius=250000, maxSteps=400) (sudaryta autoriaus)

Diagrama atvaizduoja žingsnių skaičių lyginant su epizodų skaičiumi atliekant testus (žr. 20 pav.). Tad *FrozenLake* aplinka iš dalies išsprendžia tyrimo keltus uždavinius. Tačiau yra svarbu pabrėžti analitinės analizės metu pastebėtą aspektą, kad sistema turi gerai veikti visiems naudotojams individualiai. Tad realybėje, realizavus tokią sistemą, gali egzistuoti situacija, kad naudotojui nepatinka tokios kortelės,

kad agentas, naviguojantis *FrozenLake* aplinkoje, fiziškai negali pasiekti tikslo. Taip pat dėl agentui suteikiamų apdovanojimų bei baudų sistemos aplinka nėra efektyvi ir idealiomis sąlygomis apmokytas agentas testuose atliko daug nereikalingų veiksmų ir ne visada eina greičiausiu keliu tikslo link.

3.3.4. Antrasis eksperimentinis tyrimas

Antrajam eksperimentiniam tyrimui buvo pasirinktas OpenAI kurtas modelis pavadinimu *Taxi*. Jis yra išskirtinis, nes agentas yra baudžiamas už kiekvieną padarytą žingsnį. *Taxi* yra kiek sudėtingesnė aplinka nei *FrozenLake*, nes turi kiek daugiau galimybių. Šio modelio aplinka yra žaidimas, kuris taip pat dalinai atitinka šio tyrimo keliamus reikalavimus. Šio žaidimo esmė yra paimti keleivius, nuvežti juos į atitinkamą vietą ir juos ten išleisti. Žaidime yra 4 miestai (pažymėti skirtingomis raidėmis), o agento darbas yra paimti keleivį vienoje vietoje ir išlaipinti atitinkamame mieste. Už sėkmingą išlaipinimą teisingoje vietoje gaunama 20 taškų, o už neteisėtai atliktus paėmimo ir išlaipinimo veiksmus numatyta 10 taškų bauda. Šis modelis, teoriškai, turėtų būti kiek optimalesnis tyrimo situacijai spręsti, nes agentas yra baudžiamas už kiekvieną atliktą žingsnį. Už tai yra prarandamas 1 taškas. Tad, teoriškai, *Taxi* aplinkoje veikiantis agento algoritmas turėtų būti suinteresuotas kuo greičiau pasiekti norimą tikslą. *Taxi* atsitiktinio žemėlapių pavyzdys bei būsenų paketas yra pateikiamas žemiau:

```
"-----+",  
"|R: | : :G|",  
"| : | : : |",  
"| : : : : |",  
"| | : | : |",  
"|Y| : |B: |",  
"+-----+",
```

Žemėlapyje visi tušti langeliai žymi paprastus laukelius ant kurių gali atsistoti agentas prarasdamas 1 tašką, tai reiškia apsimokęs agentas bandytų pasiekti reikiamą tikslą per kiek įmanoma mažiau žingsnių taip didinant agento efektyvumą. „|“ simboliai žemėlapyje žymi sienas, kurių kirsti agentui negalima ir jas reikia apvažiuoti aplinkui, o „:“ simboliai žymi sienas, kurias agentui kirsti galima. Pradėjus žaidimą atsitiktinėje vietoje atsiranda taksi automobilis bei atsitiktiniame mieste arba jau pačiame automobilyje atsiranda keleivis. Tuomet atsitiktinai išrenkamas keleivio tikslas. R, G, Y bei B raidės žymi „miestus“, į kuriuos taksi automobilio agentui reikia nuvežti ir išleisti keleivį. Tad žaidimo esmė – nusigauti iki keleivio, jį išlaipinti ir nuvežus iki reikiamos vietos išlaipinti.

Ši aplinka, panašiai kaip ir *FrozenLake* aplinka gali būti pritaikyta šio darbo tikslams pasiekti. Visi žemėlapių laukeliai gali būti traktuojami kaip kortelės, kurios būtų rodomos sistemos naudotojui. Šios kortelės yra aplinkoje išdėliojamos žemėlapiu formatu. Tuomet aplinkoje veikiančio agento veiksmai gali atitikti:

0: judėti Žemyn – rodyti kortelę iš kitos kategorijos.

- 1: judėti Aukštyn – rodyti kortelę iš praėjusios kategorijos.
- 2: judėti Dešinèn – rodyti kitą kortelę toje pačioje kategorijoje.
- 3: judėti Kairèn – rodyti praėjusią kortelę toje pačioje kategorijoje.
- 4: paimiti keleivį – išrinkti pirmą naudotojui patinkančią kortelę.
- 5: išleisti keleivį – išrinkti antrą naudotojui patinkančią kortelę, kuri yra susijusi su pirmąja.

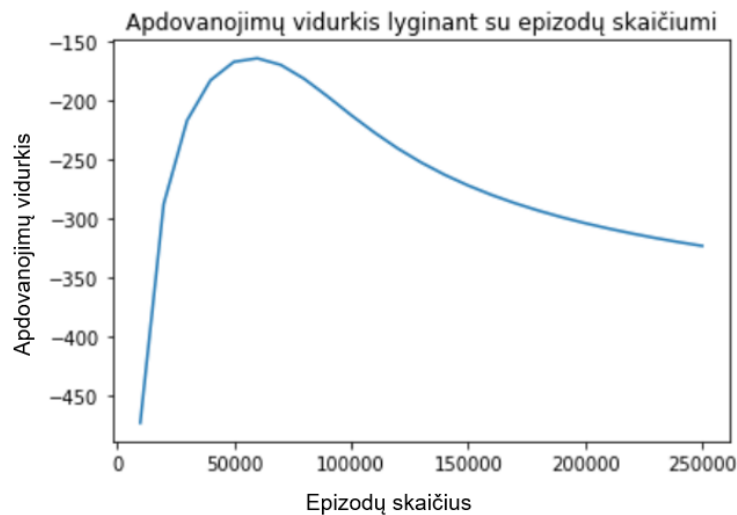
Tad naudotojui yra pradedama rodyti atsitiktinė kortelė, tuomet keliaujama iki pradžios kortelės, t. y. kortelės, kuri patinka naudotojui ir ją parodžius ir užfiksavus bandoma keliauti iki kitos kortelės, kuri, anot naudotojo, yra susijusi su pirma kortele ir jam lygiai taip pat patinka.

Atliekant tyrimus pastebėta, kad *Taxi* aplinkos modelis, kaip ir *FrozenLake* aplinka, pritaikyta tyrimo situacijai rezultatą grąžina teisingą, t. y. pateikia vaikui patinkančias korteles bei suranda korteles, kurios yra susijusios. *Taxi* aplinkoje veikiantis agentas užduotį įvykdo vidutiniškai greičiau nei *FrozenLake* aplinkoje, nes agentas yra nuolat baudžiamas už padarytus žingsnius, tačiau pats agento judėjimas kortelių žemėlapyje yra neefektyvus, nes galima judėti tik po vieną kortelę ar kategoriją į kažkurį šoną, taip agentą priverčiant prarasti daug taškų.

Taip pat reikia pabrėžti, kad *Taxi* aplinkoje visos kortelės yra traktuojamos kaip teigiamos arba neutralios, o iš analizės metu įvardintų galimų reakcijų į korteles matome, kad tai nėra teisinga tyrimo situacijoje. Realiam pasaulyje gali egzistuoti žmogus, kuriam kažkuri dalis kortelių gali ir nepatikti.

Atliekant tyrimus pastebėta, kad nors lyginant su *FrozenLake*, *Taxi* aplinkoje yra 2 papildomi apibrėžti veiksmai, nei vienas jų nespirtina pačio agento judėjimo žemėlapyje, tad nors agento judėjimas yra gerokai efektyvesnis dėl nustatomos baudos už kiekvieną žingsnį, vis tiek yra atliekama pakankamai daug žingsnių, ypač padidinus kortelių skaičių. Pavyzdžiui, tyrime su 81 kortele kilo situacija, kai agentas atlikęs virš 200 žingsnių, bet galiausiai vis tiek pasiekė epizodo tikslą ir sėkmingai atliko užduotį. Tikėtina, kad jei tokia situacija pasikartotų tikroje aplinkos realizacijoje, naudojantis tikram naudotojui, jis tiesiog nebūtų išlaukęs teigiamo epizodo rezultato ir būtų tiesiog nustojęs naudotis sistema.

Taip pat situacijose, kai agentas atlikdavo labai daug žingsnių, vien dėl žingsnių skaičiaus ir egzistuojančios taškų sistemos agentas vidutiniškai gaudavo daug baudos taškų. Anksčiau minėtoje situacijoje, kai yra atliekama 300 žingsnių, agentas yra tuo pat metu nubaudžiamas -300 balų, tad aplinkos duodami +20 taškų už teisingai atliktus veiksmus tiesiog tampa nereikšmingi. Dėl šios priežasties kartais pasitaikydavo situacijų, kai mokomas agentas nesugebėdavo tinkamai apsimokyti ir tiesiog darydavo veiksmus, kurie jį bausdavo mažiau. T. y. vaikščiodavo ratais tol, kol būdavo pasiekiamas nustatytas epizodo žingsnių limitas. Tokį agento vengimą judėti ir išleidinėti keleivį (t. y. vykdyti aplinkos tikslą) galima matyti diagramoje žemiau (žr. 21 pav.).



21 pav. Apdovanojimų vidurkis lyginant su epizodų skaičiumi (learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, decayRate=0.00005, epizodų skaičius=250000, maxSteps=400) (sudaryta autoriaus)

Padidinus apdovanojimą už sėkmingai atliktą užduotį problema išsisprendė, tad apdovanojimo kiekis privalo būti atitinkamai proporcingas agento judėjimo galimybėms bei gautoms neišvengiamoms baudoms.

Taip pat *Taxi* aplinkoje veikiantis agentas pradėdavo veikti prasčiau padidinus kortelių, o tuo pačiu metu ir būsenų kiekį. Net ir visaip bandant varijuoti parametrais, padidinus galimų būsenų kiekį iki 81 kortelės, agentas, dėl galimų veiksmų gausos, pradėjo nebeapsimokyti per pakankamai mažą epizodų kiekį. Tai pagrindžianti lentelė yra pavaizduota žemiau (žr. 9 lentelė).

9 lentelė *Taxi* aplinkos apsimokymo duomenys padidinus galimų kortelių skaičių iki 81. (learningRate=0.8 gamma=0.9 maxEpsilon=1 maxEpisodes=250000) (sudaryta autoriaus)

minEpsilon	decayRate	maxSteps	Epizodų, pasiekusių tikslą, dalis, [0;1]	Epizodų, pasiekusių tikslą, dalis (iš paskutinių 100 epizodų), [0;1]	Vidutinis žingsnių skaičius	Vidutinis žingsnių skaičius (iš paskutinių 100 epizodų)
0.001	0.00005	400	0.1649	0.18	335.86	328.93
0.01	0.00005	1000	0.1659	0.19	836.52	811.47
0.01	0.0005	1000	0.1654	0.16	835.84	840.85
0.01	0.00005	10000	sustabdyta po 42280 epizodų, nes per 40000 epizodų tikslą pasiekė tik 6675 epizodai (t. y. apie 16 %)			

Atsižvelgiant į aukščiau pateikiamus duomenis galima teigti, kad OpenAI kurta *Taxi* aplinka be didelių pakeitimų yra pilnai tinkama naudoti tik tais atvejais, kai galima reakcija į korteles yra arba teigiama arba neutrali, nes agentas veikdamas šioje aplinkoje nėra apmokomas vengti rodyti kai kurių kortelių ir visas jas rodo vienodai dažnai. Ši aplinka geriausiai veikia, kai galimų kortelių kiekis yra mažas, o jam didėjant didėja ir tikimybė, kad apsimokymo etape algoritmas užtruks neefektyviai daug laiko. Taip pat nors agento judėjimas šioje aplinkoje yra efektyvesnis nei *FrozenLake*, agentas vis tiek gali judėti tik po vieną kategoriją arba po vieną kortelę toje kategorijoje, tad tikslo pasiekimas gali užtrukti didėjant galimų kortelių skaičiui.

3.3.5. Atliktų tyrimų rezultatai ir išvados

Iš atliktų empirinių tyrimų nustatyta, kad apmokytam tyrimo modeliui norint atvaizduoti tinkamas korteles, tinkamas kortelių būsenų atvaizdavimo būdas yra jas vaizduoti žemėlapyje ir leisti per jį naviguoti agentui. Tačiau pastebėta, kad norint tai atlikti efektyviau, nepakanka aplinkoje taikyti tik 4 agento judėjimo aplinkoje veiksmus, t. y. kortelių žemėlapyje suteikti galimybę agentui judėti į 4 kryptis. Taip pat atliktų tyrimų rezultatai parodė, kad yra svarbu taikyti tinkamą agento baudų bei prizų logiką, o svarbiausia yra tinkamai bausti agentą ir už nueitą kiekį žingsnių. Tyrimo metu apsimokymo laikas ir efektyvumas nėra labai svarbus, daug svarbiau, kad jau apmokyta sistema teisingai atliktų sprendimus. Atsižvelgiant į šiuos pastebėjimus galima teigti, kad labiausiai tyrimui atlikti tinka Q naudos funkcijos tipo algoritmas.

3.4. Projektavimo metodo individualizuoto algoritmo kūrimas

Apibendrinus duomenų Q lentelę, galimas jos būsenas bei galimus agento veiksmus, galima atitinkamai struktūrizuoti tyrimo tikslinei auditorijai pritaikytą aplinką. O remiantis analitinės dalies ir atliktų tyrimų išvadomis darbe buvo išsikelti reikalavimai individualizuotos aplinkos kūrimui ir algoritmo apmokymui. Jų laikymasis leis tiksliau atspindėti ASS turinčių asmenų patirtį. Nustatyti reikalavimai aplinkai yra pateikiami žemiau:

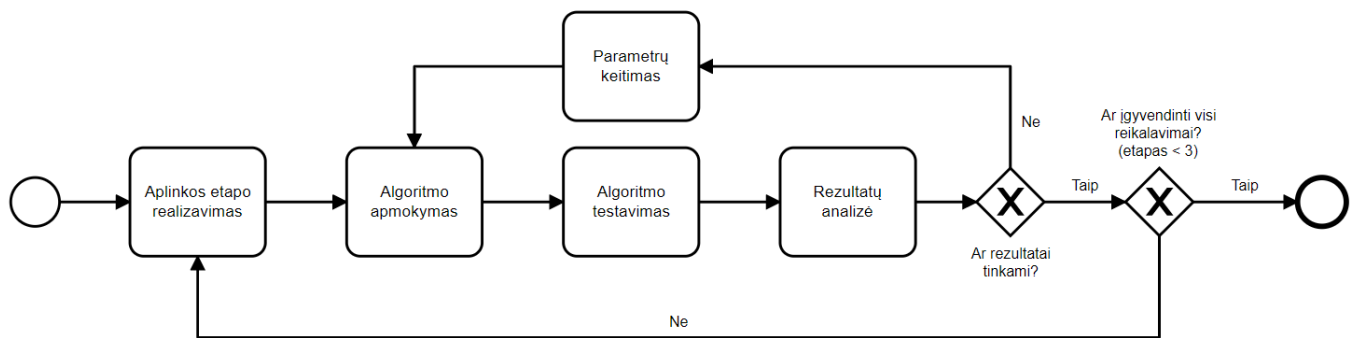
1. Remiantis dalykinės srities analize nustatyta, kad duomenų rinkiniui optimaliausia naudoti 81 kortelę, suskirstytą į 9 kategorijas. Taip pat atsižvelgiant į būsimų naudotojų ypatumus nustatyta, kad algoritmas turėtų veikti tinkamai ir su gerokai mažesniu ar didesniu kortelių kiekiu.
2. Atliktų tyrimų rezultatai parodė, kad norint optimizuoti agento judėjimą yra būtina sukurti sąryšį tarp kortelių, taip leidžiant agentui judėti tarp jų.
3. Atlikus dalykinės srities analizę nustatyta, kad kortelių rinkinį tyrimo metu galima adaptuoti pateikti lentelės principu, kai lentelės eilutės – kategorijos, o stulpeliai – kategorijose esančios

kortelės. Šis išdėstymas taip pat patvirtino atliekant tyrimus su skatinamojo mokymosi algoritmais.

4. Atliktų tyrimų rezultatai parodė, kad norint turėti efektyvesnį aplinkos algoritmą yra būtina agentui suteikti galimybę judėti ir tarp atsitiktinių kategorijų, kurios nebūtinai yra šalia viena kitos taip gerokai spartinant jo judėjimą.
5. Atliktų empirinių tyrimų su kitų autorių kurtomis aplinkomis bei tikslinės auditorijos analizės metu nustatyta, kad agento apmokymams minimaliai turėtų užtekti maždaug 100 epizodų mokymų, tačiau agentas turi tiksliau veikti su didesniu epizodų kiekiu.
6. Atliktų tyrimų rezultatai parodė, kad yra svarbu taikyti tinkamą agento baudų bei prizų logiką, o svarbiausia yra tinkamai bausti agentą ir už nueitą kiekį žingsnių.

3.4.1. Algoritmo kūrimo eiga

Remiantis aukščiau išsikeltais reikalavimais buvo kurta specialiai šiam tyrimui pritaikyta skatinamojo mokymosi aplinka. Buvo pasirinkta aplinkos algoritmą testuoti taip pat kaip buvo testuojami kitų autorių kurti algoritmai, o pačio algoritmo realizaciją kurti iteratyviai 3 etapais. Šios aplinkos kūrimo etapus darbo metu galima rasti diagramoje žemiau (žr. 22 pav.).



22 pav. *CardFinder* algoritmo sukūrimo ir testavimo diagrama (sudaryta autoriaus)

Duomenų rinkinys per kurį gali vaikščioti agentas susideda iš 81 PECS kortelės, suskirstytos į 9 skirtingas kategorijas, tačiau aplinka turi veikti tinkamai ir su gerokai mažesniu ar didesniu būsenų kiekiu. Tuomet aplinkoje sąveikaujantis agentas gali atlikti šiuos veiksmus:

- Rodyti praėjusią kortelę toje pačioje kategorijoje.
- Rodyti kortelę iš kitos kategorijos.
- Rodyti kitą kortelę toje pačioje kategorijoje.
- Rodyti kortelę iš praeitos kategorijos.

- Rodyti kitą kortelę susijusią su dabar naudojama naudojant metaduomenis.

Aplinkoje sąveikaujantis agentas yra apdovanojamas ir baudžiamas pagal žemiau apibrėžtas taisykles (angl. *policy*):

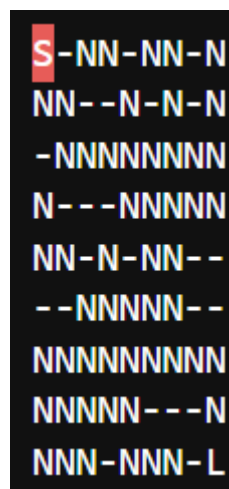
- 1 taškas už kiekvieną parodytą kortelę (agento nueinamą žingsnį).
- +20 taškų už parodytą kortelę į kurią vaikas reaguoja labai teigiamai.
- +1 taškas už parodytą kortelę į kurią vaikas reaguoja teigiamai.
- 10 taškų už parodytą kortelę į kurią vaikas reaguoja neigiamai.

Būtent taip apibrėžus taisykles algoritmas turėtų vengti neigiamų kortelių bei korteles į kurias vaikas reaguoja teigiamai pasiekti kuo greičiau.

Individualizuotos aplinkos kūrimas

Individualizuotos aplinkos kūrimas vyko 3 etapais, kurie yra suskirstyti pagal didesnius kodo pakeitimus bei įvestą naują aplinkos ar agento funkcionalumą. Kiekvieno etapo metu buvo atitinkamai testuojami atlikti pakeitimai.

1 etape buvo sukurta aplinka, kurioje buvo realizuota: 4 agento veiksmai (viršun, žemyn, kairėn, dešinėn); 81 kortelė; baudos/apdovanojimai: Nueitas žingsnis: -1, parodyta labai neigiama kortelė: -10, parodyta teigiama kortelė: +20; galimos kortelės žemėlapyje: S – startas, L – tikslas, „-“ – neutrali kortelė, N – stipriai neigiama kortelė. Taip pat įvestas -1 baudos taškas už kiekvieną parodytą kortelę/nueitą žingsnį. Norint geriau simuliuoti realią situaciją yra svarbu parodyti neigiamą kortelę bandyti korteles rodyti toliau. Todėl aplinkos epizodo metu parodyta kortelė (išskyrus labai teigiamą) neužbaigia epizodo. Pirmojo etapo žemėlapio pavyzdys pateiktas žemiau (žr. 23 pav.).



23 pav. 1-ojo etapo žemėlapio pavyzdys (30 % tikimybė, kad kortelė bus neigiama) (sudaryta autoriaus)

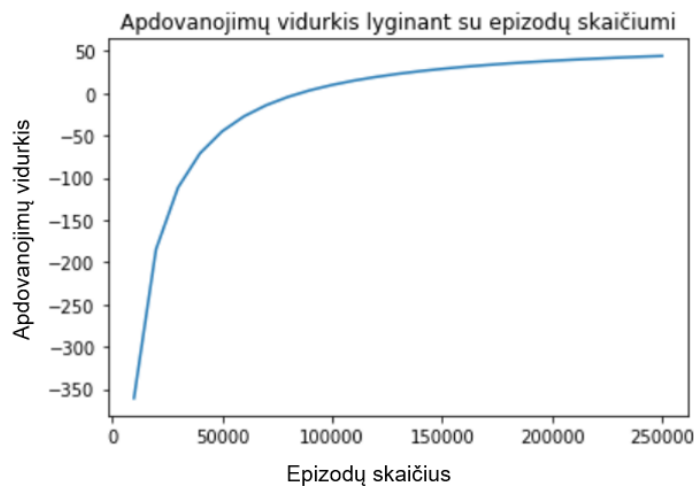
Norint užtikrinti sėkmingą aplinkos algoritmo darbą prototipo realizacijoje buvo naudojama dalis bibliotekų: *numpy* biblioteka buvo naudojama pradinių lentelės reikšmių užpildymui, aukščiausios reikšmės iš tam tikros Q lentelės eilutės paieškai bei eksponentinės parametro *decayRate* nuvertėjimo reikšmės paskaičiavimui. Apsimokymo modelio rezultatų pateikimui buvo naudojama *matplotlib* biblioteka. Pačios skatinamojo mokymosi aplinkos inicializavimui bei paleidimui iš naujo buvo naudojama biblioteka *gym*.

Tiriant pimojo etapo modelį buvo pastebėtos kylančios tokios pačios problemos kaip ir *Taxi* aplinkoje, kai agentas tiesiog vengė atlikinėti užduoties tikslą ir tiesiog vaikščiojo ratais iki kol baigėsi epizodo žingsnių limitas. Net ir pakėlus apdovanojimą už teisingai atliktą užduotį nuo 20 iki 50 taškų problema neišsprendė. Su 50 % tikimybe, kad langelis nebus teigiamas apdovanojimų vidurkis lyginant su epizodų skaičiumi atrodė taip (žr. 24 pav.):



24 pav. Apdovanojimų vidurkis lyginant su epizodų skaičiumi (tikimybė=0.5, learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, decayRate=0.00005, epizodų skaičius=250000, maxSteps=400) (sudaryta autoriaus)

2 etape buvo sprendžiamos 1 etape pastebėtos problemos. Norint paskatinti agentą daugiau judėti tikslo link buvo nuspręsta padidinti prizą už sėkmingai atliktą užduotį pakėlus nuo 50 iki 100 taškų. Šios problemos sprendimo sėkmę patvirtinantis grafikas yra pateikiamas žemiau (žr. 25 pav.).



25 pav. Apdovanojimų vidurkis lyginant su epizodų skaičiumi po pakeitimų (tikimybė=0.5, learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, decayRate=0.00005, epizodų skaičius=250000, maxSteps=400) (sudaryta autoriaus)

Tuomet norint simuliacijoje atvaizduoti realias, specialistų apibrėžtas tikslinės auditorijos reakcijas į korteles, aplinkoje buvo įvestas dar vienas galimas kortelės ženklėjimas. Po pakeitimų, generuojant žemėlapi „T“ raide pažymėti laukeliai atvaizduoja „teigiamas“ asmens reakcijas į tarpines korteles, bet aplinkoje veikiančio agento tikslas vis tiek išlieka surasti labai patinkančią kortelę (žemėlapyje pažymėtą „L“). Tai atlikus, į aplinką buvo įvesta pagrindinė savybė, kuri spartina agento judėjimą žemėlapyje. Tai yra galimybė agentui judėti pagal kortelių metaduomenis.

3 etape buvo bandoma spartinti agento judėjimą žemėlapyje suteikiant jam galimybę judėti ne tik į 4 puses. Buvo sugalvota taikyti judėjimo pagal metaduomenis agento veiksmą, kai kiekviena kortelė yra susijusi su dar viena kažkuria kortele. Tyrimo aplinkoje teoriškai kiekviena kortelė gali būti susijusi su bet kuria kita kortele ir algoritmas turėtų tinkamai veikti su bet kokia sąsajų konfigūracija. Norint tai kuo tiksliau imituoti simuliacijoje buvo pasirinkta generuojant atsitiktinį žemėlapi atsitiktinai generuoti ir kortelių tarpusavio ryšius. Naudojantis „Python“ programavimo kalbos komanda *numpy.random.choice* su išjungtu pernaudojimo parametru kiekvienos kortelės objektui buvo priskiriama kitos kortelės *id*, kuriuo naudojantis agentas yra žemėlapyje perkeliamas į susijusios kortelės vietą.

S - - - NN - N -	[[36 77 26 35 42 4 18 7 44]	S - - - NN - N -
NNNN - - NNN	[27 5 51 52 19 8 58 12 21]	NNNN - - NNN
N - NN - NNNN	[20 33 55 37 1 76 53 0 69]	N - NN - NNNN
NNNNN - NN -	[29 78 32 75 56 22 59 25 80]	NNNNN - NN -
N - NNNN - N -	[28 50 6 3 17 49 79 40 62]	N - NNNN - N -
- NNNNNTN -	[31 72 65 57 45 66 38 9 41]	- NNNNNTN -
- N - - - - N -	[43 54 11 73 61 13 46 34 74]	- N - - - - N -
N - - NN - N - T	[60 68 30 47 39 64 14 16 10]	N - - NN - N - T
- NNN - NN - L	[71 63 23 67 2 48 70 15 24]]	- NNN - NN - L

26 pav. 3 – ojo etapo judėjimo pagal metaduomenis pavyzdys (sudaryta autoriaus)

Tad rodydamas kiekvieną kortelę agentas gali nauju „šulio pagal metaduomenis“ veiksmu keliauti į susijusią kortelę. Aukščiau pateiktame paveikslėlyje (žr. 26 pav.) raudonas langelis pažymi kortelę, kurią šiuo metu rodo agentas (būseną, kurioje yra agento mašina). Paveikslėlyje pateiktoje situacijoje iš dabartinės (kairėje pusėje pavaizduotos) „-“ būsenos agentas galėtų šokti į kortelę, kuri turi 80 identifikacijos numerį. Tad pasitelkdamas naujai apibrėžtą veiksmą, agentas gali judėti tiesiai iki užduoties tikslo atlikdamas tik vieną žingsnį ir galiausiai minėtoje konfigūracijoje šio veiksmo dėka sėkmingai atlikdamas užduotį ir nueidamas nuo „S“ iki „L“ vos per 4 žingsnius.

Atlikus šiuos pakeitimus aplinka turėjo: 5 veiksmus (viršun, žemyn, kairėn, dešinėn, judėjimas į kitą kortelę, kuri nėra šalia (metaduomenys)); 81 būseną (kortelę); baudos/apdovanojimai: nueitas žingsnis: -1, parodyta neigiama kortelė: -10, parodyta teigiama kortelė: +100, parodyta teigiama kortelė: +1.

Tuomet tyrimo metu buvo atitinkamai keičiama tikimybė, kad kortelė bus teigiama norint įsitikinti, kad agentas teisingai ir greitai randa kelią įvairiose situacijose. Dalį atliktų tyrimų rezultatų galima matyti lentelėje žemiau (žr. 10 lentelė).

10 lentelė. Dalis tyrimų rezultatų keičiant tikimybę, kad kortelė bus teigiama (learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, decayRate=0.00005, epizodų skaičius=250000, maxSteps=400) (sudaryta autoriaus)

Tikimybė, kad kortelė bus teigiama	Žingsnių kiekis apsimokymo metu (kode parametras sum(steps_total))	Žingsnių kiekis testų metu (kode parametras sum(steps_total_test))
90 %	6.26	4
70 %	3.96	2.02
50 %	6.39	4
30 %	3.92	2

Tikimybė, kad kortelė bus teigiama	Žingsnių kiekis apsimokymo metu (kode parametras sum(steps_total))	Žingsnių kiekis testų metu (kode parametras sum(steps_total_test))
20 %	3.82	2
10 %	5.06	3
5 %	6.38	4.02

Iš aukščiau pateiktos lentelės galima matyti, kad net ir tokiais atvejais, kai simuliuojama, kad naudotojui nepatinka net 95 % kortelių, agentas vis tiek greitai ir sėkmingai atlikdavo apibrėžtą užduotį. O tai dažniausiai būdavo atliekama per mažiau nei 5 žingsnius (žr. 10 lentelė).

Tad modelis aplinkoje visais atvejais gana greitai surado tikslą bei puikiai veikė net ir su labai maža tikimybe, kad kortelė bus teigiama. Tuomet buvo tiriamas apibrėžtos aplinkos galimybės kiek įmanoma greičiau apsimokyti pradedant nuo 20000 epizodų ir baigiant 500. Norint testuoti tik aplinką, buvo pasirinkta nustatyti 30 % tikimybę, kad kortelė nebus teigiama, visoms simuliacijoms, taip pat nekeisti kitų Q naudos funkcijos parametrų (žr. 11 lentelė).

11 lentelė. Tyrimai atlikti varijuojant epizodų kiekiu (tikimybė=0.3, learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, decayRate=0.00005, maxSteps=400) (sudaryta autoriaus)

Epizodų skaičius	Epizodų, pasiekusių tikslą, dalis, [0;1]	Iš 100 testų teisingai tikslą pasiekė	Vidutinis žingsnių skaičius per testą
200000	0.99	100	6
150000	0.99	100	5
100000	0.99	100	3
50000	0.99	100	3
45000	0.99	100	2
40000	0.99	100	3
30000	0.99	100	4
25000	0.99	100	3
20000	0.99	100	6
15000	0.99	100	2
10000	0.99	100	3
5000	0.99	100	2
2500	1	100	4
2000	0.99	100	2

Epizodų skaičius	Epizodų, pasiekusių tikslą, dalis, [0;1]	Iš 100 testų teisingai tikslą pasiekė	Vidutinis žingsnių skaičius per testą
1500	0.99	100	2
1000	1	100	2
500	0.99	100	2

Iš atliktų tyrimų galima matyti, kad aplinkoje dirbantis agentas gerai apsimoko net ir su mažu kiekiu epizodų (žr. 11 lentelė). Pastebėjus tokius rezultatus buvo bandyta dar labiau mažinti epizodų kiekį apmokymuose. Žinoma buvo būtina optimizuoti epsilon decayRate dydį norint užtikrinti, kad agentas kiek įmanoma anksčiau pradėtų atlikinėti veiksmus pagal išminktą duomenų Q lentelę.

12 lentelė. Tyrimai atlikti varijuojant epizodų kiekį bei epsilon vertės decayRate (tikimybė=0.3, learningRate=0.8, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, maxSteps=400) (sudaryta autoriaus)

Epizodų skaičius	Epizodų, pasiekusių tikslą, dalis, [0;1]	Iš 100 testų teisingai rezultatą rado	Vidutinis žingsnių skaičius per testą
400	0.99	100	5
300	1	100	2
200	0.99	100	4
100	0.99	99	3

Analizuojant tokius rezultatus galima teigti, kad aplinkos algoritmas sugeba apsimokyti net per itin mažą mokymosi epizodų kiekį, kai agentas jau pradėdavo tinkamai bei per mažą atliktų veiksmų kiekį pasiekti tikslą net ir apmokinus jį vos 100 epizodų (žr. 12 lentelė).

Pastebėjus tokius rezultatus buvo bandoma kiek įmanoma labiau mažinti epizodų kiekį iki tol, kol sistema pradės rodyti neteisingus rezultatus. Norint užtikrinti, kad agentas mokintųsi kiek įmanoma daugiau buvo nuspręsta padidinti learningRate parametą iki 0.9 bei nekeisti epsilon mažinimo decayRate parametro. Dėl šių priežasčių galutinė epsilon vertė apmokymų pabaigoje turėtų kilti mažėjant epizodų skaičiui.

13 lentelė. Tyrimai kai buvo kritiškai mažinamas epizodų kiekis (tikimybė=0.3, learningRate=0.9, Gamma=0.9, maxEpsilon=1, minEpsilon=0.001, maxSteps=400) (sudaryta autoriaus)

Epizodų skaičius	Epizodų, pasiekusių tikslą, dalis, [0;1]	Vidutinis žingsnių skaičius apmokymų metu	Vidutinis žingsnių skaičius per testą	Epsilon vertė apmokymų pabaigoje
50	0.98	55.48	4	0.613014
40	1	46.2	1	0.67738
40	1	44.17	2	0.67738
30	1	65.67	5	0.748515
15	1	113.73	5	0.869489

Iš atliktų tyrimų (žr. 13 lentelė) matyti, kad net ir kritiškai sumažinus epizodų kiekį iki 15 ir tuomet bandant testuoti sistemą atliekant 100 atsitiktinių testų, agentas vis tiek teisingai bei per tik 5 žingsnius pasiekė tikslą testuose. Tačiau svarbu pastebėti, kad tyrime su 15 epizodų, jiems pasibaigus dėl nustatyto epsilon vertės funkcionalumo, agentas teorinio 16 epizodo metu vis dar turėtų epsilon vertę rodančią, kad su 86 % tikimybe agentas turėtų atlikti atsitiktinį veiksmą norint geriau pažinti aplinką.

3.4.2. Siūlomos apsimokymo aplinkos aprašymas

Išanalizavus aplinką bei atlikus tyrimus galiausiai buvo apsisistota su šiame poskyryje pateikiama aplinka bei jos logika. Žemiau yra pateikiama aplinkos bei agento veiksmų diagrama (žr. 27 pav.). O nuoroda į pilną aplinkos bei tyrimų kodą yra pateikiama šio darbo prieduose (žr. 1 priedas).

Aplinkos duomenų išdėstymas yra toks: aplinkoje naudojamos kortelės pateikiamos žemėlapyje formate, kai kiekviena eilutė yra tam tikra kategorija, o kiekvienas stulpelis – tam tikra kortelė toje kategorijoje. Žemėlapyje yra naudojami 5 simboliai:

1. „-“ – neutrali kortelė.
2. „S“ – starto kortelė.
3. „L“ – itin teigiamą reakciją iššaukianti kortelė (tikslas kortelė).
4. „T“ – teigiamą reakciją iššaukianti kortelė.
5. „N“ – neigiamą reakciją iššaukianti kortelė.

Taip pat sukurtoje aplinkoje funkcionuojantis agentas gali atlikti 5 veiksmus:

- Veiksmas 1 – rodyti praėjusią kortelę toje pačioje kategorijoje.
- Veiksmas 2 – rodyti kortelę iš kitos kategorijos.
- Veiksmas 3 – rodyti kitą kortelę toje pačioje kategorijoje.
- Veiksmas 4 – rodyti kortelę iš praėjusios kategorijos.
- Veiksmas 5 – judėjimas į kitą kortelę, kuri nėra šalia (pagal kortelės metaduomenis).

Sukurtoje aplinkoje yra nustatomos agentui teikiamos baudos: agento nueitas žingsnis: -1, parodyta neigiama kortelė: -10.

Sukurtoje aplinkoje yra nustatomi agentui teikiami apdovanojimai: parodyta teigiama kortelė: +1, parodyta labai teigiama kortelė (tikslas) +100.

Aukščiau pateikiama aplinka, atlikus tyrimus, leido pasiekti geresnius rezultatus bei geriau išsprendė analizės metu išsikeltas problemas. Tačiau ir kitos tyrinėtose aplinkos gali būti naudojamos tokiems uždaviniams spręsti, jei yra pasiekiami atitinkami reikalavimai.

3.5. Tyrimo rezultatų apibendrinimas

Atlikus darbo tyrimus buvo gauti žemiau pateikti rezultatai:

1. Tiriant OpenAI kurtą *FrozenLake* aplinką buvo atliktos 26 parametrų rinkinių iteracijos. Tyrimuose su 10000 epizodų, 16 galimų būsenų bei 91 % tikimybe, jog laukelis nebus skylė pastebėta, kad net 92.04 % iš visų epizodų apmokymo metu pasiekė tikslą. Tad buvo atlikti tyrimai vis mažinant epizodų kiekį. Tik sumažinus epizodų kiekį iki maždaug 50, buvo pastebėta, jog apmokytas agentas pradeda nebe visada pasiekti tikslą testuose. Atliktuose 500 testų su 50 apmokymo epizodų kiekiu, tikslas buvo pasiektas 437 testuose, o atliktuose 400 testų su 25 apmokymo epizodais tikslas buvo teisingai pasiektas tik 172 iš jų (žr. 7 lentelė). Taip pat tiriant *FrozenLake* aplinką pastebėta, kad agentas prasčiau apsimoko, o kartais net visai nesugeba apsimokyti mažėjant tikimybei, kad kortelė bus teigiama. Šių tyrimų metu sumažinus tikimybę iki 50 % agentas neatliko užduoties 2 iš 3 kartų leidžiant su tais pačiais parametrais. O sumažinus tikimybę dar labiau (iki 40 %, 30 %, 15 %, 10 % bei 5 %) agentas nei vieną kartą nesugebėjo tinkamai apsimokyti bei atlikti užduotį apmokant jį 1000 epizodų.
2. Tiriant OpenAI kurtą *Taxi* aplinką buvo atliktos 25 parametrų rinkinių iteracijos. Jose pastebėta, jog agentas dėl nustatytos apdovanojimų bei baudų logikos nuolat eina trumpiausiu keliu link tikslo, tačiau neatsižvelgia į galimas reakcijas į korteles. Taip pat pastebėta, kad aplinkoje veikiantis agentas pradėjo veikti prasčiau padidinus kortelių, o tuo pačiu metu ir būsenų kiekį. Net ir visai bandant varijuoti parametrais (atlikus 5 parametrų iteracijas), padidinus galimų būsenų kiekį iki 81 kortelės agentas, dėl galimų veiksmų gausos, nesugebėjo apsimokyti net per 250000 epizodų (žr. 9 lentelė).
3. Tiriant darbo autoriaus kurtą *CardFinder* aplinką buvo atlikta ~ 40 skirtingų parametrų iteracijų. Tyrimų metu nustatyta, kad agentas gerai apsimoko ir mažinant tikimybę, kad kortelė bus teigiama net iki 5 % bei net ir tokiais atvejais kai egzistuoja 81 galima būsena, tikslas yra pasiekiamas agentui atlikus mažiau nei 5 žingsnius (žr. 10 lentelė). Taip pat buvo nustatyta, kad šioje aplinkoje veikdamas agentas sugeba apsimokyti per itin mažą epizodų kiekį, kai testuojant su 81 galima kortele, net ir sumažinus epizodų kiekį iki 15 vis tiek agentas pasiekė tikslą per apytiksliai 5 žingsnius (žr. 13 lentelė).

4. Išvados

Šiame darbe buvo apibrėžta darbo problematika, tikslai bei uždaviniai, iškeltam darbo tikslui pasiekti. Atlikus su darbo tematika susijusios mokslinės literatūros analizę, išnagrinėjus kitų autorių atliktus tyrimus, atlikus eksperimentus su kitų autorių kurtomis aplinkomis ir eksperimentus su sukurta aplinka galima daryti darbo metu gautas bendrąsias išvadas:

1. Išnagrinėjus ASS turinčių naudotojų patirtį ir komunikacijos ypatumus nustatyta, kad PECS kortelės yra naudojamos komunikacijos sistemose dėl pateikiamos informacijos paprastumo bei aiškumo. Dėl tikslinės auditorijos specifikos kiekvienam PECS sistemos naudotojui pati sistema turi būti adaptuota bei individualizuota pagal jo poreikius. Taip pat dėl naudotojų individualumo nėra įmanoma sukaupiti iš anksto paruošto duomenų rinkinio.
2. Išanalizavus kitų autorių atliktus ASS turintiems asmenims skirtų sistemų mašininio mokymosi algoritmų tyrimus buvo pastebėta, kad didžioji jų dalis tiria pačios PECS sistemos efektyvumą, bet nėra atsižvelgiama į galimas naudotojų reakcijas norint gerinti apsimokymą. Taip pat tyrimai, kuriuose buvo taikomas skatinamasis mokymasis, pasiekė teigiamų rezultatų. Dėl šių priežasčių bei norint atlikti tikslius skaičiavimus ir išvengti sunkiai nuspėjamų situacijų, tyrime pasirinkta naudoti skatinamojo mokymosi algoritmus paremtus Q naudos funkcija.
3. Eksperimentų su OpenAI kurtomis *FrozenLake* bei *Taxi* aplinkomis metu pastebėta, kad šios aplinkos tik dalinai tinka tyrimo uždaviniui spręsti, nes:
 - OpenAI kurta *FrozenLake* aplinka pilnai įvykdo tyrimo reikalavimus tik tais atvejais, kai tikimybė, kad kortelė bus neigiama, nėra didelė. Net ir su ~ 28 % tikimybe, testuojant su 16 galimų kortelių, gali egzistuoti situacija kai agentas negalės pasiekti tikslo, o šiai tikimybei didėjant, kartu didėja tikimybė, kad agentas veikdamas aplinkoje išvis negalės baigti darbo.
 - OpenAI kurta *Taxi* aplinka be didelių pakeitimų yra pilnai tinkama naudoti tik tais atvejais, kai galima reakcija į korteles yra arba teigiama arba neutrali, nes agentas veikdamas šioje aplinkoje nėra apmokomas vengti rodyti kai kurių kortelių ir visas jas rodo vienodai dažnai. Taip pat nors agento judėjimas šioje aplinkoje yra efektyvesnis nei *FrozenLake*, agentas vis tiek gali judėti tik po vieną kategoriją arba po vieną kortelę toje kategorijoje, tad tikslas gali būti pasiekiamas per daugiau žingsnių didėjant galimų kortelių skaičiui.
4. Remiantis tyrimų ir atliktų eksperimentų rezultatais buvo sukurta naudotojo sąsajos apsimokymo *CardFinder* aplinka, visiškai atitinkanti tyrimo iškeltus reikalavimus, o įvedus judėjimą tarp kortelių pagal metaduomenis, vidutiniškai 81 kortelės aplinkoje tikslas pasiekiamas itin greitai – per ~ 4 agento žingsnius. Ši aplinka, lyginant su *FrozenLake* bei *Taxi*, sugeba apsimokyti per itin mažą mokymosi epizodų kiekį, kai agentas jau pradėdavo 81 būsenos žemėlapyje tinkamai bei greitai pasiekti tikslą net ir apmokinus jį vos 15 epizodų.

5. Rekomendacijos ir papildomos tyrimų kryptys

Darbo metu atlikus tyrimus ir eksperimentus pateikiamos rekomendacijos naudotojo sąsajos projektavimui su specifine naudotoju grupe: siūlymai kuriant mašininio mokymosi paremtas skaitmenizuotas PECS sistemas ASS turintiems asmenims:

1. Situacijose, kai yra kuriamos skatinamojo mokymosi aplinkos ir tikslinę auditoriją sudaro itin specifinės grupės naudotojai, jiems reikia projektuoti naudotojo sąsajas tyrimo būdu, nes dažniausiai neįmanoma apibrėžti išankstinių reikalavimų bei pradinio duomenų rinkinio.
2. Situacijose, kai nėra turimas iš anksto sužymėtų duomenų rinkinys, rekomenduojama naudoti skatinamąjį mokymąsi su Q naudos funkcija, nes taip išvengiama sunkiai nuspėjamų situacijų.

Taip pat darbo metu atlikus tyrimus galima pateikti keletą papildomų tyrimų krypčių:

- Siekiant didinti praktinę vertę: darbo prototipo testavimas buvo atliekamas simuliuojant galimas naudotojo reakcijas į rodomas korteles, o dėl realių naudotojų naudojimo duomenų rinkinio neegzistavimo bei pačių duomenų rinkimo komplikacijų prototipas nebuvo testuojamas su tikrais naudotojų duomenimis. Nors ir tyrimai buvo bandomi atlikti kuo arčiau imituojant realią situaciją bei nuolat konsultuojantis su specialistais norint pilnai ištestuoti kiekvienos aplinkos panaudojamumą bei realų efektyvumą būtų reikalinga jas testuoti su dideliu kiekiu realių naudotojų duomenų analizuojant realią bei ilgalaikę jų patirtį.
- Siekiant gilinti mokslinį tyrimą ir didinti jo mokslinę vertę: tyrimas buvo atliktas su skatinamojo tipo Q naudos funkcija paremtu mašininio mokymosi algoritmu, nes nebuvo turima išankstinių naudotojo duomenų bei buvo bandoma išvengti neaiškių, sunkiai apibrėžiamų situacijų. Turint realių naudotojų duomenų būtų galima efektyviau taikyti kitus algoritmus, pvz., taikyti rekomendacijos variklius, kuriais mėginti lyginti naudotojus tarpusavyje. Taip pat tyrimą bei testavimą atliekant su realiais naudotojais būtų galima taikyti giliuosius neuroninius tinklus, nes tokiu atveju visos dėl neuroninių tinklų kylančios neapibrėžtos algoritmo elgesio situacijos galėtų būti testuojamos bei pagrindžiamos realių naudotojų sistemos naudojimo patirtimis.

Literatūros sąrašas

- Altameem, T., Amoon, M., & Altameem, A. (2020). A deep reinforcement learning process based on robotic training. *Neural Computing and Applications*.
- Autismag. (2020). all 40 PECS Communication card categories. Nuskaityta iš <https://www.autismag.org/members/all-40-pecs-communication-card-categories/>
- Ayodele, T. O. (2010). Types of Machine Learning Algorithms. Esantis *New Advances in Machine Learning*. Nuskaityta iš https://books.google.lt/books?hl=en&lr=&id=XAqhDwAAQBAJ&oi=fnd&pg=PA19&dq=types+of+machine+learning+algorithms&ots=r2FrdZygIq&sig=CR4ncJkuP8bFxlLcbK7g18Nbs4U&redir_esc=y#v=onepage&q=types%20of%20machine%20learning%20algorithms&f=false
- Baio, J., Wiggins, L., Christensen, D. L., Maenner, M. J., Daniels, & Dowling, J. (2018). Prevalence of Autism Spectrum Disorder Among Children Aged 8 Years — Autism and Developmental Disabilities Monitoring Network, 11 Sites, United States, 2014.
- Baird, L., & Moore, A. (1999). Gradient Descent for General. Nuskaityta iš https://www.ri.cmu.edu/pub_files/pub1/baird_leemon_1999_1/baird_leemon_1999_1.pdf
- Balakrishnama, S., & Ganapathiraju, A. (1998). LINEAR DISCRIMINANT ANALYSIS - A BRIEF TUTORIAL. Mississippi: INSTITUTE FOR SIGNAL AND INFORMATION PROCESSING. Nuskaityta iš http://www.music.mcgill.ca/~ich/classes/mumt611_07/classifiers/lda_theory.pdf
- Cai, L., & Zhu, Y. (2015). The Challenges of Data Quality and Data Quality Assessment in the Big Data Era. Nuskaityta iš <http://doi.org/10.5334/dsj-2015-002>
- Carman, A. (2018). This is how Instagram's algorithm populates your feed. Nuskaityta iš www.theverge.com/2018/6/1/17418254/instagram-algorithm-how-it-works-posts-downrank
- Chan, J. C., Chan, K. P., & Yeh, A. G. (2001). Detecting the nature of change in an urban environment: A comparison of machine learning algorithms. *Photogrammetric Engineering and Remote Sensing*.
- Chowdhary, K. R. (2020). Natural Language Processing. *Fundamentals of Artificial Intelligence*, pp 603–649.
- Clement, J. (2019 m. Gruodžio 3 d.). *Mobile application user retention rate worldwide from 2012 to 2019*. Nuskaityta iš Statista: <https://www.statista.com/statistics/751532/worldwide-application-user-retention-rate/>
- Degrís, T., White, M., & Sutton, R. S. (2013). Off-Policy Actor-Critic. Nuskaityta iš <https://arxiv.org/abs/1205.4839>
- Ding, S., Zhao, X., Xu, X., Sun, T., & Jia, W. (2019). An effective asynchronous framework for small scale reinforcement learning problems. Nuskaityta iš <https://link.springer.com/article/10.1007%2Fs10489-019-01501-9>

- Doherty, A., Bracken, M., & Gormley, L. (2018). Teaching Children with Autism to Initiate and Respond to Peer Mands.
- Doshi, S. (2019). Brief on Recommender Systems. Nuskaityta iš <https://towardsdatascience.com/brief-on-recommender-systems-b86a1068a4dd>
- Ghahramani, Z. (2003). Unsupervised Learning. *Lecture Notes in Computer Science*, 72–112.
- Google Inc. (2020). <https://developers.google.com/machine-learning/recommendation/overview/types>. *Recommendation Systems Overview*. Nuskaityta iš <https://developers.google.com/machine-learning/recommendation/overview/types>
- Gupta, A., & Chaurasiya, V. (2019). Reinforcement Learning Based Energy Management in Wireless Body Area Network: A Survey.
- Huang, H., Hei, X., Gao, Y., & Zhang, C. (2020). Design an Applied-Behavior-Analysis Learning WeChat Tool to Assess the Learning Capacities for Autistic Children. 2020 IEEE International Conference on Teaching, Assessment, and Learning for Engineering.
- IONOS Inc. (2020). Data mining tools for better data analysis. Nuskaityta iš <https://www.ionos.com/digitalguide/online-marketing/web-analytics/a-comparison-of-data-mining-tools/>
- Jordan, M., & Mitchell, T. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, Vol 349, pp. 255-260.
- Keyes, K. M., Susser, E., Cheslack-Postava, K., Fountain, C., Liu, K., & Bearman, P. S. (2011 m. Gruodžio 7 d.). Cohort effects explain the increase in autism diagnosis among children born from 1992 to 2003 in California. *International Journal of Epidemiology*, 496-503. Jungtinės Amerikos Valstijos: Oxford University Press on behalf of the International Epidemiological Association.
- Khabbaz, A. H., Pouyan, A. A., Fateh, M., & Abolghasemi, V. (2019). An Adaptive Learning Game for Autistic Children using Reinforcement. *Journal of AI and Data Mining*, 7, No 2.
- Kohonen, T. (1995). Learning vector quantization. *Self-organizing maps*. Springer, Berlin, Heidelberg.
- Kung-Hsiang, H. (2018). Introduction to Various Reinforcement Learning Algorithms. Part I (Q-Learning, SARSA, DQN, DDPG). *Towards Data Science*. Nuskaityta iš <https://towardsdatascience.com/introduction-to-various-reinforcement-learning-algorithms-i-q-learning-sarsa-dqn-ddpg-72a5e0cb6287>
- Kuźelewska, U. (2020). Effect of Dataset Size on Efficiency of Collaborative Filtering Recommender Systems with Multi-clustering as a Neighbourhood Identification Strategy. *Lecture Notes in Computer Science*, 12139. Nuskaityta iš https://link.springer.com/chapter/10.1007/978-3-030-50420-5_25
- Leidykla Presvika. (2022). BENDRAUKIME PAVEIKSLĖLIAIS (UGDymo ir BENDRAVIMO PAVEIKSLĖLIŲ PRIEMONIŲ RINKINYS). NAMAI. UGDYMO ĮSTAIGA. MARGARITA

JUREVIČIENĖ, TOMA JOKUBAITIENĖ. Nuskaityta iš <https://www.presvika.lt/bendraukime-paveikslėliais>

- Li, M., Gu, X., Zeng, C., & Feng, Y. (2020). Feasibility Analysis and Application of Reinforcement Learning Algorithm Based on Dynamic Parameter Adjustment.
- Li, M., Li, X., Xie, L., Liu, J., Wang, F., & Wang, Z. (2019). Assisted therapeutic system based on. *Computer Assisted Surgery*. Informa UK Limited, trading as Taylor & Francis Group.
- Li, W., Qiu, X., Li, Y., Ji, J., Liu, X., & Li, S. (2021). Towards a novel machine learning approach to support augmentative. *International Journal of Speech Technology*.
- Lokesh, A. (2019). A Comparative Study of Recommendation Systems. Nuskaityta iš <https://digitalcommons.wku.edu/cgi/viewcontent.cgi?article=4171&context=theses>
- Mahoney, B., Johnson, A., McCarthy, M., & White, C. (2018). *Systematic Review: Comparative Efficacy of the Picture Exchange Communication System (PECS) to Other Augmentative Communication Systems in Increasing Social Communication Skills in Children with Autism Spectrum Disorder*. Vermontas, Jungtinės Amerikos Valstijos: University of Vermont.
- Malandraki, G. A., & Okalidou, A. (2007). The application of PECS in a deaf child with autism: A case study.
- Malandraki, G. A., & Okalidou, A. (2007). The Application of PECS in a Deaf Child With Autism: A Case Study. Nuskaityta iš <https://doi.org/10.1177/10883576070220010301>
- Marizel, B., & Louella, S. M. (2018). Bitter Melon Crop Yield Prediction using Machine Learning Algorithm. *International Journal of Advanced Computer Science and Applications*.
- Matson, J. L., Mahan, S., Hess, J. A., Fodstad, J. C., & Neal, D. (2010). Progression of challenging behaviors in children and adolescents with Autism Spectrum Disorders as measured by the Autism Spectrum Disorders-Problem Behaviors for Children (ASD-PBC). *Research in Autism Spectrum Disorders*.
- Molnar, C. (2019). *A Guide for Making Black Box Models Explainable*.
- Murphy, R. (2019). Introduction to AI robotics.
- My PECS. (2022). PECS Card Categories. Nuskaityta iš <http://www.mypecs.com/categories.aspx>
- Nagpal, A., & Gabrani, G. (2019). Python for Data Analytics, Scientific and Technical Applications. *2019 Amity International Conference on Artificial Intelligence (AICAI)*. Dubai, United Arab Emirates: IEEE. Nuskaityta iš https://ieeexplore.ieee.org/abstract/document/8701341?casa_token=HSmL_PCMXhEAAAAA:WqyuYAMODb20NQryvkiV37fx91nEkRVWvvH4N01EGxLrTFnm0ZRdc3WApJlK_xT4tGS4NYjUKWQ

- National Autism Resources. (2022). *The Picture Exchange Communication System (PECS)*. Nuskaityta iš National Autism Resources: <https://www.nationalautismresources.com/the-picture-exchange-communication-system-pecs/>
- Nilsson, N. J. (2009). *The quest for artificial intelligence*. Cambridge University Press.
- Pal, A., Parhi, P., & Aggarwal, M. (2017). An improved content based collaborative filtering algorithm for movie recommendations. *2017 Tenth International Conference on Contemporary Computing (IC3)*. Noida, India.
- Pujahari, A., & Padmanabhan, V. (2015). Group Recommender Systems: Combining User-User and Item-Item Collaborative Filtering Techniques. *2015 International Conference on Information Technology (ICIT)*. Bhubaneswar, India.
- Qiang, W., & Zhongli, Z. (2011). Reinforcement learning model, algorithms and its application. *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*.
- Rudovic, O., Zhang, M., Schuller, B., & Picard, R. W. (2019). Multi-modal Active Learning From Human Data: A Deep Reinforcement Learning Approach.
- Shapiro, L., & Stockman, G. (2001). Computer vision. *Vol. 3*.
- Slobodin, O. P., Heffler, K. F., & Davidovitch, M. M. (2019 m. Gegužė). Screen Media and Autism Spectrum Disorder. *Journal of Developmental & Behavioral Pediatrics*, 40(4), 303-311.
- Soomro, N., & Soomro, S. (2018). Autism Children's App using PECS. *Annals of Emerging Technologies in Computing*, 2. Nuskaityta iš <https://arxiv.org/ftp/arxiv/papers/1801/1801.03529.pdf>
- Stančin, I., & Jović, A. (2019). An overview and comparison of free Python libraries for data mining and big data analysis. *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics*.
- Vicker, M. B. (2002). *What is the Picture Exchange Communication System or PECS?* Indiana, Jungtines Amerikos Valstijos: Indiana University Bloomington, Indiana Institute on Disability and Community, Indiana Resource Center for Autism. Nuskaityta iš <https://www.iidc.indiana.edu/irca/articles/what-is-the-picture-exchange-communication-system-or-pecs.html>
- Wenlin, W., & Jingkai, J. (2020). Review of reinforcement learning research. 020 35th Youth Academic Annual Conference of Chinese Association of Automation (YAC).
- World Health Organization. (2019). Nuskaityta iš International Statistical Classification of Diseases and Related Health Problems 10th Revision: <https://icd.who.int/browse10/2019/en#/F84.0>
- Yang, Y., & Webb, G. I. (2008). Discretization for naive-Bayes learning: managing discretization bias and variance. *Machine learning* 74.1.
- Zhao, M., Lu, H., Yang, S., & Guo, F. (2020). The Experience-Memory Q-Learning Algorithm for Robot Path Planning in Unknown Environment.

6. Priedai

1 priedas. Pilnas atliktų tyrimų katalogas

Visas su tyrimu susijusias modifikuotas aplinkas, pseudokodą bei atliktus tyrimus galima rasti Github kolekcijoje nurodytu adresu:

<https://github.com/augustmik/Card-Finder-Masters>

2 priedas. MyPECS svetainėje siūlomos kortelių kategorijos

Visos MyPECS svetainėje siūlomos kortelių kategorijos

Kategorija	Subkategorijos
Veiklos	Pramogos Kasdienės veiklos Bendrieji veiksmai Muzika Žaidimai Sportas Terapija
Gyvūnai	Paukščiai Dinozaurai Žinduoliai Vandens gyvūnai Ropliai
Bendrieji dalykai	Abėcėlė Spalvos Valiutos Atostogos Mėnesiai Sezonai Formos Laikas
Apranga	Kojos Galva Apatinė kūno dalis Kiti drabužiai Viršutinė kūno dalis
Komunikacija	Elgesys Emocijos Bendroji komunikacija Skausmas Jausmai

Kategorija	Subkategorijos
Maistas	Pusryčiai Desertai Greitas maistas Vaisiai Mėsa ir kt. Užkandžiai ir kt. Daržovės
Namai	Vonios kambarys Šeimos kambarys Virtuvė Kiti
Žmonės	Kūno dalys Grupės Individai
Vietos	Priežiūros patalpos Bažnyčia Laukas Mokykla Žaislų parduotuvė
Pagalba sau	Apsirengimas Bendroji pagalba Asmeninė priežiūra
Įrankiai	Rankiniai įrankiai Kiti įrankiai Elektriniai įrankiai
Transportas	Oro Žemės Karinis Geležinkelio

3 priedas. 25-ojoje Lietuvos jaunųjų mokslininkų konferencijoje „Mokslas – Lietuvos ateitis“ šio darbo tema pristatyto pranešimo pažymėjimas



4 priedas. 25-ojoje Lietuvos jaunųjų mokslininkų konferencijoje „Mokslas – Lietuvos ateitis“ šio darbo tema pristatyto pranešimo sertifikatas



Viešoji įstaiga, Saulėtekio al. 11, 10223 Vilnius, tel.: (8 5) 274 5000, (8 5) 274 5030, faks. (8 5) 270 0112, el. p. vilniustech@vilniustech.lt
Duomenys kaupiami ir saugomi Juridinių asmenų registre, kodas 111950243, PVM mokesčio kodas LT119502413
Katedros duomenys: Saulėtekio al. 11, SRL-I, 424 kab., 10223 Vilnius, tel. (8 5) 274 4829, el. p. isk@vilniustech.lt
Saulėtekio al. 11, LT-10223 Vilnius, Lithuania, tel.: +370 5 274 5000, +370 5 274 5030, fax +370 5 270 0112, e-mail vilniustech@vilniustech.lt
Department: Saulėtekio al. 11, SRL-I, room 424, LT-10223 Vilnius, Lithuania, tel. +370 5 274 4829, e-mail isk@vilniustech.lt

SERTIFIKATAS / CERTIFICATE

2022-04-14, Vilnius

Šiuo sertifikatu patvirtinama, kad vadovaujantis 25-osios Jaunųjų mokslininkų konferencijos sekcijos „IT sauga ir informacinės sistemos“ taisyklėmis, studentui: / *By this certificate we are to confirm that according to the rules of the 25th Conference of Young Scientist section „IT security and information systems“ student:*

Augustas Mikulėnas

yra skiriama 1.0 papildomo balo prie galutinio baigiamojo darbo gynimo pažymio už kokybišką tyrimą ir įdomų pranešimą JMK konferencijoje, vykusioje 2022 m. balandžio 13 d. / *is awarded with 1.0 extra points that will be added to the final thesis evaluation grade for the high quality research and interesting presentation at the CYS conference, that took place on 13th of April, 2022.*

Mokslo komiteto pirmininkas /
Head of scientific committee

doc. dr. / Assoc. Prof. Dr. Nikolaj Goranin

5 priedas. Vaiko raidos centro vadovės dr. Jovitos Petrulytės atsiliepimas



VIEŠOJI ĮSTAIGA
VILNIAUS UNIVERSITETO LIGONINĖ
SANTAROS KLINIKOS

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETO
FUNDAMENTINIŲ MOKSLŲ FAKULTETO
INFORMACINIŲ SISTEMŲ KATEDRAI

ATSILIEPIMAS
AUGUSTO MIKULĖNO MAGISTRO DARBUI
2022-05-25

Autizmo spektro sutrikimą turintiems vaikams padėti bendrauti su jo aplinka naudojamos alternatyviosios komunikacijos sistemos. PECS paveikslėlių apsikeitimo sistema šiuo metu yra viena plačiausiai naudojamų tokių priemonių.

Šiuo metu egzistuojančios skaitmeninės PECS paveikslėlių apsikeitimo realizacijos nėra pakankamai efektyvios ir plačiai ištirtos. Dėl autizmo spektro sutrikimą turinčių asmenų specifikos, kuriant jiems skirtas sistemas yra svarbu jas pritaikyti kiekvieno žmogaus individualiems poreikiams.

Studento Augusto Mikulėno magistro darbe „Mašininio mokymusi grindžiamas naudotojo sąsajos kūrimas specialiųjų poreikių turintiems naudotojams“ atliktas tyrimas ieškant efektyvesnių būdų kaip pateikti alternatyviosios komunikacijos sistemos realizacijas yra naudingas, o jo metu gauti rezultatai yra reikšmingi, siekiant ateityje vystyti praktines skaitmenizuotų PECS sistemų realizacijas.

Vaiko raidos centro vadovė

dr. Jovita Petrulytė