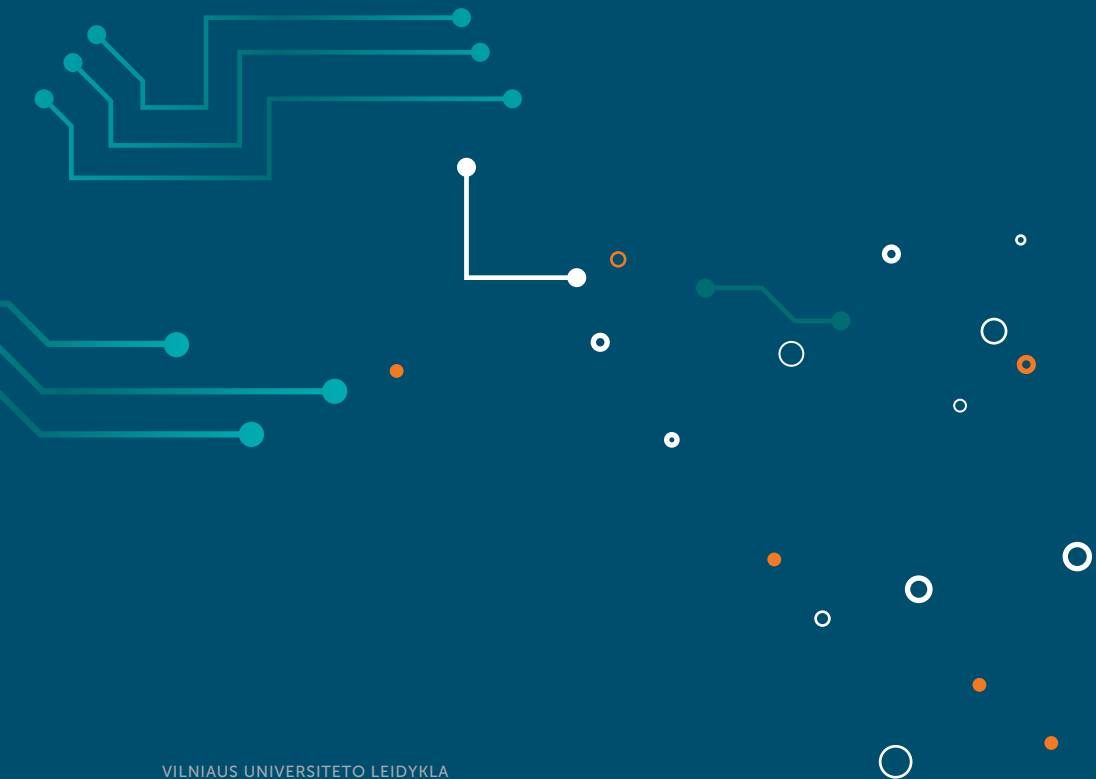




Konferencijos „Lietuvos magistrantų informatikos ir IT tyrimai“ darbai



2022 m.





Konferencijos „Lietuvos magistrantų informatikos ir IT tyrimai“ darbai

2022 m. gegužės 16 d.



VILNIAUS
UNIVERSITETO
LEIDYKLA

2022

Programinis komitetas:

Dr. Jolita Bernatavičienė

Prof. habil. dr. Gintautas Dzemyda

Prof. dr. Olga Kurasova

Prof. dr. Julius Žilinskas

ISBN 978-609-07-0742-5 (skaitmeninis PDF)

<https://doi.org/10.15388/LMITT.2022>

Autorių teisės © Autoriai, 2022. Publikavo Vilniaus universiteto leidykla.

Tai yra atvirosios prieigos žurnalas. Žurnalas platinamas vadovaujantis *Kūrybinių bendrijų licencija – Priskyrimas (CC BY)*, kuri leidžia laisvai ir neribotai naudoti, kaip mano esant reikalinga, be kita ko, kopijuoti, daryti pakeitimus ir kurti išvestinius kūrinius, naudoti komerciniais ir nekomerciniais tikslais nurodant informacijos šaltinį ir autorių.

Ižanga

Konferencija „Lietuvos magistrantų informatikos ir IT tyrimai“ skirta pristatyti magistrų baigiamųjų darbų tyrimus informatikos ir IT srityse. Šio renginio tikslas – pakelti magistrantų įgūdžius, supažindinti su kitų magistrantų atliekamais tyrimais, paskatinti domėtis moksline veikla. Konferencijoje savo pranešimus skaitys studentai iš Kauno technologijos universiteto ir Vilniaus universiteto.

Konferenciją organizuoja Lietuvos mokslų akademija (LMA) kartu su Vilniaus universitetu. LMA – tai įstaiga, jungianti žymiausius Lietuvos ir savo veikla su Lietuva susijusius mokslininkus. Ji yra nepriklausoma Seimo, Vyriausybės ir jai pavaldžių institucijų ekspertė ir patarėja mokslo bei studijų, kultūros, socialinės raidos, ūkio, gamtosaugos, sveikatos apsaugos, technologijų bei kitais klausimais. LMA įgyvendina šalies mokslui ir eksperimentinei plėtrai reikšmingus ES struktūrinių fondų projektus, rengia šalies bei tarptautines mokslines konferencijas, užsienio mokslininkų seminarus, mokslininkų susitikimus, akademinis skaitymus, parodas. Tikime, kad dalyvavimas šioje LMA kuruojamoje konferencijoje paskatins magistrantus ir kitus studentus tęsti mokslinę veiklą ir pabaigus studijas.

Konferencijos darbuose spausdinti recenzuoti studentų parengti moksliniai straipsniai. Tai dažniausiai pirmosios mokslinės publikacijos, bet tikimės, kad ateityje virs į straipsnius prestižiniuose mokslo žurnaluose. To norėtume palinkėti konferencijos dalyviams.

Organizatoriai

dr. Jolita Bernatavičienė

prof. habil. dr. Gintautas Dzemyda

prof. dr. Olga Kurasova

prof. dr. Julius Žilinskas

Turinys

<i>Mantas Bacevičius</i> . IDS/IPS technologijomis grįsto mobiliųjų įrenginių atakų prevencijos metodo sukūrimas ir tyrimas.	5
<i>Dominykas Eirošius</i> . Keliamo turinio strategijos analizė socialinio tinklo „Facebook“ puslapiuose	15
<i>Grantas Gadliauskas, Andrius Kriščiūnas</i> . Machine learning algorithm application in trip planning.	25
<i>Volodymyr Kadzhaia, Aistis Raudys</i> . Distributed machine learning for IoT	35
<i>Lukas Kuzma, Igoris Belovas, Martynas Sabaliauskas</i> . Precalculated arrays-based algorithms for the calculation of the Riemann zeta-function.	44
<i>Vieslav Lapin</i> . Nefunkcinių reikalavimų kintamumas paslaugų architektūros stiliaus sistemose.	54
<i>Ilya Lasy, Virginijus Marcinkevičius</i> . Dialogue System Augmented with Commonsense Knowledge	67
<i>Arnas Nakrošis, Andrius Kriščiūnas</i> . 3D paviršiaus formavimo 2D vaizdų pagrindu algoritmų tyrimas	77
<i>Ieva Rizgelienė</i> . Įmonių bankroto prognozavimas naudojant gilųjį mokymą	84
<i>Aušra Šubonienė</i> . Pilvo aortos aneurizmos segmentavimas neuroniniais tinklais kompiuterinės tomografijos nuotraukose.	101
<i>Artūras Tarasenka</i> . Apgaulingi šablonai įtaigioje kompiuterijoje	115

IDS/IPS technologijomis grįsto mobiliųjų įrenginių atakų prevencijos metodo sukūrimas ir tyrimas

Mantas Bacevičius

Kauno Technologijos Universitetas, Informatikos fakultetas,
Studentų g. 50, Kaunas
mantas.bacevicius@ktu.edu

Santrauka. Spartus išmaniųjų įrenginių skaičiaus ir juose saugomų duomenų kiekio ir jautrumo augimas lemia taip pat augančias ir duomenų saugumo rizikas. Šias rizikas siekia sumažinti operacinių sistemų kūrėjai, periodiškai išleisdami saugos atnaujinimus, tačiau yra nustatyta, kad pavojingiausios aplikacijos gali būti įdiegiamos kartu su šiais operacinės sistemos ar saugos OTA (angl. *over-the-air*) atnaujinimais – apie 5% įrenginių gamintojų įdiegtų aplikacijų yra kenkėjiškos. Taip pat apsaugos priemonės yra taikomos ir šiems mobiliems įrenginiams pritaikytoms infrastruktūroms – elektroninėms aplikacijų parduotuvėms, tačiau 67% kenkėjiškų aplikacijų vartotojus pasiekia būtent per jas. Siekiant atliepti saugumo rizikų mažinimo poreikį, šiame darbe yra pasiūlytas naujas IDS/IPS technologijomis grįsto mobiliųjų įrenginių atakų prevencijos metodas, paruošta eksperimentinė metodo realizacija ir atliktas šio metodo tyrimas.

Raktiniai žodžiai: išmanusis įrenginys, Android, IDS/IPS sistema, kenkėjiškas srautas, apsaugos metodas.

1 Įvadas

Nuolat didėjant prie interneto prijungtų įrenginių skaičiui, auga ir kibernetinių atakų, nukreiptų prieš šiuos įrenginius, pavojus. Tarptautinių kibernetinės saugos specialistų teigimu, potencialiai pavojingos aplikacijos gali būti įdiegiamos kartu su operacinės sistemos ar saugos OTA atnaujinimais – apie 5% įrenginių gamintojų įdiegtų aplikacijų yra kenkėjiškos [1]. Taip pat populiariais atakos vektoriais tampa sparčiai rinkoje platinamos aplikacijos [2], kurios gali pasirodyti nežalingos [3] [4], tačiau vykdo duomenų vagystės, šnipinėjimo bei kitas veiklas. Tai kelia pavojų ne tik plačioms vartotojų grupėms bet ir medicinos [5], informacinių technologijų ir pramonės [6] verslo sektoriams. Tai vyksta todėl, kad neretai organizacijose yra priimtina BYOD (angl. *Bring your own device*) politika. Tai atveria kelius

kenkėjiškoms programoms plisti į organizacijos viduje esančius įrenginius ir tinklus. Kibernetiniai incidentai, orientuoti į mobiliuosius įrenginius kelia vis didesnį susirūpinimą mobiliųjų įrenginių ir juose saugomų duomenų saugumu, o kibernetinės apsaugos priemonėmis, kuria suteikia aplikacijų ar operacinių sistemų kūrėjai, negalima. Šiuo metu vis efektyvus mobiliųjų įrenginių tinklų apsaugojimas vis dar kelia daug iššūkių. Dėl šios priežasties būtina sukurti metodus, leidžiančius apsaugoti mobiliųjų išmaniųjų įrenginių tinklo traktą nuo tinklo bei mobiliųjų įrenginių pažeidžiamumą ir atakų [7]. Šiame darbe yra pateikiamas apsaugos nuo tinklo atakų prieš mobiliuosius įrenginius metodas. Šis metodas leidžia vykdyti tinklo srauto analizę mobiliajame įrenginyje, naudojant mažus sisteminių resursų kiekius, kurie mobiliuosiuose įrenginiuose yra riboti. Taip pat šiame darbe yra pateikiami metodo eksperimentinės realizacijos bandymų realiomis ir sintetinėmis sąlygomis rezultatai.

2. Atakų prevencijos metodo realizacija

2.1. Realizacijos architektūra

Prototipo realizacija yra paremta Android operacinei sistemai skirta aplikacija. Aplikacija sukuria izoliuotą aplinką naudojant *chroot* konteinerį. *Chroot* konteineryje yra įdiegiama pagalbinė *Linux* operacinė sistema. Dėl lengvo pritaikymo, prieinamumo ir reikiamos minimalios papildomos konfigūracijos, buvo pasirinkta *Arch Linux* distribucija. *Chroot* konteineryje įdiegus pagalbinę operacinę sistemą, yra diegiama pasirinktina IDS/IPS sistema. Prototipo realizacijos metu buvo pasirinkta IDS/IPS sistema – „*Suricata*“. Svarbu pažymėti, kad konteinerio kūrimo, pagalbinės operacinės sistemos diegimo, IDS/IPS sistemos diegimo ir naudojimo metu veiksmus vykdančiai aplikacijai turi būti suteiktos „administratoriaus“ teisės, dėl šių priežasčių: (1) *Chroot* konteinerio naudojami failai yra diegiami aplikacijos konteinerio viduje tam, kad pašalinus aplikaciją, būtų pašalinti ir konteinerio viduje veikiančios pagalbinės operacinės sistemos failai, taip atlaisvinant mobiliojo įrenginio išorinės atminties talpyklos resursus; (2) Pagalbinėje operacinėje sistemoje veikiančios IDS/IPS sistemos paleidimui reikalingos administratoriaus teisės. Šios teisės leidžia IDS/IPS sistemai prieiti prie tinklo trakto.

Konteinerio kūrimo, pagalbinės operacinės sistemos, IDS/IPS sistemos diegimo veiksmams atlikti aplikacija naudoja virtualų terminalą.

2.2. Tinklo srauto perėmimo ir persiuntimo IDS/IPS procesams realizacija

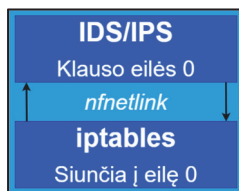
Tinklo srauto valdymui yra panaudotas pagrindinėje operacinės sistemos branduolyje veikiantis tinklo srauto maršrutizavimo karkasas *Netfilter*. Šis karkasas vykdo tinklo srauto maršrutizavimo funkcijas bei leidžia programiškai prisijungti ir stebėti su tinklo srautu susijusius įvykius. Šio karkaso veikimo koregavimui atlikti yra naudojamas pagalbinės operacinės sistemos vartotojo erdvėje veikianti tinklo srauto maršrutizavimo karkaso manipuliavimo programa *iptables*, leidžianti valdyti maršrutizavimo karkaso veiklą, naudojant konfigūruojamus taisyklių rinkinius, kurie yra taikomi taisyklėse nurodytiems tinklo srautams.

iptables palaiko taikinius *NFQUEUE*, kurie leidžia siųsti paketus į vartotojo erdvę, kur jie gali būti valdomi ir siunčiami atgal į operacinės sistemos branduolį, arba pašalinti. *NFQUEUE* eilėse saugomus paketus IDS/IPS sistemos gali perimti naudojant *Netlink* – duomenų schemomis orientuotą pranešimų siuntimo sistemą, leidžiančią perduoti pranešimus iš branduolio erdvės į vartotojo erdvę ir atvirkščiai. *Netlink* sąsaja perduodami tinklo paketai yra enkapsuliuojami į *Netlink* paketus, kurio struktūra yra pateikta 1 lentelėje.

1 lentelė *Netlink* paketo struktūra

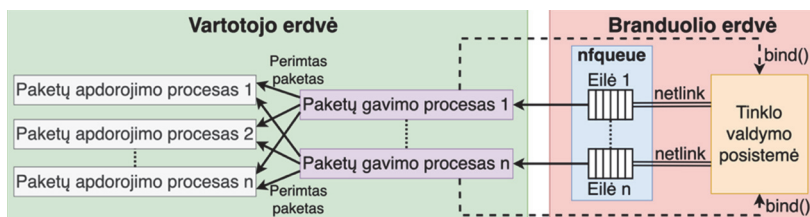
Bitų poslinkis	0-15	16-32
0	Paketo ilgis	
32	Tipas	Vėliavos
64	Sekos numeris	
96	Proceso identifikatorius (PID)	
128	Duomenys	

Netlink leidžia naudoti iki 32 ryšio magistralių operacinės sistemos branduolio erdvėje. Realizacijos atveju, ryšio magistralė yra naudojama vienos branduolio posistemės. Paketų perdavimo tarp IDS/IPS sistemos ir *iptables* schema yra pavaizduota 1 paveiksle.



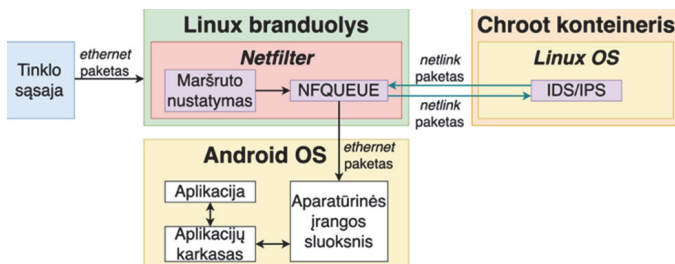
1 pav. Paketų perdavimo tarp IDS/IPS sistemos ir *iptables*

Tinklo srauto maršrutizavimo karkasas palaiko *Unicast* ir *Multicast* tipų komunikacijas, kurias palaiko ir IDS/IPS sistemos. Prototipo realizacijoje naudojama *Multicast* tipo komunikacija, kuri leidžia paketų gavimo procedūrų sudaromas apkrovas išskaidyti per keletą procesų, veikiančių skirtingose gijose. Priklausomai nuo konkrečios IDS/IPS sistemos realizacijos, paketai iš gavimo procesų gali būti perduodami asinchroniškai veikiantiems paketų apdorojimo procesams, kurių įprastai veikia bent du kart daugiau, nei yra veikiančių paketų gavimo procesų dėl didesnių skaičiavimo resursų kaštų. Paketų perdavimo, tarp vartotojo erdvės ir branduolio erdvės, procesų schema pateikta 2 paveiksle.



2 pav. Paketų perdavimas vartotojo erdvėje, naudojant *multicast* tipo komunikacijas

Pasitelkus *Netlink* sąsajos teikiamą tinklo maršrutizavimą į *NFQUEUE* eiles, iš kurių IDS/IPS sistemos perima paketus iš branduolio erdvės bei paketų grąžinimą įrenginio branduoliui tolimesniam Android OS apdorojimui, buvo pasiektas eksperimentinės realizacijos architektūrinis išdėstymas, pateiktas 3 paveiksle.

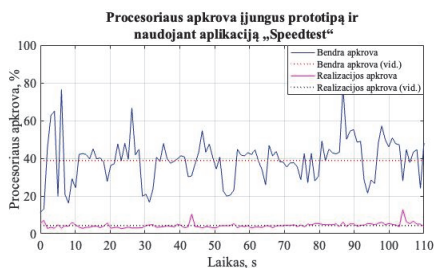


3 pav. IDS/IPS veikimo mobiliajame įrenginyje realizacijos architektūra

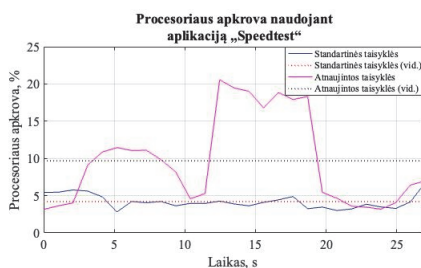
3. Tyrimai

3.1. Eksperimentinės realizacijos naudojamumo tyrimas

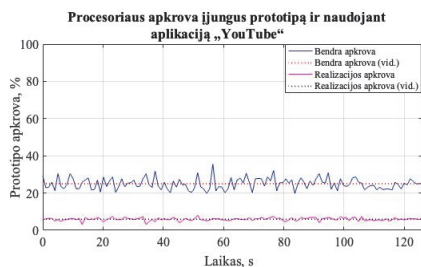
Eksperimentinės realizacijos naudojamumo tyrimui buvo atlikti keturi bandymai, kurių metu naudotos aplikacijos „YouTube“ ir „Speedtest“, siekiant atkurti įprastą mobiliojo įrenginio naudojamą, paremtą skirtingais tinklo naudojimo profiliais. „YouTube“ aplikacija buvo pasirinkta dėl populiarus aplikacijos naudojimo bei gebos kompensuoti tinklo pralaidumo nestabilumus, pasitelkiant buferizavimą. Aplikacija „Speedtest“ buvo pasirinkta dėl maksimalaus tinklo pralaidumo ir sisteminių resursų išnaudojimo. Tyrimo metu buvo atlikti atskiri bandymai, siekiant įvertinti eksperimentinės realizacijos generuojamos procesoriaus apkrovos ir realizacijoje naudojamų taisyklių kiekio poveikį naudojamumui. Bandymų rezultatai yra pateikti 4 paveiksle. Taip pat, tyrimų metu nustatytų metrikų paaiškinimai pateikti 2 lentelėje.



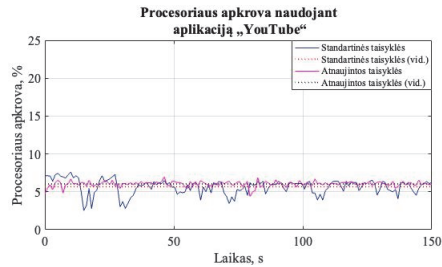
(a)



(b)



(c)



(d)

4 pav. Eksperimentinės realizacijos sugeneruojama sisteminių resursų apkrova, kartu naudojant ir aplikaciją „Speedtest“ (a), naudojant standartinės ir atnaujintas taisykles (b) bei naudojant aplikaciją „Youtube“ (c), eksperimentinėje realizacijoje naudojant standartinės ir atnaujintas taisykles (d)

2 lentelė. Tyrimų metu nustatytų metrikų paaiškinimai

Eil. Nr.	Žymėjimas	Paaiškinimas
1	$\overline{\mu_{e_a}}$	Realizacijos vidutinė procesoriaus apkrova,
2	β_a	Procesoriaus resursų vidutinis likutis
3	$\max \mu_{b_a}$	Didžiausia fiksuota bendra procesoriaus apkrova
4	$\max \mu_{e_a}$	Didžiausia fiksuota realizacijos generuota procesoriaus apkrova
5	σ_{b_a}	Bendros procesoriaus apkrovos standartinis nuokrypis
6	$\overline{\mu_{s_b}}$	Realizacijos, veikiančios su standartinėmis taisyklėmis, generuota vidutinė procesoriaus apkrova
7	$\overline{\mu_{a_b}}$	Realizacijos, veikiančios su atnaujintomis taisyklėmis, generuota vidutinė procesoriaus apkrova
8	$\max \mu_{a_b}$	Didžiausia realizacijos generuota procesoriaus apkrova
9	σ_{a_b}	Procesoriaus apkrovos standartinis nuokrypis, naudojant realizaciją su atnaujintomis taisyklėmis
10	$\overline{\mu_{e_c}}$	Realizacijos vidutinė procesoriaus apkrova,
11	$\max \mu_{b_c}$	Didžiausia fiksuota bendra procesoriaus apkrova
12	$\overline{\mu_{b_c}}$	Vidutinė bendroji procesoriaus apkrova
13	σ_c	Bendrosios apkrovos standartinis nuokrypis
14	$\overline{\mu_{s_d}}$	Realizacijos, veikiančios su standartinėmis taisyklėmis, generuota vidutinė procesoriaus apkrova
15	$\overline{\mu_{a_d}}$	Realizacijos, veikiančios su atnaujintomis taisyklėmis, generuota vidutinė procesoriaus apkrova
16	$\Delta\mu_{s_d a_d}$	Realizacijos veikimo su standartinėmis ir atnaujintomis taisyklėmis, sistemos apkrovos skirtumas
17	σ_{s_d}	Bendrosios apkrovos standartinis nuokrypis, naudojant realizaciją su standartinėmis taisyklėmis
18	σ_{a_d}	Bendrosios apkrovos standartinis nuokrypis, naudojant realizaciją su atnaujintomis taisyklėmis

Naudojant aplikaciją „Speedtest“, eksperimentinės realizacijos prototipas naudoja $\overline{\mu_{e_a}} = 4,4\%$ procesoriaus resursų (4 pav. a). Taip pat lieka $\beta_a = 95,6\%$ skaičiavimo resursų kitiems sistemos procesams vykdyti. Be to, šio bandymo metu fiksuota didžiausia bendra sistemos apkrova $\max \mu_{b_a} = 77,7\%$, nors maksimali realizacijos apkrova siekė $\max \mu_{e_a} = 12,7\%$. Šis pokytis tarp maksimalių bendrųjų apkrovų skirtumų ir maksimalios realizacijos apkrovos siejamas su IDS/IPS sistemos geba tinklo gavimo ir ap-

dorojimo procesus vykdyti lygiagrečiai. Buvo pastebėta, kad šio bandymo bendros apkrovos standartinis nuokrypis $\sigma_{b_a} = 11,9 \%$.

Iš 4 paveikslo (b) dalies galima matyti, kad naudojant aplikaciją „Speedtest“ su standartinių taisyklių rinkiniu veikianti IDS/IPS sistema „Suricata“ vidutiniškai generavo $\overline{\mu_{s_b}} = 4,2\%$ procesoriaus apkrovos, o su atnaujintų taisyklių rinkiniu $\overline{\mu_{a_b}} = 9,7\%$. Šių rodiklių skirtumas siejamas su didesniu atnaujintame rinkinyje esančių taisyklių skaičiumi. Dėl to grafike yra matomi keli procesoriaus apkrovos rodiklių šuoliai, kurie pasiekė $max\mu_{a_b} = 20,57 \%$ ribą, o rodiklių standartinis nuokrypis buvo $\sigma_{a_b} = 5,95 \%$. Dėl šių priežasčių galima teigti, kad IDS/IPS sistemai taikomų taisyklių skaičius turi tiesioginį poveikį tik esant didelei tinklo apkrovai.

Pagal rezultatus 4 paveikslo (c) dalyje, kuomet buvo naudota „YouTube“ eksperimentinė realizacija generavo vidutiniškai $\overline{\mu_{e_c}} = 6,2\%$ procesoriaus apkrovos. Fiksuota didžiausia bendra procesoriaus apkrova – $max\mu_{b_c} = 35,6\%$, o vidutinė bendroji procesoriaus apkrova – $\overline{\mu_{b_c}} = 25,0\%$. Taip pat verta pabrėžti, kad bendrosios apkrovos standartinis nuokrypis buvo lygus $\sigma_c = 3,0 \%$, todėl galima teigti, kad eksperimentinė realizacija nesukelia didelių sisteminių resursų apkrovos šuolių, kurie turėtų įtakos sistemos naudojamumui.

Su standartinių taisyklių rinkiniu veikianti eksperimentinė realizacija vidutiniškai generavo $\overline{\mu_{s_d}} = 5,6\%$ procesoriaus apkrovos, o su atnaujintų taisyklių rinkiniu $\overline{\mu_{a_d}} = 6,0\%$ (4 pav. d). Tai yra $\Delta\mu_{s_d a_d} = 0,4\%$ skirtumas, kuris gali būti siejamas su didesniu taisyklių skaičiumi atnaujintame rinkinyje. Eksperimentinės realizacijos veikimo su standartinėmis taisyklėmis procesoriaus apkrovos rodiklių gautas standartinis nuokrypis $\sigma_{s_d} = 0,99 \%$, o su atnaujintomis taisyklėmis $\sigma_{a_d} = 0,36 \%$.

3.2. Eksperimentinės realizacijos gebos fiksuoti kenkėjišką tinklo srautą tyrimas

Eksperimentinės realizacijos gebos fiksuoti kenkėjišką tinklo srautą tyrimui atlikti buvo naudota IDS/IPS sistema „Suricata“, veikianti „pakartojimo“ režimu bei du taisyklių rinkiniai. Standartinės taisyklės – taisyklių rinkinys, kuris įdiegiamas kartu su IDS/IPS sistema ir atnaujintos taisyklės – kurios yra įdiegiamos „suricata-update“ [8] įrankio pagalba. Abu taisyklių rinkiniai buvo bandyti su tinklo srauto įrašais, saugomais „*.pcap“ failuose, kuriuose yra išsaugotas kenkėjiškų programų tinklo srautas, klasifikuotas pagal kenkėjišką programą ir kenkėjiškos programos šeimą [9]. Eksperimentinės realiza-

cijos veikla buvo bandyta su $p = 42$ kenkėjiško tinklo srauto įrašais, kuriuos sudarė $n = 3636640$ tinklo paketų (žr. 3 lentelėje).

3 lentelė. Kenkėjiško tinklo srauto detekcijos, naudojant standartinės ir atnaujintas taisykles

Eil. Nr., i	Kenkėjiškos programos pavadinimas	Paketų skaičius	Kenkėjiškos programos šeima	Detekcijų skaičius		Atmestų paketų dalis, μ_i	
				Atnaujintos taisyklės	Standartinės taisyklės	Atnaujintos taisyklės	Standartinės taisyklės
1	Dowgin	113005	Adware	1192	25	80,69%	6,52%
...
4	Gooligan	406956		485	0	67,82%	4,15%
...
10	Youmi	99524		1320	57	79,65%	7,21%
11	Charger	86855	Ransomware	1470	23	48,09%	4,13%
...
15	Pletor	59448		675	0	63,69%	0,22%
...
20	WannaLocker	82619		1425	61	50,90%	2,77%
21	Android-Defender	98932	Scareware	2624	1925	62,27%	5,10%
...
31	VirusShield	154540		974	60	54,47%	0,92%
32	Beanbot	18757	SMSMalware	328	8	60,45%	6,85%
...
35	fakemart	10688		184	0	37,57%	1,02%
...
42	zsone	25022		512	25	42,13%	1,35%

Naudojant standartinės „Suricata“ IDS/IPS sistemos taisykles, „Gooligan“, „Pletor“ ir „fakemart“ kenkėjiškų programų, kurios priklauso 3-ims iš 4-rių bandytų kenkėjiškų programų šeimų, tinklo srautas nebuvo fiksuojamas kaip žalingas ir maža dalis tinklo srauto buvo atmesta (3 lentelė). Naudojant atnaujintas taisykles, visų bandytų kenkėjiškų programų tinklo srautas buvo fiksuotas kaip žalingus ir sugeneravo $a = 44577$ kenkėjiško tinklo srauto perspėjimo pranešimų. Tai yra $\Delta pr = 1429\%$ daugiau kenkėjiško tinklo srauto perspėjimo pranešimų, palyginus su IDS/IPS sistema, veikiančia

su standartinėmis taisyklėmis. Dėl šių priežasčių galima teigti, kad su IDS/IPS sistema „Suricata“ diegiamų standartinių taisyklių nepakanka pilnavertiškai kenkėjiškų programų generuojamo tinklo srauto detekcijai mobiliajame įrenginyje. Svarbu pabrėžti, kad IDS/IPS sistemai veikiant su atnaujintu taisyklių rinkiniu, ji atmetė $\overline{\mu}_\alpha = 51,48\%$ visų tinklo srauto įrašuose buvusių paketų.

Išvados

1. Atlikus literatūros analizę buvo nustatyta, kad pavojingiausios aplikacijos gali būti įdiegiamos kartu su operacinės sistemos ar saugos OTA atnaujinimais – apie 5% įrenginių gamintojų įdiegtų aplikacijų yra kenkėjiškos. Tai kelia pavojų ne tik plačioms vartotojų grupėms, bet ir medicinos, informacinių technologijų ir pramonės verslo sektoriams.
2. Šiame tyrime pasiūlytas naujas metodas, skirtas mobiliajame įrenginyje sustabdyti tinklo atakas. Šis metodas naudoja izoliuotą pagalbinę operacinę sistemą ir leidžia diegti bei vykdyti IDS/IPS sistemos procesus, nepriklausomai nuo įrenginio architektūros. Pagal pasiūlytą metodą paruošta eksperimentinė realizacija, gebanti izoliuotame konteineryje įdiegti *Unix* šeimos pagalbinę operacinę sistemą, IDS/IPS sistemą ir ją valdyti.
3. Nustatyta, kad standartinių, IDS/IPS sistemoje diegiamų, tinklo srauto patikros taisyklių rinkinių nepakanka, pilnavertiškai IDS/IPS sistemos veiklai ir apsaugai nuo kenkėjiško tinklo srauto. Pastebėta, kad eksperimentinės realizacijos veikimas neturi žymios įtakos įrenginio naudojamumui. Tyrimų metu buvo nustatyta, kad naudojant atnaujintas taisykles, realiomis sąlygomis, naudojamų resursų kiekis išaugo $\Delta\mu_{s_{dad}} = 0,4\%$.
4. Testavimui panaudoti $p = 42$ kenkėjiško tinklo srauto įrašai, kuriuos sudarė $n = 3636640$ tinklo paketų. Eksperimentiniai rezultatai parodė, kad $\overline{\mu}_\alpha = 51,48\%$ paketų fiksavo kaip žalingus ir juos atmetė. Apie kenkėjišką tinklo srautą buvo sugeneruoti $a = 44577$ perspėjamieji pranešimai.

Literatūra

- [1] J. Gamba, M. Rashed, A. Razaghpanah, J. Tapiador and N. Vallina-Rodriguez, „An Analysis of Pre-installed Android Software“, *2020 IEEE Symposium on Security and Privacy (SP)*, 2020, pp. 1039-1055, doi: 10.1109/SP40000.2020.00013.
- [2] P. Kotzias, J. Caballero and L. Bilge, „How Did That Get In My Phone? Unwanted App Distribution on Android Devices“, *2021 IEEE Symposium on Security and Privacy (SP)*, 2021, pp. 53-69, doi: 10.1109/SP40001.2021.00041.

- [3] B. Acohido, „Mobile threats are everywhere – here’s what you can do“, 2019, <https://blog.avast.com/mobile-device-cyberattacks>.
- [4] CrowdStrike, „A COMPREHENSIVE REVIEW OF MOBILE MALWARE TRENDS,“ 2019, https://go.crowdstrike.com/rs/281-OBQ-266/images/Report2019MobileThreatLandscape.pdf?lb_email=noEmail@noemail.com&utm_source=Marketo&utm_medium=Web&utm_campaign=Threat_Landscape_Mobile_Malware_2019.
- [5] B. Gyunka, O. C. Abikoye, „THE IMPACT OF ANDROID MALWARE ON MOBILE HEALTH APPLICATIONS (mHealth Apps) SERVICES“, *1st International Conference of the IEEE Nigeria Computer Chapter 2016 (IEEE Nigeria ComputConf'16)*, 2016.
- [6] Cybersecurity & Infrastructure Security Agency, „ICS Advisory (ICSA-18-256-01)“, 2018, <https://us-cert.cisa.gov/ics/advisories/ICSA-18-256-01>.
- [7] X. Su, M. Chauah, G. Tan, „Smartphone Dual Defense Protection Framework: Detecting Malicious Application in Android Markets“, *2012 8th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, 2012, pp. 153-160, doi: 10.1109/MSN.2012.43
- [8] Open Information Security Foundation, „Github,“ <https://github.com/OISF/suricata-update>.
- [9] A. H. Lashkari, A. F. A. Kadir, L. Taheri, A. A. Ghorbani, „Toward Developing a Systematic Approach to Generate Benchmark Android Malware Datasets and Classification“, *2018 International Carnahan Conference on Security Technology (ICCST)*, 2018, pp. 1-7, doi: 10.1109/CCST.2018.8585560.

Keliamo turinio strategijos analizė socialinio tinklo „Facebook“ puslapiuose

Dominykas Eirošius

Vilniaus universitetas, Matematikos ir informatikos fakultetas,
Didlaukio g. 47, LT-08303 Vilnius
dominykas.eirosius@mif.stud.vu.lt

Santrauka. Straipsnyje yra pristatoma socialinio tinklo „Facebook“ puslapiuose keliamo turinio strategijos analizė. Turinio strategija apibūdinama kaip turinio publikavimo planavimas, kūrimas pasirinktomis tematikomis bei kitoks turinio valdymas. Ši analizė buvo atlikta analizuojant tris „Facebook“ puslapius ir naudojantis latentiniu Dirichlė paskirstymo algoritmu, skirtu temų modeliavimui. Temų modeliavimas padeda analizuoti ir geriau suprasti nestruktūrizuotų tekstų rinkinius, kurie savyje slepia daugybę naudingos informacijos. Tyrimo metu, iš beveik 9 tūkst. vaizdo įrašų pavadinimų buvo sudarytos 10 temų, kurios leido detaliau pažvelgti į keliamą turinį. Išanalizavus duomenis buvo atskleistos keliamo turinio strategijos bei jų pasikeitimai nagrinėjamu laikotarpiu.

Raktiniai žodžiai: socialiniai tinklai; „Facebook“; latentinis Dirichlė paskirstymas; temų modeliavimas.

1 Įvadas

Temų modeliavimas yra vienas iš galingiausių teksto apdorojimo metodų, skirtų latentiniams duomenims aptikti bei tekstinių dokumentų ir kitų duomenų ryšiams rasti. Mokslininkai yra paskelbę daugybę įvairių temų modeliavimo straipsnių įvairiausiose srityse [1], [2], [3], [4], [5]. Šiam uždaviniui atlikti yra įvairių metodų, tačiau latentinis Dirichlė paskirstymas (angl. Latent Dirichlet allocation (LDA)) yra vienas populiariausių naudojamų algoritmų, leidžiančių išskirti iš žodžių rinkinių pagrindines temas be papildomos žmogaus intervencijos.

LDA galime įvardinti kaip metodą, kuris automatiškai suformuoja temas, kurioms priklauso tam tikri žodžiai. Kitaip tariant, tai yra temų, kurios sudarytos iš žodžių su tam tikromis tikimybėmis, rinkinys. Taigi, įvardinto metodo tikslas – iš žodžių ir dokumentų sukurti tiesiogiai nestebimą (latentinę) temą. Iš struktūrizuoto tekstų rinkinio (angl. corpus) LDA pirmiausia atkuria esančius dokumentus. To veikimas yra paremtas iš žodžių ir dokumentų

sudarytų temų svarba [6]. Prieš nurodant LDA specifikacijas, apibrėžkime sąvokas:

- Pagrindinis diskrečių duomenų vienetas yra žodis $\{1, \dots, V\}$, kuris įvardinamas kaip svarbiausias reikšminis kalbos vienetas. Žodžiai yra vaizduojami kaip vienetiniai vektoriai, o v -asis žodis yra vaizduojamas vektoriumi w , kur $w^v = 1$ ir $w^u = 0$, kai $v \neq u$.
- Dokumentas apibrėžiamas kaip N žodžių seka $\mathbf{w} = (w_1, w_2, \dots, w_N)$, kur w_n yra n -tasis sekos žodis.
- Struktūrizuotas teksto rinkinys (angl. corpus) yra M dokumentų kolekcija, žymima $D = \{w_1, w_2, \dots, w_M\}$.

LDA generavimo proceso specifikacijas kiekvienam dokumentui \mathbf{w} , esančiame struktūrizuotame teksto rinkinyje D galime užrašyti taip:

1. $N \sim \text{Poisson}(\xi)$
2. $\theta \sim \text{Dir}(\alpha)$ – dokumento temos skirstinys.
3. Kiekvienam žodžiui w_n iš N :
 - 3.1. Pasirenkama tema $z_n \sim \text{Multinomial}(\theta)$.
 - 3.2. Pasirenkamas žodis w_n iš $p(w_n | z_n, \beta)$ – tikimybės, sąlygotos temos z_n .

Taip pat modelyje yra daromos ir kelios prielaidos. Dirichlė skirstinio dimensija k ir temos kintamojo dimensija z yra fiksuoti ir žinomi. Žodžių tikimybės apibrėžiamos kaip $k \times V$ matrica β , kur $\beta_{ij} = p(w^j = 1 | z^i = 1)$ yra laikomu fiksuotu kiekiu. Tuo tarpu N nepriklauso nuo visų duomenis generuojančių kintamųjų, tokių kaip θ ir z .

θ yra k -matis Dirichlė atsitiktinis kintamasis, kuris įgyja reikšmes $(k - 1)$ -simplekse (k - vektorius θ įgyja reikšmes $(k - 1)$ -simplekse, jei $\theta_i \geq 0$, $\sum_{i=1}^k \theta_i = 1$, ir turi tikimybių tankį, lygų:

$$p(\theta | \alpha) = \frac{\Gamma(\sum_{i=1}^k \alpha_i)}{\prod_{i=1}^k \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \dots \theta_k^{\alpha_k - 1},$$

kur parametras α yra k -vektorius, kurio komponentai $\alpha_i > 0$, o $\Gamma(x)$ yra gama funkcija. Turėdami α ir β parametrus, bendras θ , temų rinkinio \mathbf{z} ir \mathbf{w} paskirstymas išreiškiamas:

$$p(\theta, \mathbf{z}, \mathbf{w} | \alpha, \beta) = p(\theta | \alpha) \prod_{n=1}^N p(z_n | \theta) p(w_n | z_n, \beta),$$

kur $p(z_n | \theta)$ yra θ_i unikaliam i , kad $z_n^i = 1$. Integruodami θ ir sumuodami z , gauname ribinį dokumento skirstinį:

$$p(\mathbf{w}|\alpha, \beta) = \int p(\theta|\alpha) \left(\prod_{n=1}^N \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \beta) \right) d\theta.$$

Galiausiai, paėmę atskirų dokumentų ribinių tikimybių sandaugą, gauname struktūrizuoto teksto rinkinio tikimybę:

$$p(D|\alpha, \beta) = \prod_{d=1}^M \int p(\theta_d|\alpha) \left(\prod_{d=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta) p(w_{dn}|z_{dn}, \beta) \right) d\theta_d.$$

Šio tyrimo tikslas – pasinaudojant latentiniu Dirichlė paskirstymo algoritmu ištirti trijų, socialinio tinklo „Facebook“, puslapių keliamo turinio strategijas. „Facebook“ puslapis yra apibūdinamas kaip profilis, kurį gali susikurti įmonės, organizacijos ar individualūs asmenys. Šiuose puslapiuose gali būti skelbiama įvairi informacija, kuri yra prieinama ir matoma visiems socialinės platformos vartotojams [7]. Socialinių tinklų kontekste, turinys apima ne tik puslapyje rašomus tekstus, bet ir naudojamas nuotraukas bei vaizdo įrašus. Todėl turinio strategija apibūdinama kaip turinio publikavimo planavimas, kūrimas pasirinktomis tematikomis bei kitoks turinio valdymas [8]. Tyrimas buvo atliekamas trimis etapais: pirmiausia duomenys buvo paruošiami analizei: išvalytos trūkstamos reikšmės bei sukurti papildomi išvestiniai kintamieji. Po to atliekamas vaizdo įrašų temų modeliavimas. Šiame etape sutvarkomos tekstinio kintamojo reikmės, atlikta tokenizacija ir žodžių redukavimo procesas iki žodžių kamienų. Sekančiu žingsniu buvo pritaikytas latentinio Dirichlė paskirstymo algoritmas ir susisteminamos keliamo turinio tematikos. Galiausiai, buvo atliekama „Facebook“ keliamo turinio skirtingais pjūviais statistinė analizė bei pateikiamos gautos išvados.

2 Duomenų aprašas

Tyrimo metu buvo naudojami duomenys, ištraukti iš „Tubular“ platformos [9]. Ši platforma yra skirta vaizdo įrašų parametrų sekimui ir jų analizėms. Naudojama duomenų imtis yra subalansuota, kurią sudaro 14 kintamųjų: 4 kokybiniai bei 10 kiekybiniai, ir 8792 skirtingų vaizdo įrašų iš 3 skirtingų „Facebook“ puslapių. Kintamieji apibūdina vaizdo įrašų pagrindinę informaciją: universalų identifikacijos numerį, puslapio ir vaizdo įrašo pavadinimą, paskelbimo laiką, įrašo trukmę, peržiūrų skaičių bei auditorijos skirtingų interakcijų kiekį.

Pradėjus analizuoti duomenis buvo pastebėtas papildomų išvestinių kintamųjų poreikis, be kurių būtų sudėtinga atlikti analizę. Vienas iš trūkstatų kintamųjų - bendras vaizdo įrašo surinktų reakcijų skaičius. 2016 m. „Facebook“ įvykdė pakeitimus savo platformoje ir išleido naują ypatybę „Reakcijas“ [10]. Ši funkcija yra esamos funkcijos paspausti mėgti ant vaizdo įrašo papildymas: vartotojai dabar gali pasirinkti vieną iš penkių papildomų animuotų jaustukų, kuriais gali išreikšti savo požiūrį į pamatytą vaizdo įrašą. Taigi, dabar egzistuoja šešios tokios reakcijos: „Patinka“, kurį jau yra žinoma, taip pat „Myliu“, „Haha“, „Vau“, „Liūdna“ ir „Pikta“ emocijos. Vis dėlto, pradinuose duomenyse yra pateikiamas tik kintamasis, kuris parodo, kiek žmonių paspaudė mėgti mygtuką. Trūkstant duomenų apie kitas reakcijas negalima tiksliai įvertinti, kaip „Facebook“ vartotojai reagavo į vaizdo įrašus ir kiek susidomėjimo įrašas sulaukė. Šiai problemai spręsti buvo pasitelktas atliktas tyrimas, kurio metu buvo nustatytas vidutinis reakcijų pasiskirstymas vaizdo įrašuose [11]. Buvo nustatyta, jog mėgti reakcija sudaro 77,55 proc. visų reakcijų, paliekant 22,45 proc. visoms kitoms reakcijoms. Remiantis šio tyrimo rezultatais ir proporcijos lygtimis, buvo sukurtas naujas kintamasis, parodantis bendrą vaizdo įrašo surinktų reakcijų skaičių.

Dar vienas kintamasis, kuris buvo sukurtas siekiant analizuoti bendrą vartotojų interakciją su vaizdo įrašų, yra bendras visų interakcijų skaičius. Socialinių tinklų kontekste, interakcija yra bet koks vartotojo atliktas veiksmas, susijęs su vaizdo įrašu. Taigi, šis kintamasis buvo sudarytas sudedant vaizdo įrašo komentarus, pasidalinimus ir bendrą reakcijų skaičių. Taip pat buvo sukurti išvestiniai kintamieji, kurie padėjo atlikti analizę ir parodė vidutinę vaizdo įrašo trukmę minutėmis, įrašo ilgio kategoriją bei kurį metų ketvirtį, savaitės dieną bei valandą buvo publikuotas vaizdo įrašas.

Galiausiai buvo nustatyta, jog duomenyse yra trūkstatų reikšmių. Skirtingi kintamieji turėjo skirtingą kiekį trūkstatų reikšmių, tačiau jos nesudarė daugiau kaip 3,9 proc. visos imties. Kadangi trūkstama informacija sudarė sąlyginai nedidelę dalį visų duomenų, nepilni įrašai buvo panaikinti.

3 Eksperimentinis tyrimas

Siekiant įvertinti puslapiuose keliamo turinio strategiją ir nustatyti jos pasikeitimus laikui bėgant, pirmiausia siekta nustatyti, koks yra keliamas turinys. Dėl šios priežasties buvo atliekama keliamų įrašų temų analizė. Temų sukūrimui ir raktinių žodžių generavimui buvo pasitelkti vaizdo įrašų pavadinimų duomenys.

Pirmiausiai, buvo atliktas tokenizavimas. Tokenizavimas yra eilučių sekos suskaidymas į dalis, tokias kaip žodžiai, raktiniai žodžiai, frazės, simboliai ir kiti elementai, vadinami žetonais (angl. Tokens). Žetonai gali būti atskiri žodžiai, frazės ar net ištisi sakiniai. Taigi, kiekvienas sakinyss buvo suskirstytas į žodžių sąrašą, visiškai pašalinti skyrybos ženklai ir nereikalingi simboliai. Atlikus tokenizaciją buvo atliktas kamienuų sudarymas. Tai yra žodžio redukavimo procesas iki žodžio kamieno, kitaip žinomo kaip lema, kuris pridedamas prie priesagų ir priešdėlių. Tokio proceso tikslas yra sumažinti bendrą žodžių skaičių žodyne ir palikti tik unikalius žodžius. Dėl kamienuų sudarymo temų modeliavimas turėtų pavykti geriau ir turėtų būti sukurtos tikslinės temos. Paskutiniame teksto apdorojimo etape buvo sukurta žodžių dokumento matricą (angl. Document-Word matrix) ir pašalinti angliški stabdomieji žodžiai (angl. stopwords), puslapio ir vaizdo įrašo pavadinimą, paskelbimo laiką, įrašo trukmę, peržiūrų skaičių bei auditorijos skirtingų interakcijų kieki.

Sukūrus žodžių dokumento matricą buvo pasinaudota latentinio Dirichlė paskirstymo modeliu. Natūralios kalbos apdorojimo (angl. Natural language processing) kontekste latentinis Dirichlet paskirstymas (angl. Latent Dirichlet allocation) yra statistinis modelis. Tai yra temų modelio algoritmas, kuris naudojamas tekstus priskirti tam tikroms temoms. Modelio veikimas yra paremtas prielaida, kad dokumentai yra sudaryti iš žodžių, kurie padeda nustatyti temas. Šia prielaida vadovaujantis yra bandoma susieti dokumentus su temų rinkiniu, priskiriant kiekvieną dokumento žodį skirtingoms temoms. Toks priskyrimo procesas yra atliekamas sąlyginių tikimybių įverčių pagalba. Taigi, įvertinus žodžiams tikimybes, yra sudaromas žodžių rinkinys, kuris atvaizduoja tam tikrą temą. Rinkinys yra sudaromas pasirenkant žodžius su didžiausiomis tikimybėmis arba nustatant tikimybinį slenkstį ir pasirenkant tik tuos žodžius, kurių tikimybė yra didesnė arba lygi pasirinktai slenkstinei vertei. Tyrime buvo pasirinkta surasti ir sudaryti 10 temų, nes nustatant kitokį kiekį modelio tikslumas mažėja. Pritaikius latentinio Dirichlė paskirstymo modelį buvo gauta temų žodžių matrica, kuri pateikiama 1 pav.

Iš temų žodžių matricos buvo sudarytos šios temos: „Handicrafts & Food“, „Makeover & Changeover“, „Construction & Water“, „Furniture & Construction Tools“, „Life hacks & Tricks“, „Makeover & Renovation“, „Makeup & Garden decor“, „Wood crafts & Art“, „Design & Decoration“, „Satisfaction & Repair“.

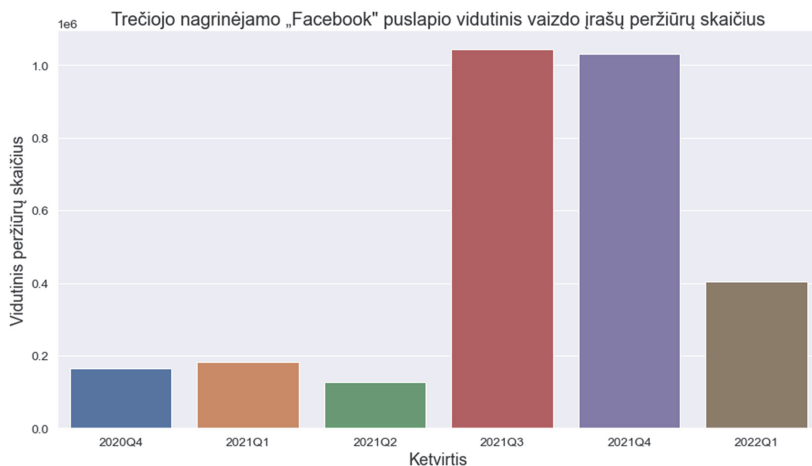
	Word 0	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7	Word 8	Word 9	Word 10	Word 11	Word 12	Word 13	Word 14
Topic 0	turn	knife	vase	bowl	food	bike	box	wood	steel	shell	coin	piece	building	cast	tank
Topic 1	diy	hair transformation	house	aquarium	brick	shape	axe	soap	concrete	pallet	use	stove	pen	hand	dress
Topic 2	build	transform	make	fashion	glass	backyard	log	water	rebar	wood	bicycle	project	waterfall	concrete	wheel
Topic 3	restoration	table	make	know	try	work	coffee	style	gun	drill	grill	hammer	dam	chain	toy
Topic 4	hack	tip	life	know	try	trick	car	look	learn	weld	fix	repair	leather	clothe	kitchen
Topic 5	make	use	epoxy	makeover	tool	nail	metal	wood	chair	recycle	resin	bend	cake	bolt	time
Topic 6	idea	craft	cement	technique	tip	makeup	hairstyle	shoe	decor	garden	need	wood	plant	welding	check
Topic 7	make	lamp	pipe	woodturne	wood	art	night	ring	cut	cardboard	science	experiment	candy	man	paper
Topic 8	tip	way	restore	candle	create	construction	egg	design	decoration	lock	wire	door	girl	gift	guy
Topic 9	home	make	process	satisfy	beauty	resin	gadget	paint	repair	cleaning	iron	problem	extension	grinder	tile

1 pav. Žodžių matrica.

4 Keliamo turinio strategijos analizė skirtingais pjūviais

Turinio strategija buvo nagrinėjama pagal: vidutinį vaizdo įrašo peržiūrų skaičių kiekvieną ketvirtį, kiekvieno ketvirčio savaitės dienomis Vaizdo įrašų publikavimo pasiskirstymą, vaizdo įrašų publikavimo laiką kiekvieną ketvirtį, vidutinį vaizdo įrašų peržiūrų skaičių pagal vaizdo įrašų trukmę, keliamų vaizdo įrašų kategorijų pasiskirstymą kiekvieną ketvirtį, vidutinį įrašų interakcijų kiekį kiekvieną ketvirtį, vidutinę vaizdo įrašų trukmę kiekvieną ketvirtį, kiekvieną ketvirtį įkeliamų vaizdo įrašų temų pasiskirstymą, vidutinį vaizdo įrašų peržiūrų skaičių pagal temas.

Ištyrus „Facebook“ konkurentų kanalus išsiaiškinta, jog vidutinis vaizdo įrašų peržiūrų skaičius kiekvieną ketvirtį pirmame kanale nebuvo pastovus, tačiau ženkliai nesikeitė. 2 pav. yra pateikiamos trečiojo nagrinėjamo „Facebook“ puslapio vidutinės vaizdo įrašo peržiūros. Taip pat pastebėta, jog antro ir trečio kanalų grafikai buvo labai panašūs. Įdomu tai, jog 2021 m. trečio ir ketvirto ketvirčio vidutinis peržiūrų skaičius šiuose kanaluose buvo nepaprastai didelis. Tokio teigiamo rezultato priežastys gali būti sėkmingos marketinginės kampanijos, labai kokybiškų ir sudominančių vaizdo įrašų sukūrimas.



2 pav. Trečiojo nagrinėjamo „Facebook“ puslapio vidutinis vaizdo įrašo peržiūrų skaičius.

Žvelgiant į kiekvieno ketvirčio savaitės dienų vaizdo įrašų publikavimo pasiskirstymą aiškiai matyti, jog pirmojo nagrinėjamo puslapio visų ketvirčių keliamų įrašų strategija buvo išlaikyta tokia pati. Visomis dienomis buvo stengiamasi kelti po panašų kiekį vaizdo įrašų. Antrasis nagrinėjamas puslapis 2020 m. ketvirtame ketvirtyje didesnę kiekį įrašų stengėsi kelti antradieniais, trečiadieniais ir ketvirtadieniais, o trečiasis – pusę keliamų įrašų publikuodavo ketvirtadieniais, likusią dalį po lygiai antradieniais ir sekmadieniais. Tačiau galima pastebėti, jog tokia strategija nepasitvirtino, nes likusiais ketvirčiais įrašai buvo keliami apytiksliai po lygiai kiekvieną dieną.

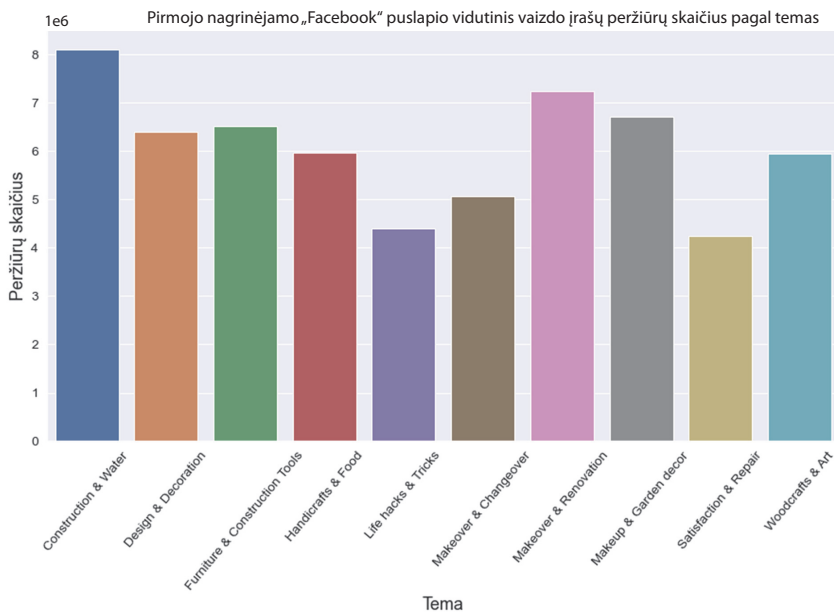
Analizuojant vidutinį vaizdo įrašų peržiūrų skaičių pagal vaizdo įrašų trukmę pastebėta, jog pirmuose dvejuose kanaluose didžiausio susidomėjimo sulaukė vidutinės trukmės vaizdo įrašai. Paskutiniojo puslapio lankytojams labiausiai patiko stebėti trumpesnius vaizdo įrašus. Svarbu paminėti tai, kad visuose kanaluose labiausiai dominuoja labai trumpi vaizdo įrašai. Vis dėlto galima daryti prielaidą, kad kanalų moderatoriai atsižvelgė į žiūrovų susidomėjimą ilgesniais įrašais, nes vėlesniuose ketvirčiuose buvo pradedami publikuoti ir ilgesnės trukmės filmukai.

Pradėjus nagrinėti kanalų vidutinį įrašų interakcijų kiekį iš karto pastebėtas pirmojo puslapio publikuojamų įrašų prastėjantis rezultatas. Matoma, kad auditorijos sąveikavimas su vaizdo įrašais palaipsniui mažėja. Tai gali indikuoti žiūrovų nepasitenkinimą turinio kokybe. Nors šiame kanale nėra stebimas vaizdo įrašų peržiūrų skaičiaus mažėjimas, interakcijų smukimas atskleidžia, jog ateityje šio kanalo populiarumas ir peržiūrų skaičius taip pat gali smukti. Tuo tarpu kitų puslapių pirmais trimis ketvirčiais sąveikų su vaizdo įrašais buvo labai mažai. Tačiau tokio rezultato galima buvo tikėtis, nes šiais ketvirčiais ir vaizdo įrašai atitinkamai sulaukdavo mažai peržiūrų.

Peržvelgus vidutinę publikuojamų vaizdo įrašų trukmę kiekvieną ketvirtį nepastebėta jokių stebinančių rezultatų. Visuose kanaluose vidutinė trukmė auga. Taip yra todėl, nes kaip jau buvo pasakyta, vėlesniuose ketvirčiuose kanalų savininkai atsižvelgė į faktą, jog didžiausio žiūrovų susidomėjimo sulaukia trumpi ir vidutinės trukmės vaizdo įrašai.

Paskutiniame etape buvo analizuojami duomenys pagal įkeliamų vaizdo įrašų temų pasiskirstymą ir vidutinį vaizdo įrašų peržiūrų skaičių pagal šias sudarytas temas. 3 pav. atskleidžia, jog pirmame kanale vaizdo įrašai keliami apytiksliai po lygiai visomis temomis. Vis dėlto, labiausiai žiūrimos temos buvo šios: „Construction & water“ bei „Makeover & revonation“. Nors antrame puslapyje su lyg kiekvienu ketvirčiu pradėjo pasirodyti vis įvairesnis turi-

nys, tačiau labiausiai publikuojantys vaizdo įrašai buvo „Makeup & Garden decor“ bei „Life hacks & Tricks“ temomis. Įdomu tai, jog kanalo savininkų daugiausiai keliamo turinio temos išsiskyrė su žiūrovams labiausiai patikusiomis temomis. Pasirodo, jog auditorijai labiausiai patiko žiūrėti „Satisfaction & repair“, „Construction & water“ bei „Design & decoration“. Tuo tarpu paskutiniame puslapyje pirmąjį ketvirtį dominavo tik „Satisfaction & repair“, „Design & decoration“ bei „Woodcrafts & art“ keliamų vaizdo įrašų tematikos. Vėliau turinys buvo publikuojamas įvairesnis. Vis dėlto lankytojų mėgstamiausios temos buvo: „Makeup & garden decor“, taip pat „Makeover & renovation“.



3 pav. Pirmojo nagrinėjamo „Facebook“ puslapio vidutinis vaizdo įrašų peržiūrų skaičius pagal temas.

5 Išvados

Tyrimo metu buvo ištirtos trijų, socialinio tinklo „Facebook“, puslapių keliamo turinio strategijos, nustatytas požymių rinkinys su papildomais išvestiniais parametrais, kuris leido įvertinti tiriamų strategijų reikšmingu-

mą. Strategijas sudarė tokie požymiai, kaip publikuojamo turinio laikas bei diena, kuriamų vaizdo įrašų trukmė ir turinio tematikos. Pasinaudojus latentiniu Dirichlet paskirstymo algoritmu buvo sudarytos 10 temų. Vėliau, šiomis temomis buvo pasinaudota analizuojant keliamą turinį. Pastebėta, jog 2021 m. trečias ir ketvirtas ketvirčiai buvo sėkmingi nagrinėjamiems puslapiams. Lyginant su kitais ketvirčiais, šiuo laikotarpiu vidutiniškai buvo surenkama ženkliai didesnis vaizdo įrašų peržiūrų kiekis. Taip pat buvo atskleista, jog stebimu laikotarpiu „Facebook“ puslapiai pradėjo kurti ir publikuoti ilgesnės trukmės vaizdo įrašus. Galiausiai pastebėta, jog puslapiuose pradėjo pasirodyti vis įvairesnis turinys, nors temų populiarumai ir žiūrovų susidomėjimas jomis skyrėsi.

Literatūra

- [1] McAuliffe, Jon, and David Blei. Supervised topic models. *Advances in neural information processing systems*, 20, 2007.
- [2] Li, D., He, B., Ding, Y., Tang, J., Sugimoto, C., Qin, Z., Yan, E., Li, J. and Dong, T. Community-based topic modeling for social tagging. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, 1565-1568, 2010.
- [3] Daud, A. Using time topic modeling for semantics-based dynamic research interest finding. *Knowledge-Based Systems*, 26: 154-163, 2012.
- [4] Lim, K.W., Chen, C. and Buntine, W. Twitter-network topic model: A full Bayesian treatment for social network and text modeling, arXiv preprint arXiv:1609.06791, 2016.
- [5] Weng, J., Lim, E.P., Jiang, J. and He, Q. TwitterRank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, 261-270, 2010.
- [6] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3: 993-1022, 2003.
- [7] Kelly Kennedy. Facebook Page vs Profile: Everything You Need To Know. <https://yourblogworks.com/facebook-page-vs-facebook-profile/>.
- [8] Kristina Halvorson. The Discipline of Content Strategy. <http://alistapart.com/article/thedisciplineofcontentstrategy/>.
- [9] Tubular. <https://tubularlabs.com/>.
- [10] Sammi Krug. Reactions Now Available Globally. <https://about.fb.com/news/2016/02/reactions-now-available-globally/>.
- [11] Olga Rabo. We Analyzed 2,810 Pages to Calculate Average Facebook Engagement Rate. <https://blog.iconosquare.com/average-facebook-engagement-rate/>.

Machine learning algorithm application in trip planning

Grantas Gadliauskas, Andrius Kriščiūnas

Kaunas University of Technology, Faculty of Informatics,
K. Donelaičio St. 73, 44249 Kaunas, Lithuania
grantas.gadliauskas@ktu.edu, andrius.krisciunas@ktu.lt

Abstract. This article explores how machine learning can be applied in efficiently solving a variation of the Travelling Salesman Problem (TSP) in the context of air travel tourism. Large number of cities create too many trip route combinations to be efficiently evaluated in real time. The method proposed uses a feed-forward neural network to narrow down the number of trip route combinations, while a more traditional algorithm based on dynamic programming is then able to select the best trip offers. It was shown that the method could be applied in practice to achieve almost real-time generation of best possible trip offers while evaluating a large amount of real-world flight data.

Keywords: travelling salesman problem, flight search, combinatorial optimization, neural network.

1 Introduction

Consider a tourist who wants to visit several different cities in a specific date range in a round trip from his home city. The tourist might also have preferences to which cities one wants to visit or avoid. A list of N best possible trip offers then should be provided to the user, based on the real-world flight data. The quality of the trip is determined by its price, but additional metrics could be added.

Since flight data updates very often and the number of possible date ranges is immensely huge it is not practical to pre-calculate all the offers. On the other hand, finding the best offers in real-time is inefficient due to the need to compute the best scored combination of flights for a large amount of possible trip routes.

In the combinatorial optimization domain, the more simplified version of this problem is well known as the Travelling Salesman Problem (TSP) [1]. More recent works on the topic also include machine learning approaches such as one by Chaitanya K. *et al.* [2] which makes use of neural networks to

perform TSP efficiently with hundreds of nodes. For our problem, however, the number of nodes (possible trip flights) will never be more than a few hundred, but the more important issue is the number of trip routes growing exponentially because of the number of different cities.

This article proposes a heuristic solution that allows to efficiently find the best trip offers using a feedforward neural network combined with a tree search algorithm based on dynamic programming (hereinafter DP). The feedforward neural network (hereinafter FNN) model can narrow down the total number of possible trip route combinations to a smaller amount of potential best trip candidates, while the algorithm based on dynamic programming is then able to select the N best trips. 3 scenarios with different constraints on the trip offer are tested.

2 Method

Each trip contains a set of cities T , $|T| = N$, with a particular order. A trip starts at the start city t_1 and the last city visited is denoted as t_n . Every trip is a round trip and ends at the start city t_1 . The ordered set T is referred to as a *trajectory*.

The trajectories used in our experiment are made up from 100 selected European cities. City selections are based on OpenStreetMap Place Importance Score (OSM PIS). Each city has a set of airports assigned to it, which is used to associate flight data with the city. The cities used are marked in Figure 1.

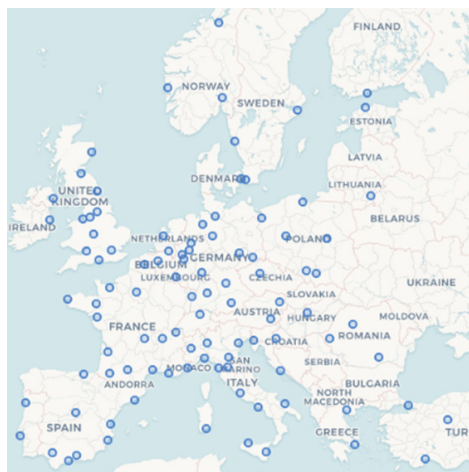


Figure 1. Selected cities

Each of the trajectories consist of 5 cities. Trajectories are generated based on the real-world flight routes which seldom change. Since there could be a total of $100 \cdot 99 \cdot 98 \cdot 97 \cdot 96$ trajectories, it would be impractical to consider all the possible combinations. Trajectory amount can be reduced by selecting only the most attractive trajectories – ones with the best trajectory round score and combined OSM PIS. Round score is calculated by dividing the total trajectory distance by the minimal possible distance connecting all the cities. Since only 5 cities make the trajectory, calculating the round score is trivial. 210000 trajectories were generated for our experiment. An example of a trajectory with its adjacent cities connected by a blue line is presented in Figure 2.

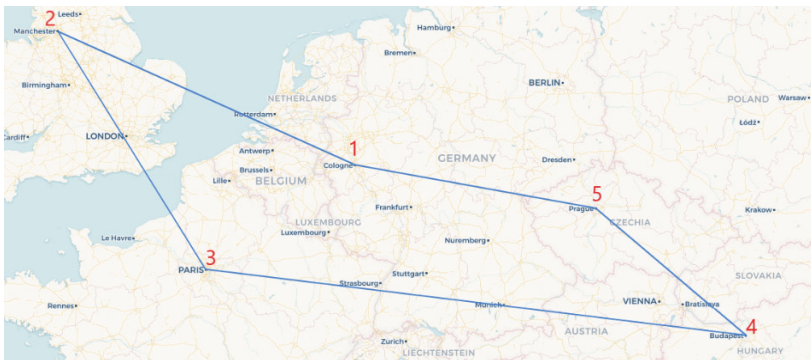


Figure 2. Trajectory graph example

The score of the trip equals to the sum of its flight ticket prices. The lower the score, the better the trip is considered.

Our method to find the best N trips given M possible trajectories and flight data is as follows:

1. Pass the flight prices of M trajectories of the given date range to the FNN model. Each day can have at most a single flight for a given route between two cities.
2. Pass some amount of best predicted trajectories and their flight prices to the DP algorithm.
3. Use the best N trajectories returned by the DP algorithm and the flight data to build the best N trip offers.

To determine if our method is viable in practice, we evaluate speed and accuracy metrics. Speed is measured as the combined computing time of FNN prediction and DP algorithm. Accuracy is determined by comparing the

final output of N best trajectories to expected N best trajectories and dividing the sum of matching pairs by N . Accuracy is influenced by the number of total trajectories passed to the FNN model and the number of best model predictions passed to the DP algorithm.

3 trip generation scenarios are explored in this research. They differ by the constraints applied to what can be considered a viable trip.

- Scenario 1: No additional constraints.
- Scenario 2: Each city has a minimum, maximum and preferred number of days to spend in that city. Trip generation must respect the minimum and maximum constraints and apply a penalty if the time spent in a city does not match its preferred number. The penalty subtracts a value of 5 from the trip score per absolute day offset.
- Scenario 3: Same as scenario 2, also, a total trip length constraint is added. The trip length must be in one of the three intervals:
 - o 10 – 13 days
 - o 14 – 20 days
 - o 21 – 24 days

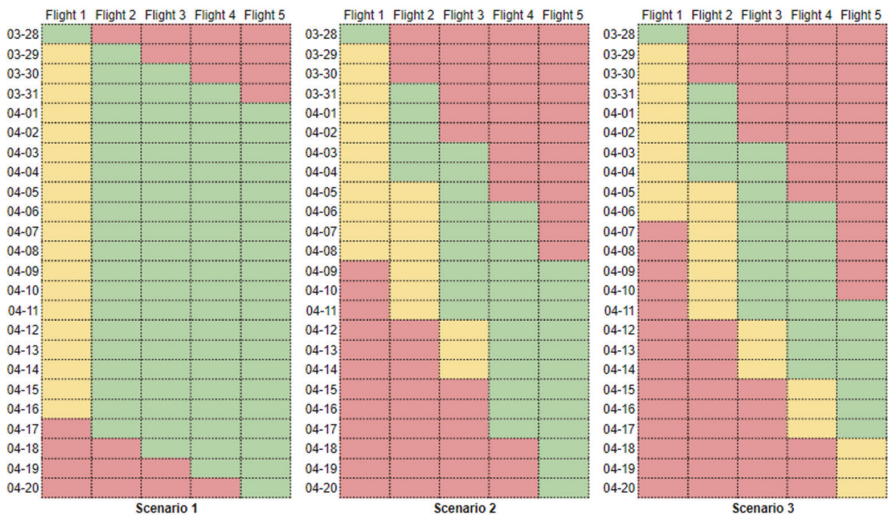


Figure 3. Testing scenarios

The scenarios are visualized in Figure 3. Testing scenarios, where the X axis represents the flights, and the Y axis represents the days. The red

cells mark the days in which it is impossible to take the flight to match the given constraints. The green cells represent the possible days to take flights if trip is starting from the earliest day (03-28). Yellow cells represent other possible days if trip were to start from another day. With each scenario, the amount of possible flight combinations is reduced. In the Figure 3. Testing scenarios example, the min. and max. number of days to stay in every city are 3 and 7 respectively and the total trip length should span from 14 to 20 days. In our experiment, for all scenarios we constrain the maximum trip length to 24 days.

3 Experiment

The computing environment used for this research has the following parameters:

- RAM memory: 32 GB
- CPU: AMD Ryzen 7 3700X 8-Core Processor 3.60 GHz (1 CPU core used)
- OS: Windows 10 64-bit architecture

Python 3.9 programming language is used to program the software. PyTorch machine learning framework is used to create the neural network model.

Common neural network training hyperparameters for each scenario are as follows:

- Batch size: 128
- Optimizer: Adam [3]
- Loss function: Mean absolute error (MAE)
- Learning rate: Reducing learning rate on plateau by a factor of 0.5 on 3 consecutive epochs without improvement.
- Epochs: Until does not improve for 10 epochs or until 60.

The departure dates for all the flight data in the experiment span between 2021-03-28 and 2021-05-15. Flight data from 2021-03-28 to 2021-04-20 is used for training, while data from 2021-04-21 to 2021-05-15 is used for validation. For scenario 1, flight prices of 24 days for each of the 5 trajectory cities are passed to the model as an input, for a total input length of 120. If the flight data for a particular day is missing, it is passed to the model as a value of -1. For scenario 2, the number of min., max. and preferred days for each city is added to the input, which increases the input length to 135.

For scenario 3, numbers for min. and max. trip length are added, for an input length of 137. The target trip scores for the model training were built using the DP algorithm. If not a single trip can be built for a trajectory under certain constraints, the target is set to a value of 1000. The model outputs a single value – a trip score. 210000 inputs were used to train model for scenarios 1 and 2, while scenario 3 trained with three times number of inputs (630000) due to three distinct intervals used for total trip length.

Neural network validation accuracy during training for each scenario is presented in figures Figure 4, Figure 5 and Figure 6. The used notation to describe the model architectures in the figure legends is as follows: $I \times H * N \times O$, where I is the number of inputs for the input layer, H is the number of inputs for each hidden layer, N is the number of hidden layers and O is the number of outputs (1 output describing the trip score). The best model architecture is highlighted with a yellow marker. In general, to obtain the optimal validation accuracy, the models had to become more complex as the trip constraints increased. Model for scenario 2 tends to overfit the most and the model state after 4th epoch is used for its metric check. Techniques such as dropout [4] and dataset scaling were tested but failed to improve the model accuracy.

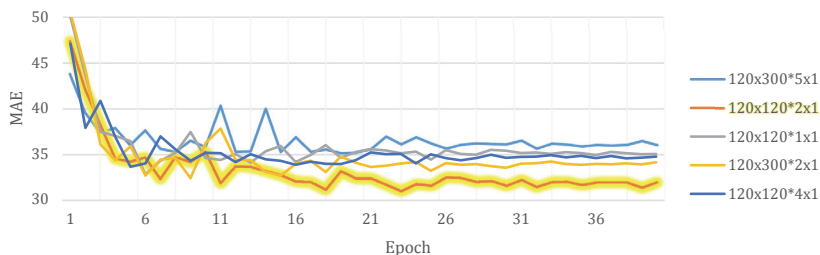


Figure 4. Scenario 1 validation accuracy

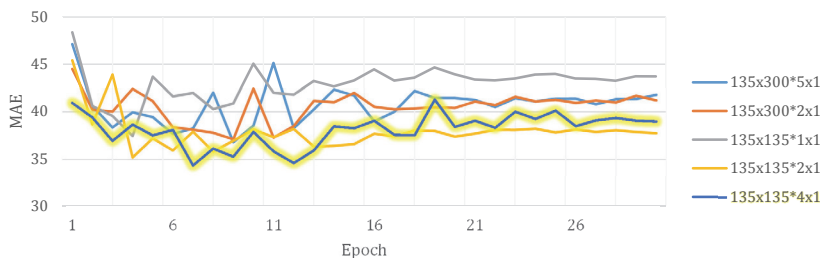


Figure 5. Scenario 2 model validation accuracy

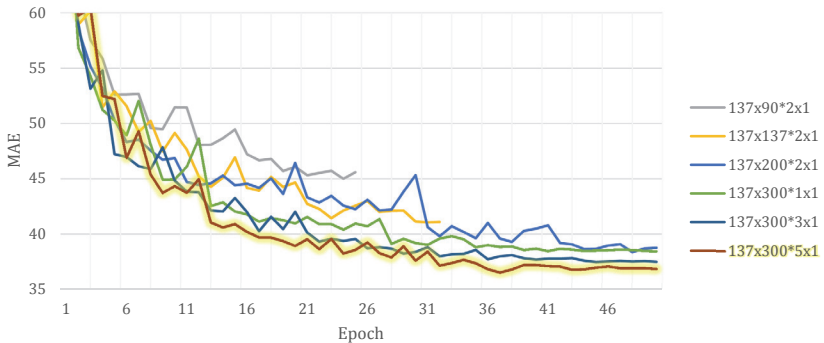


Figure 6. Scenario 3 model validation accuracy

The dynamic programming algorithm is used to find the actual best possible flight combination for a given trajectory. It works like a brute force tree search algorithm, but with optimizations. Instead of checking each possible flight combination, it stops traversing the flights if the flight for a given day was already traversed and had a better total flight price. It also only iterates through the dates which match the constraints of the min. and max. days to stay in a certain flight city and ignores days which do not match the total trip length constraint (such days are marked red in Figure 3).

4 Results

The final accuracy and speed results are presented in tables Table 1 and Table 2. Accuracy match results were averaged over 50 test runs. In the accuracy result table, cells marked in red, yellow, and green represent respectively the worst, the second best and the best scenario for the testing parameters of the rightmost 3 columns. Columns "Total predictions made", "Top N predictions to search in" and "Required top N matches" denote respectively how many inputs were passed to the FNN model, how many of the best results then were passed to the DP algorithm, and how many final trip offers do we want to output. The cell values in bold mark the values for which at least 80% of the required top N matches do match, which is considered a good result.

It is possible to infer from the accuracy results that the more constrained the trip generation scenario is, the more accurate the final matches tend to be. Since the mean absolute error of the FNN model validation accuracy was lower for the more constrained models, this might not seem reasonable.

Table 1. Accuracy results

Actual matches			Required top N matches	Top N predictions to search in	Total predictions made
1 Scen.	2 Scen.	3 Scen.			
5,48	6,72	8,72	10	50	2000
10,86	12,20	16,22	20	50	2000
23,04	23,84	31,78	50	50	2000
6,76	8,26	9,50	10	100	2000
13,92	15,38	18,42	20	100	2000
32,42	34,18	41,74	50	100	2000
8,96	9,82	9,82	10	250	2000
18,16	19,26	19,46	20	250	2000
43,62	44,76	47,58	50	250	2000
9,84	9,98	9,84	10	500	2000
19,70	19,94	19,56	20	500	2000
48,58	48,90	48,82	50	500	2000
3,60	3,92	6,08	10	50	10000
6,26	6,50	11,00	20	50	10000
11,32	12,32	21,28	50	50	10000
5,16	6,32	8,12	10	100	10000
9,22	11,06	15,34	20	100	10000
19,56	22,62	32,82	50	100	10000
7,00	8,02	9,32	10	250	10000
13,14	14,92	18,40	20	250	10000
29,28	34,88	43,92	50	250	10000
8,12	9,24	9,76	10	500	10000
15,46	17,44	19,44	20	500	10000
35,74	41,90	48,04	50	500	10000
3,48	1,24	3,92	10	50	50000
4,46	2,06	6,74	20	50	50000
5,80	4,24	12,12	50	50	50000
4,86	2,24	5,50	10	100	50000
6,70	3,44	9,94	20	100	50000
10,14	7,80	19,50	50	100	50000
6,60	6,12	8,34	10	250	50000
10,04	10,98	15,10	20	250	50000
17,10	21,58	31,74	50	250	50000
7,98	8,44	9,44	10	500	50000
12,70	15,54	17,50	20	500	50000
24,18	33,50	40,68	50	500	50000

Table 2. Speed results

Trajectories	Scenario 1		Scenario 2		Scenario 3	
	FNN time	DP time	FNN time	DP time	FNN time	DP time
50	0:00:00:001	0:00:00:022	0:00:00:001	0:00:00:017	0:00:00:002	0:00:00:013
100	0:00:00:001	0:00:00:045	0:00:00:001	0:00:00:035	0:00:00:004	0:00:00:023
250	0:00:00:001	0:00:00:109	0:00:00:003	0:00:00:093	0:00:00:009	0:00:00:065
500	0:00:00:002	0:00:00:227	0:00:00:004	0:00:00:183	0:00:00:018	0:00:00:123
2000	0:00:00:008	0:00:00:881	0:00:00:016	0:00:00:653	0:00:00:070	0:00:00:470
10000	0:00:00:041	0:00:04:351	0:00:00:080	0:00:03:141	0:00:00:338	0:00:02:305
50000	0:00:00:191	0:00:21:987	0:00:00:423	0:00:15:972	0:00:01:706	0:00:11:449

However, it may be explained by the greater value of the standard deviation of more constrained scenario model target array (trip scores) compared to less constrained scenario targets. The final matches tend to be less accurate the more trajectory inputs are passed to the FNN model and the fewer top predictions are ran through the DP algorithm.

The speed results show that the performance of the FNN is extremely quick running faster than half of a second for 50000 trajectory inputs in 1st and 2nd scenario and in 1.7 seconds in 3rd scenario, which uses a more complex neural network architecture. The DP algorithm time decreases as the amount of trip constraints increases.

5 Conclusions

In this article it was investigated if combining the speed of feedforward neural networks and the accuracy of traditional search algorithms can be used to quickly generate attractive trip offers using real world flight data. The results show that for cities which contain as much as 50000 trajectories, it is possible to generate as much as 50 trip offers in which at least 80% of them match the best possible offers in under 2 seconds under the constraints of this experiment. This shows that the method can be applied in practice, and it will be strongly considered to be integrated into a newly developing trip planning software system.

References

- [1] R. M. Karp, „Reducibility among combinatorial problems,“ *Complexity of computer computations*, pp. 85-103, 1972.
- [2] C. K. Joshi, Q. Cappart, L.-M. Rousseau, T. Laurent, „Learning the Travelling Salesperson Problem Requires Rethinking Generalization,“ 2020.
- [3] D. P. Kingma, J. Ba, „Adam: A Method for Stochastic Optimization,“ 2014.
- [4] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, „Dropout: a simple way to prevent neural networks from overfitting,“ *The journal of machine learning research*, 2014.

Distributed machine learning for IoT

Volodymyr Kadzhaia, Aistis Raudys

Faculty of Mathematics and Informatics,
Vilnius University, Didlaukio st. 47, Vilnius
vkadzhaia@gmail.com, aistis.raudys@mif.vu.lt

Abstract. In the modern world, big data is used in machine learning, which is quite difficult to process on a single computer, so various methods for parallel processing of such data are being developed. But what about microcontrollers? In a cloud system, microcontrollers are often found, thanks to which they make pacification of various devices, and sometimes you have to work with big data. In microcontrollers, the memory is quite small and the processor is not as productive as on modern supercomputers. Therefore, many scientists propose various methods for parallel processing of big data for embedded systems, one of such methods is proposed by the author of this article.

Keywords: Distributed machine learning, IoT, MapReduce, grid computing, microcontrollers.

1 Introduction

In light of recent technological changes and advancements, distributed systems are becoming more and more popular. Many leading companies have built sophisticated distributed systems to handle billions of queries and updates without downtime. Also, in the modern world, traditional storage systems cannot expand indefinitely or at least fast enough, which is especially critical for deep learning tasks when there is more data than can fit on one machine. Keeping the data separate from the machine learning system where it is processed and where the model is trained can undermine efforts to speed up the neural network training process, especially when there is so much data that it does not fit on one machine.

2 Distributed system and Machine Learning

A distributed system is a system consisting of many devices distributed in space, each of which is independent of the others, but interacts with them to perform a common task. A process control system, characterized by the

construction of a distributed input-output system and decentralized data processing. Elements of the system can be located at a fairly large distance, and communication between them can be performed via the Internet.

One can think of several methods to parallelize and/or distribute computation across multiple machines and multiple cores [1]. There are some methods used to achieve faster training times:

- Local training:
 - The model and data are stored on a single machine [1].
- Multi-core:
 - The whole model and the data can be fit into the memory of a single machine with multiple cores. These multiple cores share the memory (PRAM model). There are two ways to use multiple cores to speed up the training process [1].
- Distribute training [2]:
 - Data parallelism [1]: The same operation is performed simultaneously (that is, in parallel) on the elements in the original collection or array. In data parallel operations, the source collection is partitioned so that multiple threads can work on different segments at the same time.

An important area of research is the development of machine learning methods specifically designed for large samples, as well as the development of distributed computing systems that allow the application of existing methods on large samples. These systems are required to:

- Flexibility - Since the code for map and reduce functions are written by the user, there is considerable flexibility in specifying the exact processing that is required over the data rather than specifying it.
- Scalability - The main problem with many applications is the ability to scale to increase data volumes. In particular, many are pursuing elastic scalability, which can be dynamically scaled up and down as compute requirements change. Such a “pay-as-you-go” service model is now widely adopted by the cloud computing service providers, and MapReduce can support it seamlessly through data parallel execution [3].
- Fault tolerance - The failed map task can be repeated correctly by reloading the replica. The failed reduce task can also be repeated by re-pulling the data from the completed map tasks [3].
- Versatility - Machine learning is usually a part of the data analysis process, including data preprocessing, feature calculation, selection

of training hyperparameters, analysis of the results of using the model in an applied problem, etc. Therefore, the possibility of solving all problems in one system is also a weighty argument.

There are the following approaches to developing programs for distributed machine learning: MapReduce computational model, MPI message passing, parameter server architecture, Spark system, graph computational models. Each of these approaches limits the developer in some way and assumes a certain style of software development.

When it comes to distribution, there are two fundamentally different ways of partitioning the problem across all machines:

- Data parallel: The data is partitioned as per number of worker nodes in the system. All workers apply the same algorithm to different partitions of data. The same model is available to all worker nodes (either through centralization, or through replication) so that a single coherent output emerges naturally. This assumes i.i.d (independent and identically distribution) of data samples which is valid for most of the ML algorithms.
- Model parallel: Exact copies of entire data is processed by the worker nodes, which operate on different parts of the model. The model is aggregate of all model parts. This cannot be applied to every ML algorithm, as parameters are often not divisible.
- Data and Model parallel: Ensemble applies a combination of two approach mentioned above. Training happens in two stages; first at local sites where the data is stored and second in the global site that aggregates over the individual results of the first stage. This global aggregation can be achieved by applying ensemble methods such as Bagging, Boosting, Random Forests, Stacking.

The topology of a distributed machine learning system is an important part of a distributed machine learning architectural design. Based on the architectural pattern, various nodes of a distributed system are used. However, the choice of template affects the role that a node can play in the degree of communication between nodes and in the fault tolerance of the system. Also one of the decisive factors for the topology is the degree of distribution:

- Centralized systems use a strictly hierarchical approach to aggregation, which occurs in one central place.
- Decentralized systems allow intermediate aggregation.

There are several schemes for distributed topologies:

- Trees: In Tree topology, each node communicates only with its parent or child nodes.
- Rings: In situations where the communication system does not provide efficient support for broadcast or where communication overhead needs to be kept to a minimum, nodes synchronize only through messages.
- Parameter server: Uses a decentralized set of workers with a centrally maintained shared state. All model parameters are stored in a segment on each parameter server. From the global shared memory, they can be read and written as a key and value store.
- Peer to peer: In a fully distributed model, each node has its own copy of the parameters, and workers interact directly with each other. This has the advantage of higher scalability and the elimination of single points of failure in the system.

3 Approach

A distributed system is a system consisting of many devices distributed in space, each of which is independent of the others, but interacts with them to perform a common task.

The proposed approach in distributed machine learning is to distribute data between different microcontrollers. For this, the project used I2C communication protocols as well as network protocols. Algorithms for data parallelization and algorithms for distributed systems were also used.

The principle of operation of this idea is that communication protocols between microcontrollers are used, as well as algorithms for dividing large data into smaller pieces that this microcontroller can process. This principle of operation can be similar to the principle of operation of the grid system. Figure 1 shows how microcontrollers are configured to communicate with each other via the I2C protocol. As can be seen from Fig. 1 communication protocol works through wires, thus the advantage may be that there is no delay in data transmission, as well as loss as is the case with wireless transmission (tcp, udp protocols, etc.). Moreover, the performance of I2C protocol is 100kB/s. One of the algorithms for distributing data between microcontrollers is shown in Figure 2, you can see that this algorithm resembles MapReduce which divides the data into smaller data, after which it is processed and combined to obtain the final result.

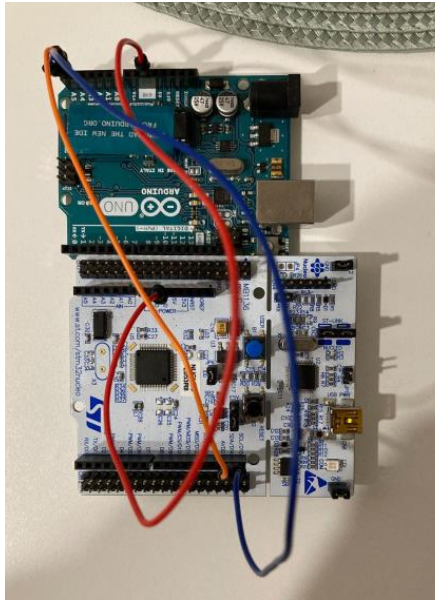


Fig. 1. Communication through I2C.

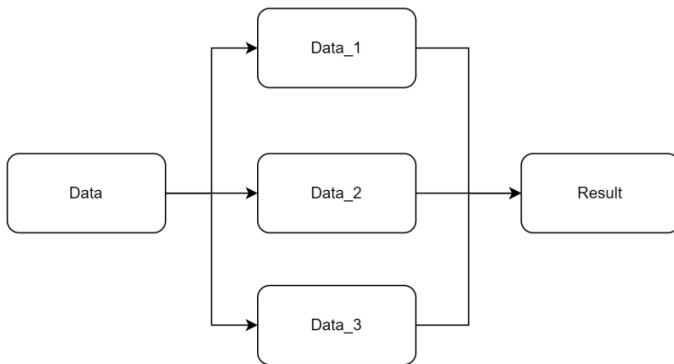


Fig. 2. Work process

Also in this project, various machine learning algorithms (KNN, k-means, perceptron, etc.) are used, which have shown themselves quite effectively. In the table 1 can be seen all existing algorithms with their time complexity.

One of the most important in the study and analysis of the algorithm is to determine the complexity of the algorithm. Big(O) notation is an algorithm complexity metric. It defines the relationship between the number of inputs and the steps taken by the algorithm to process those inputs. Table 1 shows algorithms in machine learning and their complexity.

Table 1. Complexity algorithm

Algorithm	Complexity	Worst complexity
k-NN	$O(n)$	$O(n*m)$
Support Vector Machine	$O(n^2)$	$O(n^3)$
Decision Tree	$O(\log(n))$	$O(n)$
k-Means	$O(n)$	$O(n^2)$
Naive Bayes	$O(n*d)$	$O(m*(n-m+1))$
Random Forest	$O(v * n \log(n))$	$O(\log(n))$
Linear Regression	$O(k^2*(n + k))$	-

3.1 I2C

I2C is a synchronous communication protocol, which means that both devices that communicate using this protocol must share a common clock signal. Since only 2 lines (wires) are used in this protocol, a synchronization signal must be transmitted along one of them, and useful information must be transmitted along the other. The defining feature of I2C is that each device on the bus must be connected to the clock (SCL for short) and data (SDA for short) lines via open-drain (or open-collector) output drivers. There are two conditions in I2C protocol that is START condition and STOP condition which are always generated by the master. In START condition high to low transition on the SDA line while SCL is high and in STOP condition a low to high transition on the SDA line while SCL is high. Data transfer is initiated by the master, which sends the address of the required device to the bus, clocking is also carried out by the master. But, at the same time, the Slave device has the ability to “hold” the clock line, as if informing the Master device that it does not have time to receive or send data, which is sometimes very useful. Clock generation is always the responsibility of the master; each master generates its own clock signal when sending data over the bus.

The exchange procedure ends with the master generating a STOP state - the transition of the state of the SDA line from a low state to a HIGH state with a HIGH state of the SCL line.

The START and STOP states are always generated by the master. The bus is considered to be busy after the START condition is latched. The bus is considered free some time after the STOP condition is fixed.

This acts as an "Attention" signal to all of the connected devices, all ICs on the bus will listen to the bus for incoming data [4]. After receiving the Slave address, the device must inform the master about the acceptance of the address, which confirms the very fact of the existence of a Slave device with such an address on the line. Having received the address, all ICs will compare it with their own address, if it does not match, they simply wait until the bus is released by the stop condition [4].

Confirmation is required when transferring data. The corresponding synchronization pulse is generated by the master. The transmitter releases (HIGH) the SDA line for the duration of the acknowledge clock. The receiver must hold the SDA line during the HIGH state of the acknowledgment clock in a stable LOW state.

In the event that the receiving slave cannot acknowledge its address (for example, when it is currently performing any real-time functions), the data line should be left in the HIGH state. The master can then issue a STOP signal to interrupt the data transfer.

If a master-receiver is involved in the transfer, then it must report the end of the transfer to the slave-transmitter by not acknowledging the last byte. The slave-transmitter must release the data line in order to allow the master to issue a STOP signal or repeat the START signal.

3.2 Microcontrollers

To successfully develop a schematic diagram, it is needed to immediately make a list of materials that will be used in its construction. In this work, 5 microcontrollers are used, below is a table 2 of used microcontrollers and their technical characteristics.

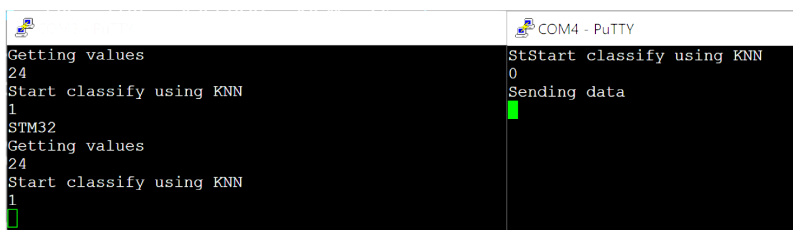
As can be seen from table 2, all 5 microcontrollers have different memory sizes, different processors, as well as different voltage consumption. According to these characteristics, these devices were selected.

Table 2. Characteristics of microcontrollers

Characteristics	Arduino Uno	Arduino Mega 2560	Arduino Nano 33 BLE Sense	STM32L053R8	Raspberry Pi 4
Processor	ATmega32P	ATmega2560	ARM Cortex-M4	ARM Cortex-M0+	ARM Cortex-A72
Clock Speed	16MHz	16MHz	64MHz	32MHz	1.5GHz
Flash Memory	32kB	256kB	1MB	64kB	32GB
SRAM	2kB	8kB	256kB	8kB	2GB
Voltage	5V	5V	3.3V	3.6V	5V
Digital I/O	14	54	14	51	40
Analog Pins	6	15	8		

3.3 Results

In Figure 3, you can see the principle of operation, the first step is data parallelization, then COM4 processes part of its data and in parallel transfers the rest of the data to the COM3 device. COM3, in turn, gives a message that it received data, displays the size and the beginning of the classification of the algorithm that was launched. Thus, it can be seen that the data transfer rate over the wire using the I2C protocol is 100kB / s, moreover, the probability of data loss is very low, as well as the delay if Internet protocols were used.



```
Getting values
24
Start classify using KNN
1
STM32
Getting values
24
Start classify using KNN
1

COM4 - PuTTY
StStart classify using KNN
0
Sending data
```

Fig. 3. Results.

Comparing my approach with the author of “Evaluation of MapReduce-Style Computation on a Cluster of Arduinos”, the algorithmic approach to splitting big data is similar, the only advantage in his approach is that his algorithm can read data from external devices. brands of microcontrollers, and not the use of machine learning methods.

4 Conclusions

In its current form, this work shows the effectiveness of using the proposed approach. Firstly, due to the flexibility of the I2C protocol, you can connect several devices that can communicate with each other and transmit information. Secondly, the algorithm that is used to implement data parallelization is similar to the MapReduce algorithm that is effectively used for commercial purposes.

At the moment, several machine learning algorithms are used, but even these algorithms can be used quite effectively in different projects. In addition, the author thinks that a faster SD card interface would be a possible improvement. Even if a single peripheral can provide I/O to an SD card at maximum I2C or SPI bandwidth, storage can be used as intermediate storage, greatly improving the size of the problems it can handle.

In the future, it is planned to add more machine learning methods as well as the use of Internet protocols.

References

- [1] Vishakh Hegde, Sheema Usmani. Parallel and Distributed Deep Learning. Stanford University, 2016
- [2] Beng Chin Ooi, Kian-Lee Tan, Sheng Wang, Wei Wang, Qingchao Cai, Gang Chen, Jinyang Gao, Zhaojing Luo, Anthony K. H. Tung, Yuan Wang, Zhongle Xie, Meihui Zhang, Kaiping Zheng. A distributed deep learning platform. ACM Multimedia, 2015
- [3] Feng Li, Beng Chin Ooi, M. Tamer Özsu, Sai Wu. Distributed Data Management Using MapReduce. ACM, 2014
- [4] Frederic Leens. An Introduction to I2C and SPI Protocols. ACM, 200

Precalculated arrays-based algorithms for the calculation of the Riemann zeta-function

Lukas Kuzma, Igoris Belovas,
Martynas Sabaliauskas

Vilnius University, Institute of Data Science and Digital Technologies
Akademijos st. 4, Vilnius LT-08412, Lithuania
lukas.kuzma@mif.vu.lt, igoris.belovas@mif.vu.lt,
martynas.sabaliauskas@mif.vu.lt

Abstract. In this paper, we continue the study of efficient algorithms for the computation of the Riemann zeta function on the complex plane. We introduce two precalculated arrays-based modifications of MB-method. We perform numerical experiments with these algorithms using *Zetafast* as a benchmark and apply the algorithms for the visualizations of fractal structures associated with the Riemann zeta function.

Keywords: Riemann zeta function, numerical algorithms, fractal structures.

1 Introduction

The paper continues the studies of efficient algorithms for the computation of the Riemann zeta function on the complex plane (see [1, 2] and Lukas Kuzma's Bachelor thesis [5]). We introduce modifications of *MB*-algorithm, aiming to accelerate computations by precalculating arrays of coefficients of a series for the Riemann zeta function and combining this approach with normal (Gaussian) approximations of the coefficients. The accuracy and the processing time of the algorithms are evaluated using *Zetafast* [4] as a benchmark.

Throughout this paper, we denote by $\Phi(x)$ the cumulative distribution function of the standard normal distribution, and by $\overline{\Phi}(x)$ we denote the corresponding tail distribution $\overline{\Phi}(x) = 1 - \Phi(x)$. $A \times B$ stands for the Cartesian product of sets A and B . $\lfloor x \rfloor$ and $\lceil x \rceil$ stand for the floor function and the ceiling functions respectively. All limits in the paper, unless specified, are taken as $n \rightarrow \infty$.

2 MB-algorithm

In [3] Borwein proposed an algorithm, applied to complex numbers $s = \sigma + it$ with $\sigma \geq 1/2$ and arbitrary t . Suppose

$$d_{n,k} = n \sum_{j=0}^k \frac{(n+j-1)!4^j}{(n-j)!(2j)!}. \quad (1)$$

then the Riemann zeta function is defined by the alternating series

$$\zeta(s) = \frac{1}{d_{n,n}(1-2^{1-s})} \sum_{k=0}^{n-1} \frac{(-1)^k (d_{n,n} - d_{n,k})}{(k+1)^s} + \gamma_n(s). \quad (2)$$

The ineligibility of the numbers $d_{n,k}$ (note that with large n factorials in the definition (1) become computationally expensive) has lead Belovas et al. [2] to introduction of a modified algorithm (*MB-algorithm*), avoiding the calculation of high factorials. Let us denote, along with Proposition 1 from [2],

$$l_{max} = \arg \max_{0 \leq k \leq n} \frac{(n+k-1)!4^k}{(n-k)!(2k)!}, \quad (3)$$

$$c_{n,k} = 1 - \frac{H_k}{H_n}, \quad n \in \mathbb{N}, \quad 0 \leq k \leq n,$$

here

$$H_l = H_{l-1} + e^{T_l - T_{l_{max}} + (l - l_{max}) \log 4}, \quad H_0 = e^{T_0 - T_{l_{max}} - l_{max} \log 4},$$

$$T_l = T_{l-1} + \log \frac{(n-l+1)(n+l-1)}{(2l-1)(2l)}, \quad T_0 = -\log n, \quad 1 \leq l \leq n. \quad (4)$$

Under these notations the Riemann zeta function is

$$\zeta(s) = \frac{1}{1-2^{1-s}} \sum_{k=0}^{n-1} \frac{(-1)^k c_{n,k}}{(k+1)^s} + \gamma_n(s). \quad (5)$$

The algorithm is nearly optimal in the sense that there is no sequence of n -term exponential polynomials that converge to the Riemann zeta function much faster than of the algorithm (see Theorem 3.1 in [3]).

3 Error term

The error term $\gamma_n(s)$ in (5) is characterized by the following proposition [1].

Theorem 3.1. *Let $\sigma \geq 1/2$, $t \geq 0$, $\varepsilon > 0$, $A = (2\pi)^{-1} \log 2$,*

$$s_k = 1 + ik/A, \quad k \in \mathbb{N}_0. \quad (6)$$

and $|s - s_k| \geq \varepsilon$, then

(i) *the error term of the series (5) is*

$$|\gamma_n(s)| \leq G_n \frac{(\cosh \pi t)^{1/2}}{|1 - 2^{1-s}|}, \quad (7)$$

(ii) *the series (5) to compute the Riemann zeta-function with d decimal digits of accuracy, require a number of terms*

$$n = \left\lceil B_1 t + B_2 d + C_\varepsilon \right\rceil, \quad (8)$$

where

$$\begin{aligned} G_n &= \frac{2}{(3 + \sqrt{8})^n}, & B_1 &= \frac{\pi/2}{\log(3 + \sqrt{8})}, \\ B_2 &= \frac{\log 10}{\log(3 + \sqrt{8})}, & C_\varepsilon &= \frac{\log 2 - \log(1 - 2^{-\varepsilon})}{\log(3 + \sqrt{8})}. \end{aligned} \quad (9)$$

The error terms are closely linked to the problem of selection of minimal number of terms in the series (5). Notably, under the conditions of Theorem 3.1, for $\varepsilon = 10^{-m}$, $m \in \mathbb{N}$, the series (5) to compute the Riemann zeta-function with d decimal digits of accuracy, require the number of terms (see Corollary 1 in [1])

$$n = \left\lceil B_1 t + B_2(d + m) \right\rceil + 1. \quad (10)$$

4 NA-modification

We have shown that the coefficients of *MB*-series satisfy (see Proposition 2 in [1])

$$c_{n,k} = \bar{\Phi} \left(\frac{k - \mu_n}{\sigma_n} \right) + O \left(\frac{1}{\sqrt{n}} \right), \quad \mu_n = \frac{n}{\sqrt{2}}, \quad \sigma_n = \frac{\sqrt{n}}{2\sqrt[4]{2}}. \quad (11)$$

This result allows us to introduce the corresponding normal approximation (*NA*) of the coefficients and to refine the number of terms n in the series (5),

$$\hat{n} = \lceil \mu_n + z_d \sigma_n \rceil, \quad (12)$$

for n large enough. Here $z_d = \Phi^{-1}(1 - 10^{-d})$. This modification is employed in Algorithm 2 (see the next section).

5 PA-based and NA-based algorithms

MB-method is well suited for calculating multiple values of the Riemann zeta functions while t is fixed. However, specific values of the Riemann zeta function for arguments with distinct t require the recalculation of the coefficients $c_{n,k}$, since the number of terms in series (5) depends on t . The first way to solve the problem is to establish limit theorems for the coefficients and to replace the coefficients with the corresponding approximations (*NA*-based approach). Alternatively, we can use a set of precomputed arrays of the coefficients $c_{n,k}$.

Let us precompute 16 arrays $\{c_{n_p,k}\}$, for $n_p = 2^{p+2}$, $1 \leq p \leq 16$ and select the number of terms in the series (5) by the rule

$$\hat{n} = \min_{n_p \geq n} n_p. \quad (13)$$

We can use the rule (13) for $0 < t < t_{max}$, with $t_{max} = 294000 < 2^{18}/B_1$. Note that C++ allows 16 decimal digits precision, hence we assume $c_{n,k} = 0$ if $c_{n,k} < 5 \cdot 10^{-17}$. We denote by k_p the corresponding index of the last non-zero element of $\{c_{n_p,k}\}$ array (see Table 1). Note that $k_p/n_p \rightarrow 1/\sqrt{2}$ in accordance with (11).

Algorithm 1 outlines the precalculated arrays-based modification of *MB*-method. This approach is more suitable for the calculation of specific values of the Riemann zeta function. Numerical experiments with the algorithm are presented in the next section.

Table 1: The indices of the last non-zero element k_p of $\{c_{n_p, k}\}$ array, $n_p = 2^{p+2}$.

p	n_p	k_p	p	n_p	k_p
1	8	8	9	2048	1601
2	16	16	10	4096	3111
3	32	32	11	8192	6097
4	64	64	12	16384	12013
5	128	125	13	32768	23776
6	256	233	14	65536	47189
7	512	437	15	131072	93882
8	1024	831	16	262144	187043

Algorithm 1 This algorithm will return values of the Riemann zeta function obtained by the precalculated arrays-based modification of MB-method. $L_k = \log k$ stand for the precalculated logarithms, $n_p = 2^{p+2}$, for $1 \leq p \leq 16$ and $t \in (0, t_{max})$.

```

1: function Zeta.PA( $\sigma, t$  : real numbers;  $d, m$  : natural numbers)
2:    $n \leftarrow \lceil ((\pi/2)t + (d + m)L_{10}) / \log(3 + \sqrt{8}) \rceil + 1$ 
3:   if  $n \leq n_1$  then
4:      $n \leftarrow n_1$ 
5:   else
6:      $p_{min} \leftarrow 1, p_{max} \leftarrow 16$ 
7:     while  $p_{max} - p_{min} > 1$  do
8:        $p \leftarrow \lfloor (p_{min} + p_{max} + 1) / 2 \rfloor$ 
9:       if  $n \leq n_p$  then
10:         $p_{max} \leftarrow p$ 
11:       else
12:         $p_{min} \leftarrow p$ 
13:       end if
14:     end while
15:      $n \leftarrow n_{p_{max}}$ 
16:   end if
17:    $S \leftarrow 0, p \leftarrow -1$ 
18:   for  $k \in \{0..k_p\}$  do
19:      $p \leftarrow -p$ 
20:      $S \leftarrow S + p c_{n, k} \exp(-\sigma L_{k+1}) (\cos(tL_{k+1}) - i \sin(tL_{k+1}))$ 
21:   end for
22:    $Zeta.PA \leftarrow S / (1 - 2 \exp(-\sigma L_2) (\cos(tL_2) - i \sin(tL_2)))$ 
23: end function

```

Algorithm 2 This algorithm will return values of the Riemann zeta function obtained by combining PA- and NA-methods.

```

1: function Zeta.NA( $\sigma, t$ : real numbers;  $d, m$ : natural numbers)
2:   if  $t \leq 32768$  then
3:      $n \leftarrow ((\pi/2)t + (d + m)L_{10})/\log(3 + \sqrt{8}) \uparrow + 1$ 
4:     if  $n \leq n_1$  then
5:        $n \leftarrow n_1$ 
6:     else
7:        $p_{min} \leftarrow 1, p_{max} \leftarrow 13$ 
8:       while  $p_{max} - p_{min} > 1$  do
9:          $p \leftarrow \lfloor (p_{min} + p_{max} + 1)/2 \rfloor$ 
10:        if  $n \leq n_p$  then
11:           $p_{max} \leftarrow p$ 
12:        else
13:           $p_{min} \leftarrow p$ 
14:        end if
15:      end while
16:       $n \leftarrow n_{p_{max}}$ 
17:    end if
18:     $S \leftarrow 0, p \leftarrow -1$ 
19:    for  $k \in \{0..k_p\}$  do
20:       $p \leftarrow -p$ 
21:       $S \leftarrow S + pc_{n,k} \exp(-\sigma L_{k+1})(\cos(tL_{k+1}) - i \sin(tL_{k+1}))$ 
22:    end for
23:     $Zeta.NA \leftarrow S/(1 - 2 \exp(-\sigma L_2)(\cos(tL_2) - i \sin(tL_2)))$ 
24:  else
25:     $n \leftarrow ((\pi/2)t + (d + m)L_{10} + L_2 - \log L_2)/\log(3 + \sqrt{8})$ 
26:     $\mu_n \leftarrow n/\sqrt{2}, \sigma_n \leftarrow \sqrt{n}/\sqrt[4]{32}, z \leftarrow \Phi^{-1}(1 - 10^{-d})$ 
27:     $k_0 \leftarrow \lceil \mu_n + z\sigma_n \rceil, k_1 \leftarrow \mu_n - z\sigma_n$ 
28:    function  $c(n, k$ : nonnegative integers)
29:      if  $k < k_1$  then
30:         $C \leftarrow 1$ 
31:      else
32:         $C \leftarrow 1 - \Phi((k - \mu_n)/\sigma_n)$ 
33:      end if
34:    end function
35:     $S \leftarrow 0, p \leftarrow -1$ 
36:    for  $k \in \{0..k_0\}$  do
37:       $p \leftarrow -p$ 
38:       $S \leftarrow S + pC(n, k) \exp(-\sigma L_{k+1})(\cos(tL_{k+1}) - i \sin(tL_{k+1}))$ 
39:    end for
40:     $Zeta.NA \leftarrow S/(1 - 2 \exp(-\sigma L_2)(\cos(tL_2) - i \sin(tL_2)))$ 
41:  end if
42: end function

```

Algorithm 2 combines precalculated arrays-based modification of *MB*-method (for $t \leq n_{\hat{p}}$) with *NA*-method (for $t > n_{\hat{p}}$). Numerical experiments have shown that $\hat{p} = 13$ is optimal.

6 Numerical experiments

Using Algorithm 1, Algorithm 2 and *Zetafast* method [4], we generate sequences of values of the Riemann zeta function $\{\zeta_{l,r}^{(PA)}\}$, $\{\zeta_{l,r}^{(NA)}\}$ and $\{\zeta_{l,r}^{(ZF)}\}$, $1 \leq l \leq N$, $1 \leq r \leq M$, $N = 10^5$, $M = 10^1$, taking as arguments uniformly distributed $z_{l,r} \in S_r \setminus \Theta$. Here (cf. Theorem 3.1)

$$S_r = \underbrace{(0.5, 2)}_{\Re z} \times \underbrace{(t_{r-1}, t_r)}_{\Im z}, \quad \Theta = \bigcup_{k=0}^{k_{max}} \{z : |z - s_k| \leq \rho\}, \quad (14)$$

$$t_r = t_0 + r\Delta_r, \quad t_0 = 0, \quad \Delta_r = t_{max}/M, \quad 1 \leq r \leq M,$$

with $k_{max} = \lfloor A t_{max} \rfloor = 32433$ and $\rho = 10^{-1}$. We take decimal digits of accuracy $d = 6$ and $m = 1$ (see (10)).

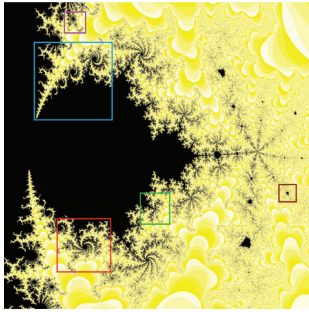
Remark 6.1. Let $t = \Im z_l$ and $k = \lfloor A(t + \rho) \rfloor$. Then $z_l \in \Theta$ if

$$k > \lfloor A(t - \rho) \rfloor \text{ and } |z_l - s_k| \leq \rho. \quad (15)$$

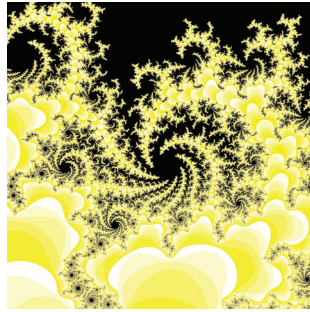
Proof. If $z_l \in \Theta$, then (cf. (6)) $\exists k \in \mathbb{N}_0: |t - k/A| \leq \rho$, or $k \in [A(t - \rho), A(t + \rho)]$. Note that the length of the interval is $2\rho A < 0.0221$. Such k exists if $\lfloor A(t - \rho) \rfloor < \lfloor A(t + \rho) \rfloor$. The proposition follows. \square

7 Visualization

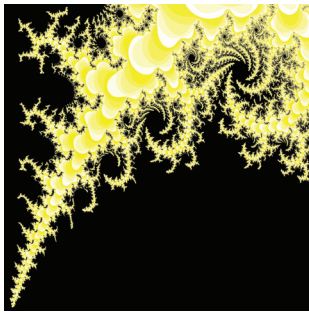
In this study we follow the guidelines of *SFH*-approach [1] visualizing fractal structures associated with the near-pole region of the Riemann zeta function (the function is calculated with $d = 6$ decimal digits of accuracy), see Figure 1. The ranges of the sets $(\sigma_1, \sigma_2) \times (t_1, t_2)$ and computation times of the figures are given in Table 2. All the frames are of 2000×2000 pixels size and have been generated using Python 3.10.2 version with AMD Ryzen 9 5950X processor (32 GB RAM, 4266 MHz).



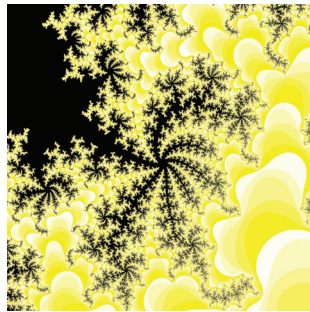
(a) main area



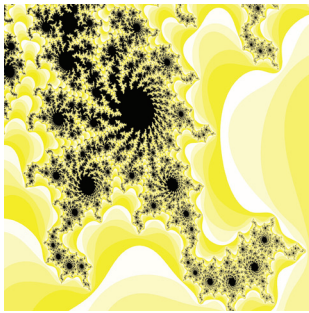
(b) area enclosed by the red square



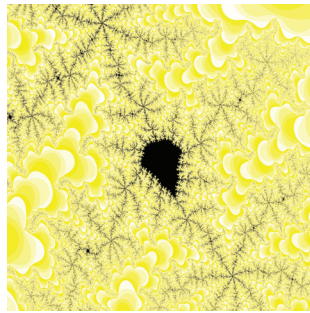
(c) area enclosed by the blue square



(d) area enclosed by the green square



(e) area enclosed by the violet square



(f) area enclosed by the brown square

Figure 1: Fractal structures associated with the near-pole region of the Riemann zeta function. Frames (b)-(f) are zoomed-in rectangles of (a). The ranges are given in Table 2.

Table 2: Ranges of the frames and corresponding computation times of Figure 1

Figure	σ_1	σ_2	t_1	t_2	Time (s)
1a	1.03000	1.04000	-0.03400	-0.02400	216
1b	1.03174	1.03347	-0.03279	-0.03106	230
1c	1.03100	1.03352	-0.02782	-0.02530	214
1d	1.03443	1.03543	-0.03123	-0.03023	222
1e	1.03197	1.03266	-0.02499	-0.02430	233
1f	1.03893	1.03953	-0.03051	-0.02995	228

8 Results

The numerical experiment has been performed on Intel® Core™ i7-8750H 2.2GHz (boosted to 4.0 GHz) processor, 16GB DDR4 RAM. The code was compiled using g++ 11.2.0 compiler using O3 optimization. Using *Zetafast* algorithm as a benchmark we calculate the accuracies

$$\begin{aligned}\delta_r^{(PA)} &= \max_{1 \leq l \leq N} \left| \zeta_{l,r}^{(PA)} - \zeta_{l,r}^{(ZF)} \right|, \\ \delta_r^{(NA)} &= \max_{1 \leq l \leq N} \left| \zeta_{l,r}^{(NA)} - \zeta_{l,r}^{(ZF)} \right|.\end{aligned}\tag{16}$$

and the processing times $\tau_r^{(\cdot)}$ of the sequences $\{\zeta_{l,r}^{(\cdot)}\}$ (see Table 3).

Table 3: Results of numerical experiments: accuracies $\delta_r^{(\cdot)}$ and processing times $\tau_r^{(\cdot)}$.

r	Accuracies		Processing times (s)		
	$\delta_r^{(PA)}$	$\delta_r^{(NA)}$	$\tau_r^{(PA)}$	$\tau_r^{(NA)}$	$\tau_r^{(ZF)}$
1	$5.09 \cdot 10^{-10}$	$5.09 \cdot 10^{-10}$	100.46	100.50	281.40
2	$2.99 \cdot 10^{-10}$	$2.80 \cdot 10^{-10}$	296.32	245.08	517.86
3	$8.15 \cdot 10^{-10}$	$7.58 \cdot 10^{-10}$	509.00	397.09	671.42
4	$8.29 \cdot 10^{-10}$	$7.42 \cdot 10^{-10}$	676.47	546.61	795.79
5	$1.46 \cdot 10^{-9}$	$1.43 \cdot 10^{-9}$	676.41	695.87	902.85
6	$1.89 \cdot 10^{-9}$	$1.72 \cdot 10^{-9}$	1335.08	866.20	998.96
7	$1.90 \cdot 10^{-9}$	$2.11 \cdot 10^{-9}$	1348.22	1017.30	1086.06
8	$2.09 \cdot 10^{-9}$	$2.10 \cdot 10^{-9}$	1348.04	1167.30	1171.45
9	$4.62 \cdot 10^{-9}$	$4.60 \cdot 10^{-9}$	1347.84	1317.71	1242.84
10	$3.93 \cdot 10^{-9}$	$3.85 \cdot 10^{-9}$	1348.16	1467.91	1314.32

9 Conclusions

The results show that both Algorithm 1 and Algorithm 2 return adequate, precise values of the Riemann zeta function. We see that the processing times of *PA*-algorithm are distributed in the pattern of a step function (in full compliance with the definition of Algorithm 1), while the processing times of *NA*-algorithm demonstrate a linear growth (in complete accordance with the proposition $n = O(t)$). Despite the fact that *PA*-algorithm is faster when t nears n_p , practical considerations favor the choice of *NA*-approach, especially if the range of the argument t is unrestricted. Indeed, the processing time of 10^5 Riemann zeta function values with arguments uniformly distributed in $\left(\bigcup_{r=1}^M S_r\right) \setminus \Theta$ is almost identical for Algorithm 1 and *Zetafast*, while Algorithm 2 is 15% faster, namely

$$\tau^{(PA)} = 904.32s, \quad \tau^{(NA)} = 785.81s, \quad \tau^{(ZF)} = 902.97s, \quad (17)$$

thus corroborating the conclusion.

References

- [1] Belovas, I.; Sabaliauskas, M; Kuzma, L. Series with binomial-like coefficients for the investigation of fractal structures associated with the Riemann zeta function. *Fractal Fract.* **2022** (in review). doi.org/10.20944/preprints202205.0122.v1
- [2] Belovas, I.; Sakalauskas, L.; Starikovičius, V. A method for accelerated computation of the Riemann zeta function on the complex plane. *Publ. Math. Debrecen* **2022**, 100(1-2), 167-184. doi.org/10.5486/PMD.2022.9120
- [3] Borwein, P. An efficient algorithm for the Riemann zeta function. In *Constructive, Experimental, and Nonlinear Analysis (Limoges, 1999)* Canadian Mathematical Society Conference Proceedings **2000**, 27, 29-34. cecm.sfu.ca/~pborwein/PAPERS/P155.pdf
- [4] Fischer, K. *The Zetafast algorithm for computing zeta functions* **2017**. arxiv.org/abs/1703.01414
- [5] Kuzma, L. Visualization of surfaces and 3D curves, associated with the Riemann zeta function and its non-trivial zeros arrangement (Bachelor thesis) **2021**. lvb.lt/permalink/f/16nmo04/ELABAETD107115273 (in Lithuanian)

Nefunkcinių reikalavimų kintamumas paslaugų architektūros stiliaus sistemose

Vieslav Lapin

Vilniaus universitetas, Matematikos ir informatikos fakultetas,
Didlaukio g. 47, LT-08303 Vilnius
vieslav.lapin@mif.stud.vu.lt

Santrauka. Šiame straipsnyje aptariamas kintamų nefunkcinių reikalavimų specifikuojimas ir modeliavimas kuriant paslaugų architektūros stiliaus programų sistemas. Paslaugų sistemų kūrime šiai problemai spręsti dar nėra skiriama pakankamai dėmesio. Pasiūlytas kintamų savybių specifikuojimo metodas remiasi modeliais grindžiama sistemų kūrimo paradigma ir naudoja produktų šeimos kūrimo idėjas. Paslaugos variantai yra nustatomi analizuojant paslaugos nefunkcines savybes ir sudaromas individualizuotas paslaugos teikimo procesas, kurio modelis gali būti generuojamas automatizuotai.

Raktiniai žodžiai: paslaugomis grindžiamos sistemos, nefunkcinės savybės, adaptuojamumas, kintamumo specifikuojimas.

1 Įvadas

Programų sistemos turėtų būti kuriamos taip, kad būtų nesunku jas modifikuoti, kai to reikalauja besikeičiantys vartotojo poreikiai. Sistemų adaptuojamumą užtikrina paslaugomis grindžiamos sistemos, kurios yra išskirstytos ir sudarytos iš silpnai sukibusių, savarankiškų ir nuo realizavimo platformos nepriklausomų paslaugų [1]. Jos gali būti sukomponuojamos vykdymo metu, kas įgalina pakartotinį paslaugų panaudojimą ir adaptuojamumą.

Tačiau netgi fundamentiniai paslaugų architektūros stiliaus sistemų principai nenumato darbo su kintamybėmis kaip esminės šio tipo sistemų savybės [1]. Todėl turi būti išspręstos šios problemos:

- Siekiant bet kuriuo metu gauti sistemos paslaugas, turi būti užtikrinta galimybė vieną paslaugą pakeisti kita to paties funkcionalumo, bet galbūt kitos kokybės paslauga.
- Siekiant užtikrinti paslaugos kokybę ir optimizuoti sistemos veikimą, turi būti galimybė perkonfigūruoti visą sistemą, t. y. turi būti galimybė paslaugos realizaciją pakeisti kita ar net pasinaudoti kito teikėjo paslauga.

- Siekiant užtikrinti tinkamą sistemos veikimą įvairiuose kontekstuose, apimant skirtingas aplinkas ir naudotojų ypatumus, turi būti galimybė pakoreguoti sistemos veikimą. Tam reikia turėti skirtingas savybes įgyvendinančius skirtingus tos pačios paslaugos egzempliorius.
- Paslaugos paprastai nėra kuriamos atsižvelgiant į galimybę jomis naudotis tam tikro jų kintamumo ribose ar lanksčiai pritaikomas konkrečiam atvejui. Siekiant paslaugų kintamumo, kartu turi būti kuriami ir atitinkami įgyvendinimui reikiami artefaktai (tokie kaip specifikacijos ir modeliai), kurie sudarytų prielaidas kintamumui užtikrinti [2].
- Jei paslaugų kintamumas užtikrinamas nesistemiškai ir sistema adaptuojama nekontroliuojamu būdu, kyla sistemos dalių interoperabilumo problema [3].

Kuriant paslaugomis grindžiamas sistemas atsiranda savita darbo su šio tipo sistemų kintamumu specifika. Pirma, nefunkcinės charakteristikos, tokios kaip prižiūrimumas, apsauga, patikimumas ir našumas, pasižymi didesniu įvairumu nei tai yra kitokio tipo sistemose [4]. Antra, paslaugų nefunkcinės charakteristikos yra apibrėžiamos paslaugų lygmens susitarimuose (toliau SLA), tai yra dėl paslaugos kokybės turi susitarti paslaugos teikėjas ir gavėjas. Ilgą kūrimo istoriją turinčiose programų sistemose nefunkcinių savybių kintamumo užtikrinimas yra neišspręsta problema [5].

Šiame straipsnyje aptariamas paslaugomis grindžiamų sistemų nefunkcinių savybių kintamumo specifikavimo metodas, užtikrinantis galimų alternatyvų nagrinėjimą ir paslaugos individualizuoto varianto konfigūravimą. Antrame skyriuje apžvelgiami susiję darbai, nagrinėjantys paslaugų stiliaus sistemų kintamųjų specifikavimą ir modeliavimą. Trečiame skyriuje aptariamas paslaugų stiliaus sistemų nefunkcinių savybių kintamųjų specifikavimo ir modeliavimo metodas. Paskutiniame skyriuje pateikiamos išvados.

2 Susiję darbai

Darbas su kintamomis sistemos savybėmis vykdomas visose programų sistemų kūrimo stadijose [6]. Tačiau literatūroje pagrindinis dėmesys skiriamas architektūrinio lygmens sprendimams ir kintamųjų tipams, akcentuojant nefunkcinių reikalavimų kintamumą. Kintamumui užtikrinti plačiausiai taikomas produktų linijų inžinerijoje naudojamas sprendimo būdas [7] [8]. Darbe [9] daroma išvada, kad produktų linijų inžinerijos sprendimai nepakankami nefunkcinių savybių kintamumui užtikrinti paslaugų stiliaus ar-

chitektūros sistemose. Antras mažiau taikytas sprendimo būdas – šablonų naudojimas kintamybės modeliuoti ir variacijų taškams aprašyti [5]. Paslaugų kokybės atributų kintamumo modeliavimas tik pradėdamas tyrinėti [10].

Kintamumas modeliuojamas pasirenkant procesų variantų klases iš kelių veiklų šakų. Kiekvienas šakojimosi taškas atitinka variacinį tašką arba sprendimo tašką. Variacija produktų šeimose nusako kaip šeimos nariai skiriasi vienas nuo kito [11]. Variacijos taškas yra sprendimo (išsišakojimo) taškas kartu su pasirinkimais. Kiekvienas pasirinkimas, išsišakojantis iš variacinio taško (funkcijos ar kokybės produktų linijų kontekste), yra apibrėžiamas kaip *variantas* [12]. Produktų linijų kūrime nėra išreikštinių skirtumų tarp variacinių ir sprendimo taškų.

Frankova ir kiti siūlo verslo procesų su fiksuotais kokybės matais išvedimą iš ankstyvųjų reikalavimų [13] [14]. Projektuojant paslaugų sistemų verslo procesus svarbu užtikrinti ne tik tam tikrus darbų srautus, bet ir tinkamą paslaugos vykdymo kokybę, t. y. nefunkcines savybes, fiksuojamas SLA. BP&SLA metodika formuoja ankstyvuosius reikalavimus ir aprašo naudotojo valdomos transformacijos ir samprotavimo priemones, ko rezultatas yra verslo procesai, aprašyti Secure BPEL kalba bei SLA kontraktų aibė. Nefunkcinės savybės yra derinamos tarp užsakovo ir paslaugos tiekėjo specifikuojant SLA.

Rosa ir kiti išskiria šiuos specifikuojamų verslo procesų kintamumo mechanizmus: mazgų konfigūravimą, elementų anotavimą, veiklos specializavimą ir fragmentų adaptavimą [15]. Šiame metode variacijos taškas yra adaptavimo proceso modelio mazgas, kuriam priskiriami skirtingi nustatymai. Veiklos, įvykiai, sietuvai (angl. *gateway*), resursai ir objektai žymimi kaip konfigūruojami mazgai. Adaptavimas yra pasiekiamas pasirenkant vieną nustatymą vienam konfigūruojamam mazgui. Elementų anotavimas suteikia valdymo srautų mazgams grafines anotacijas su dalykinės srities ypatybėmis. Specializuojant veiklas abstrakčių veiklų variaciniams taškams priskiriami vienas arba daugiau variantų. Fragmentų adaptavime ribojamas arba išplečiamas proceso modelis naudojant keitimo operacijas, pvz.: trinti, įdėti, perkelti.

DiVA metodas apibrėžia dinaminio kintamumo adaptuojamose paslaugomis grindžiamose sistemose valdymo bei kompleksiško kontrolės būdą [16]. Todėl nekontroliuojamas konfigūracijų skaičiaus augimas yra suvaldomas, modeliujant kintamumo dimensijas ir variantus, o ne visą pa-

slaugos konfigūraciją. Adaptacijos logika jame yra aprašoma aukštame abstrakcijos lygmenyje, tokiu būdu atskiriama adaptacija nuo pačios sistemos. Aspektų modeliai validuojami ir verifikuojami ankstyvuose sistemos kūrimo gyvavimo ciklo etapuose, todėl galimos problemos atsiskleidžia projektavimo, o ne vykdymo metu, kai jų sprendimas taptų sudėtingas ir brangus. Produktų šeimos kintamumo specifikavimo ir modeliavimo metodika DiVA skirta modeliuoti, analizuoti, vertinti ir komponuoti kintančias prisitaikančių sistemų funkcinį savybių konfigūracijas paslaugos gyvavimo cikle [16]. Ši metodika neapima paslaugas realizuojančio proceso automatizuoto specifikavimo, nefunkcinį savybių specifikavimo ir modeliavimo.

3 Paslaugų ir procesų kintamų nefunkcinį savybių specifikavimas

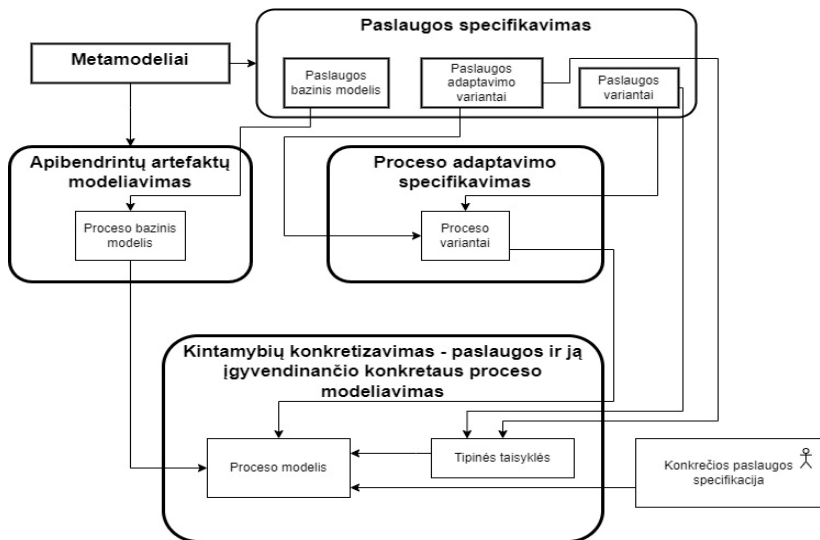
Paprastai yra kuriamos ne atskiros paslaugos, o paslaugų šeimos, skirtos individualizuoti paslaugų teikimą, kai paslaugos adaptuojamos specifiniam naudojimo kontekstui. Naudojimo kontekstas apima naudotojus, jų tikslus ir veiklas, veikimo aplinką ir technologijas [17]. Verslo procesus įtakoja suteikiamos paslaugos ir naudojimo konteksto kitimas, kas savo ruožtu sąlygoja funkcinį ir nefunkcinį reikalavimų kitimą.

Šiame skyriuje aptariamas kintamų savybių specifikavimo metodas, pateikiami paslaugos ir proceso metamodeliai, aprašomas rezultatų gavimo procesas, pateikiama jo konkretizacija ir pagrindžiantys argumentai.

3.1 Metodo bendroji charakteristika

Kintamų savybių specifikavimo metodas įgyvendinamas naudojantis modeliais grindžiama sistemų kūrimo paradigma ir produktų šeimos kūrimo idėjomis. DiVA metodika yra modifikuojama ir išplečiama, užtikrinant kintančių procesų nefunkcinį savybių kintamumo specifikavimą ir modeliavimą.

Paslaugas įgyvendinančiose sistemose skirtingi vartotojai gali gauti skirtingas to paties tipo paslaugas. Tam tikslui identifikuojamos kintamos paslaugų ir jas įgyvendinančių procesų dalys, kurios adaptuojamos konkrečiam atvejui. Siūlomame metode paslaugos variantai yra išskiriami iš naudojimo scenarijų, kuriuose akcentuojamos paslaugos nefunkcinės savybės. Metodas apima paslaugos specifikavimą, apibendrintų artefaktų modeliavimą, proceso adaptavimo specifikavimą ir kintamųjų konkretizavimą (1 pav.). Stačiakampiai su apvalintais kampais žymi procesus, stačiakampiai – artefaktus.



1 pav. Kintamybių specifikavimo koncepcinis modelis.

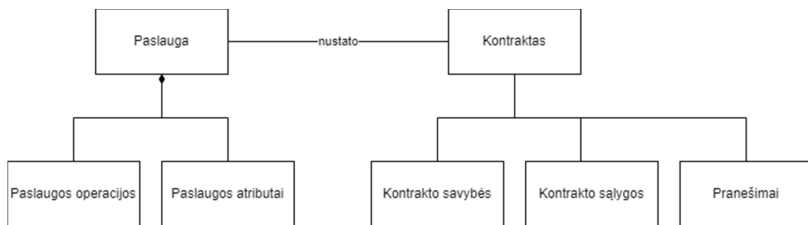
3.2 Paslaugos ir proceso metamodeliai

Paslauga suprantama kaip savybių ir funkcionalumo visuma (atributai ir operacijos); paslauga teikiama pagal kontrakte nustatytas sąlygas (2 pav.). Operacijos yra veiksmai susiję su teikiamais paslaugos variantais. Paslaugos atributai apibrėžia esybes ir jų savybes, kuriomis manipuliuoja operacijos. Paslaugos kontrakte apibrėžiami paslaugos teikimo kokybės lygmenys ir teikiamų paslaugos variantų sąlygos.

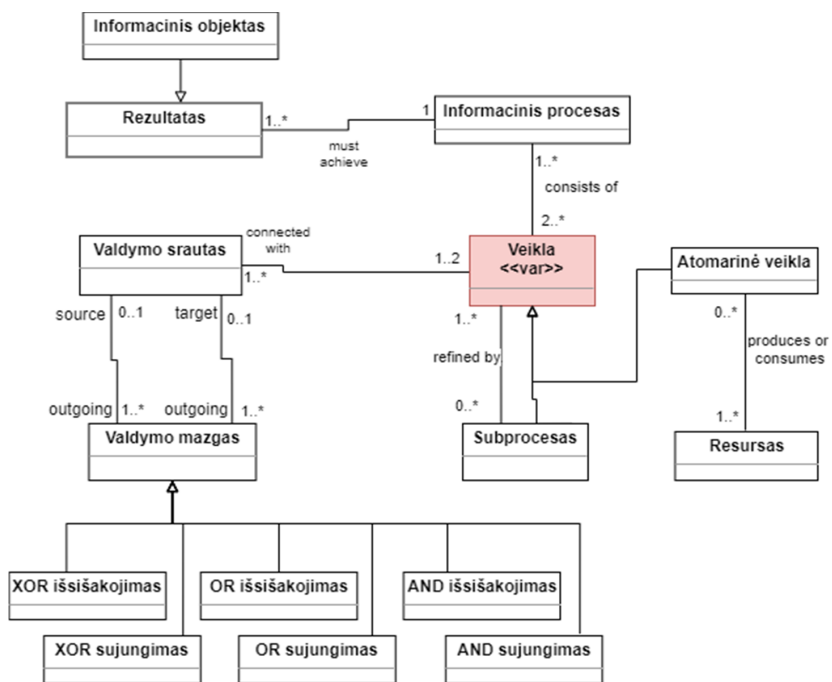
Projektuojant paslaugą su kintamybėmis, paslaugą realizuojantys verslo procesai yra sudaryti iš kintamų ir nekintamų veiklų (3 pav.). Siūlomą verslo proceso metamodelį (dalinai adaptuojamas iš [18]) sudaro šie elementai:

- 1) informacinis procesas,
- 2) veiklos (kintamos ir nekintamos),
- 3) atominė veikla,
- 4) subprocesas.

Informacinis procesas yra sudaromas iš veiklų. Kiekviena veikla turi teikti rezultatą, kuris yra informacinis objektas teikiantis reikalingą informaciją veiklą inicijavusiam naudotojui. Veikla atitinka atominę veiklą arba



2 pav. Paslaugos metamodelis.



3 pav. Verslo proceso metamodelis (kintamas elementas pažymėtas raudonai)

subprocesą. Atominė veikla yra nedaloma proceso dalis. Subprocesas, savo ruožtu, yra sudarytas iš kitų veiklų, dažnai atominių, rečiau dar iš kitų subprocesų. Iš paslaugomis grindžiamos sistemos perspektyvos, atominėje veikloje yra apdorojami informaciniai išteklių. Veiklų eiliškumą nurodo valdymo srautas. Valdymo srautas yra valdomas operatorių, vadinamų valdymo mazgais. Valdymo mazgai gali būti šių rūšių:

1. XOR išsišakojimas – iš skirtingų šakų pasirenkamas vienas pagal nustatytą sąlygą.
2. XOR sujungimas – pasirinkta šaka baigiasi ir grįžtama į bendrą proceso srautą.
3. OR išsišakojimas – iš skirtingų šakų pasirenkama viena arba daugiau pagal nustatytą sąlygą.
4. OR sujungimas – pasirinktos šakos baigia vykdomą ir visi srautai susilieja atgal į vieną srautą.
5. AND išsišakojimas – proceso šakos vykdomos lygiagrečiai.
6. AND sujungimas – proceso šakos baigia vykdomą ir visi srautai susilieja atgal į vieną srautą.

3.3 Kintamumo mechanizmas

Atsižvelgiant į literatūros analizėje nagrinėtą medžiagą ir ją apibendrinant, nustatyti atliekamo tyrimo kontekste nagrinėtini kintamumo įgyvendinimo mechanizmai, kuriems įgyvendinti anotuotos veiklos gali būti:

1. variacinis taškas,
2. numatytas variantas,
3. variantas,
4. nulinė veikla,
5. neprivaloma veikla.

Variacinis taškas nusako veiklos elementą, kuris specifikuojamas arba projektavimo metu dar neturi konkrečios suplanuotos veiklos, bet galimi keli variantai iš kurių vienas turi būti pasirinktas. Jie pasirenkami atsižvelgiant į pasirinkimo metu susiklosčiusią situaciją. Galimai toks pasirinkimas yra atliekamas paslaugos vykdymo metu.

Numatytas variantas yra veikla, kuri turėtų būti įkelta į variacinio taško vietą ir inicijuojama bendru atveju, jeigu iniciatorius nedaro išreikštinio sprendimo, kurį variantą naudoti.

Variantas yra veikla, kurią naudotojas gali pasirinkti iš galimų alternatyvų, kurios bus vykdomas tame variaciniame taške. Kitaip negu numatyto varianto atveju, varianto veiklą naudotojas turi pasirinkti išreikštiniu būdu.

Nulinė veikla yra panašus į variacinį tašką, tik gali būti praleistas, tai yra nepasirenkamas joks iš apibrėžtų variantų, jeigu konkrečioje situacijoje tam nėra poreikio.

Neprivaloma veikla yra veiklos realizacija ir nėra variacinis taškas. Ji gali įvykti arba ne.

Apibrėžti kintamumo mechanizmai leidžia specifikuoti proceso kintamumus slepiant konkretizacijos detales, kurios yra apibrėžiamos žemesniame realizacijos lygmenyje. Tokiu būdu naujo varianto atsiradimas neturi poveikio bazinio proceso specifikacijai.

3.4 Paslaugos specifikavimas

Paslaugos bazinis modelis nusako operacijas, nekintančias paslaugos teikimo procese. Tai pagrindinė paslaugos dalis, kuri būtina visuose jos variantuose.

Paslaugos adaptavimo variantai yra susiję su paslaugos kintamumu ir dinamiu adaptavimu. Adaptavimo variantai apibrėžiami remiantis naudojimo konteksto scenarijais suderintais su suinteresuotų šalių tikslais. Naudojimo konteksto scenarijuose specifikuojami paslaugos bazinis modelis, adaptavimo variantai ir paslaugos variantai. Adaptavimo variantai nusako paslaugų nefunkcines savybes.

Aukščiausio lygio sistemos ypatybes atitinka aukščiausio lygmens suinteresuotųjų šalių tikslus, kurie realizuojami kaip klasikinio pavidalo paslaugos. Skirtingi iš naudojimo konteksto išplaukiantys tikslai leidžia suformuluoti adaptavimo kategorijas, kurios savo ruožtu nusako potencialius variacijos taškus ir **paslaugos variantus**. Papildomai, bet koks svarbus kontekstas ir apribojimai yra irgi išgaunami, kad galima būtų įsitikinti, jog visas adaptacijas galima teisingai pritaikyti dinamiškai be vykdymo metu sukeltų klaidų ar konfliktų.

Paslaugos specifikacijoje išvardijamos minėtos operacijos ir atributai (1 lentelė).

Paslauga apibrėžia kontraktą, kuris išvardina nefunkcines paslaugos savybes ir aprašo jos teikimo kokybės matus. Vartotojas pasirenka atitinkantį jo poreikius paslaugos teikimo kokybės lygmenį. Paslaugų lygmenų kriterijai yra apibrėžiami, analizuojant naudojimo konteksto scenarijus (2 lentelė).

1 lentelė. Paslaugos specifikacija

Paslaugos operacijos ir atributai	Apibrėžimas
Paslaugos pavadinimas	
Paslaugos aprašas	Trumpas paslaugos aprašas
Bazinis paslaugos variantas	Aprašomi bendri paslaugų komponentai
Paslaugos variantai	Aprašomi kintami paslaugos komponentai
Paslaugos prieinamumas	Numatytas procentais paslaugos prieinamumas
Paslaugos savininkas	Atsakingo asmens kontaktai
Paslaugos naudotojai	Paslaugos tikslinės naudotojų grupės
Paslaugos veikimo kokybės matai	Vienas ar daugiau bazinio lygmens veikimo matų
Pagrindiniai tikslų rodikliai	Bendra paslaugos nauda klientui

2 lentelė. Paslaugos kontrakto aprašas

Paslaugų adaptavimo variantai	Paslaugos lygmuo 1	Paslaugos lygmuo 2	...	Paslaugos lygmuo N
Nefunkcinė charakteristika 1	Kriterijus ₁₁	Kriterijus ₂₁	...	Kriterijus _{N1}
Nefunkcinis charakteristika 2	Kriterijus ₁₂	Kriterijus ₂₂	...	Kriterijus _{N2}

3.5 Apibendrintų artefaktų modeliavimas

Šiame etape, iš ankstesnės paslaugos specifikavimo dalies sudaryto paslaugos bazinio modelio, sudaromas proceso bazinis modelis. Proceso baziniame modelyje yra sukomponuojami paslaugos specifikacijoje apibrėžti paslaugos proceso bazinės veiklos. Procese nustatomos bazinės veiklos ir jų eiliškumas.

3.6 Proceso adaptavimo specifikavimas

Šiame etape yra dekomponuojami paslaugos specifikacijoje išvardinti adaptavimo variantai, kurie susiejami su paslaugos variantais. Rezultate suformuojami proceso variantai. Dekomponuojant adaptavimo variantus, yra analizuojamos nefunkcinės savybės (NF) ir identifikuojamos charakteristikos, įtakančios paslaugos varianto pasirinkimą. Charakteristikoms yra priskiriamos vertės, gaunamos analizuojant paslaugos naudojimo kontekstą (3 lentelė).

3 lentelė. Adaptavimo kriterijų (nefunkcinių savybių) dekompozicija

Paslaugos variantai	NF ₁			NF ₂			NF _M		
	Ch ₁₁	...	Ch _{1N}	Ch ₂₁	...	Ch _{2N}	Ch _{M1}	...	Ch _{MN}
Bazinis	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
Variantas1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
Variantas2	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1

3.7 Kintamybių konkretizavimas

Konkretizuojant kintamybes modeliuojamas konkretaus paslaugos varianto pasirinkimo procesas. Pasirinkimas yra apibrėžiamas tipinėse taisyklėse, kuriose susiejami adaptavimo variantai su paslaugos variantais. Proceso modelis nusako konkretaus paslaugos varianto proceso realizaciją. Tipinės taisyklės yra aprašomos pasitelkiant šablona, kuriame yra keturi elementai:

- **pavadinimas** – identifikuoja šablona,
- **taikymo sąlyga** – aprašoma problema ir sąlygos, kada šablonas gali būti pritaikomas,
- **sprendinys** – aprašomas problemos sprendinys – elementai, jų ryšiai bei sąveika. Sprendinys yra apibendrintas ir turi būti konkretizuojamas konkrečiu atveju,
- **rezultatas** – aprašomas tipinio sprendinio taikymo rezultatas, kuris gali būti nusakytas po-sąlygomis.

Tipinės taisyklės specifikuojamos atsižvelgiant į visas nefunkcines savybes (4 lentelė) arba į konkrečią vieną nefunkcinę savybę (5 lentelė).

4 lentelė. Variacinio taško, atsižvelgiančio į visas nefunkcines charakteristikas, tipinė taisyklė

Pavadinimas	VariacinisTaškas1
Taikymo sąlyga	Jeigu proceso variaciniame taške (<<VarPoint>>) turi būti sukonkre-tinta veikla ir reikia užtikrinti visas paslaugos nefunkcines savybes.
Sprendinys	Pagal nustatytus leidžiamus visų adaptavimo variantų (nefunkcinių charakteristikų) kriterijų reikšmių intervalus, atrenkami proceso veiklos variantai (vienas arba daugiau), atitinkantys reikalavimus.
Rezultatas	Paslaugos tiekimo procesas su reikalavimus tenkinančiais veiklos variantais (<<Variant>>, vienas arba daugiau).

5 lentelė. Variacinio taško, atsižvelgiant į vieną nefunkcinę charakteristiką, tipinė taisyklė

Pavadinimas	VariacinisTaškas2
Taikymo sąlyga	Jeigu proceso variaciniame taške (<<VarPoint>>) turi būti sukonkretinta veikla ir reikia užtikrinti vieną paslaugos nefunkcinę savybę.
Sprendinys	Pagal nustatytus leidžiamus vieno adaptavimo varianto (nefunkcinę charakteristiką) kriterijų reikšmių intervalus, atrenkami proceso veiklos variantai (vienas arba daugiau), atitinkantys reikalavimus.
Rezultatas	Paslaugos tiekimo procesas su reikalavimus tenkinančiais veiklos variantais (<<Variant>>), vienas arba daugiau).

Proceso modelis nusako konkrečios paslaugos varianto realizaciją. Jeigu sąlygas atitinka keli variantai, galutinį pasirinkimą padaro vartotojas. Proceso modelis yra kuriamas atsižvelgiant į proceso bazinį modelį, proceso variantus, tipines taisykles ir konkrečios paslaugos specifikaciją, formuojamą realiuoju laiku iš konkretaus naudojimo konteksto duomenų.

Apibendrinant pabrėšime, kas pasiūlytas nefunkcinių savybių susiejimas su paslaugų šeimos kūrimu. Tai realizuojama šiuo būdu:

1. Analizuojant paslaugos naudojimo kontekstą yra kuriama paslaugos specifikacija. Jos pagrindu išskiriamas paslaugos bazinis modelis, naudojimo kontekstą atitinkantys adaptavimo variantai ir paslaugos variantai.
2. Modeliuojant apibendrintus artefaktus proceso baziniame modelyje identifikuojamos pagrindinės nekintamos ir kintamos veiklos.
3. Proceso adaptavimo specifikavimo fazėje identifikuojami proceso variacinių taškų variantai, bei siejami su nefunkcinių savybių charakteristikomis, priskiriant įverčius.
4. Konkretizuojant kintamybes, tipinės taisyklės aprašo konkretų pasirinkimą konkrečiame naudojimo kontekste.
5. Galiausiai, naudojant proceso bazinį modelį, proceso variantus ir jų adaptavimo variantus, tipines taisykles, formuojamas reikiamas proceso modelis. Jeigu jame išlieka keli pasirinkimai, galutinį sprendimą priima vartotojas.

Metodo charakterizuojamas eile ypatumų. Pirma, kreipiant pagrindinį dėmesį į variantų modeliavimą, nuo skirtingų kontekstų ir nuo paslaugos savybių priklausantis variantų skaičiaus padidėjimas gali būti valdomas kontroliuojamu būdu. Antra, naudojant modeliais grindžiamą sistemų kūrimo paradigmą, konceptai ir adaptavimo logika gali būti specifikuojama aukš-

tesniame abstrakcijos lygmenyje, taip užtikrinant turinių atskyrimą koncepcinių, kintamų ir konkrečių turinių. Trečia, taikant produktų šeimos idėjas, adaptuotas turinys gali būti generuojamas automatizuotai, kas apima konkretaus artefakto gavimą iš apibendrinto modelio.

4 Išvados

Šiame darbe sukurtas paslaugomis grindžiamų sistemų kintamumo specifikavimo metodas, kuriame paslaugos variantai yra nustatomi analizuojant paslaugos nefunkcines savybes ir sudaromas individualizuotas paslaugos teikimo procesas. Atlikus literatūros analizę nustatyta, kad išnagrinėtose metodikose modeliuojami tik funkcinių savybių kintami turiniai ir specifiikuojami paslaugomis grindžiamų sistemų procesai, apimant nefunkcinius reikalavimus, bet nekalbama apie kintamus turinius.

Sukūrus paslaugomis grindžiamų sistemų paslaugų variantų kintamumo specifikavimo metodą, nustatyta, kad nefunkcines savybes nusakančių charakteristikų pagrindu ir naudojant tipines taisykles proceso variantams pasirinkti. Taikant produktų šeimos idėjas ir tipines taisykles, konkrečiai paslaugai adaptuotas turinys, t. y., konkretaus proceso modelis, gali būti generuojamas automatizuotai. Taip pat nustatyta, kas kreipiant pagrindinį dėmesį į kintamųjų modeliavimą, nuo skirtingų kontekstų ir nuo skirtingų paslaugos savybių priklausantis variantų skaičiaus padidėjimas gali būti valdomas kontroliuojamu būdu.

Literatūra

- [1] Erl, T. (2005). *Service-Oriented Architecture (SOA): Concepts, Technology, and Design*. Upper Saddle River, NJ. Prentice Hall.
- [2] Medeiros, F. M., de Almeida, E. S. & de Lemos Mei, S. R. (2009). Towards an approach for service-oriented product line architectures. *Workshop on Service-oriented Architectures and Software Product Lines*. San Francisco, CA.
- [3] Jiang, J., Ruokonen, A. & Systa, T. (2005). Pattern-based variability management in web service development. *Third European Conference on Web Services, IEEE*, 12 p.
- [4] Gu, Q. & Lago, P. (2009). Exploring service-oriented system engineering challenges: a systematic literature review. *Service Oriented Computing Applications*, 3(3), 171-188.
- [5] Galster, M. & Avgeriou, P. (2013). *Variability in Web Services, Systems and Software Variability Management*. R. Capilla, J. Bosch & K. Kang, Mont., Springer Verlag, p. 269–278.
- [6] Galster, M., Weyns, D., Tofan, D. & Michalik, B. (2014). Variability in Software Systems – a Systematic Literature Review. *IEEE Transactions On Software Engineering*, 40(3), 282–306.

- [7] Narendra, N.C. & Ponnalagu, K. (2010). Towards a Variability Model for SOA-based Solutions, IEEE International Conference on Services Computing, Miami, USA, 562–569.
- [8] Sun, C., Rossing, R., Sinnema, M., Bulanov, P. & Aiello, M. (2010). Modeling and Managing the Variability of Web-Service-Based Systems. *Journal of Systems & Software*, 83(3), 502–516.
- [9] Mahdavi-Hezavehi, S., Galster, M. & Avgeriou, P. (2013). Variability in Quality Attributes of Service-Based Software Systems: A Systematic Literature Review, *Information and Software Technology*, 55(2), 320–343.
- [10] Alebrahim, A., Faßbender, S., Filipczyk, M. & Goedi, M. (2016). Variability for Qualities in Software Architecture. *SIGSOFT Software Engineering Notes*, 41(1), 32–35.
- [11] Weiss, D.M. & Lai, C.T.R. (1999). *Software product-line engineering: a family-based software development process*. Addison-Wesley.
- [12] Halmans, G. & Pohl, K. (2004). Communicating the variability of a software product family to customers. *Inform – Forsch und Entwicklung*, 18, 113–131.
- [13] Frankova, G., Yautsiukhin, A. & Seguran, M. (2007). From early requirements to business processes with service level agreements. University of Trento.
- [14] Frankova, G., Séguran, M., Gilcher, F., Trabelsi, S., Dörflinger, J. & Aiello, M. (2011). Deriving business processes with service level agreements from early requirements. *Journal of Systems and Software*, 84(8), 1351–1363.
- [15] Rosa, M. L., Aalst, W. M. V. D., Dumas, M. & Milani, F.P. (2017). Business process variability modeling: A survey. *ACM Computing Surveys (CSUR)*, 50 (1), 1–45.
- [16] Greenwood, P., Chitchyan, R., Ayed, D., Girard-Reydet, V., Fleurey, F., Dehlen, V. & Solberg, A. (2011). Modelling service requirements variability: The DiVA way. *Service Engineering*, 55–84.
- [17] ISO 9241-11:2018 Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts, 2018.
- [18] List, B. & Korherr, B. (2006). An evaluation of conceptual business process modelling languages. *Proceedings of the 2006 ACM Symposium on Applied Computing*.

Dialogue System Augmented with Commonsense Knowledge

Ilya Lasy

Faculty of Mathematics and Informatics, Vilnius University
Didlaukio st. 47, LT-08303 Vilnius
ilya.lasy@mif.stud.vu.lt

Virginijus Marcinkevičius

Institute of Data Science and Digital Technologies
Vilnius University, Akademijos st. 4, LT-08412 Vilnius
virginijus.marcinkevicius@mif.vu.lt

Summary. Building an open-domain dialog system is a challenging task in current research. In order to successfully maintain a conversation with human, a dialog system must develop many qualities: being engaging, empathetic, show a unique personality and having general knowledge about the world. Prior research has shown that it is possible to develop such chat-bot system that combines these features, but this work explores this problem further. Most state-of-the-art dialogue systems are guided by unstructured knowledge such as Wikipedia articles, but there is a lack of research on how structured knowledge bases can be used for open-domain dialogue generation. This work proposes usage of structured knowledge base ConceptNet for knowledge-grounded dialogue generation. Novel knowledge extraction algorithm is proposed which is then used to incorporate knowledge into existing dialogue datasets. Current state-of-the-art model BlenderBot is finetuned on new datasets which shows improvement in novelty of utterances generated by the model.

Keywords: Natural Language Generation, Dialogue System, Chat-bot, Knowledge Graph, Deep Learning

1 Introduction

Dialogue systems, or chatbots, are highly popular nowadays in various fields such as commerce, education, entertainment, finance, health, etc. Most chatbots are accessed on-line via website popups or through virtual assistants and can quickly provide answers for frequently asked questions and help navigate on a hosted web-site.

This kind of chatbots are domain-specific and are not able to maintain a conversation in open domain which is a natural requirement for such cases as virtual assistants (e.g. Amazon Alexa), AI politicians [11], mental health chatbots [21] or any kind of bot that involves continues conversation on general topics.

Current state-of-the-art open-domain chatbots are able to demonstrate personality [26], knowledge [5], empathy [15] and combine all these abilities together [17]. This work focuses on incorporating knowledge into a dialogue system that already has developed aforementioned qualities. In particular, structured knowledge base ConceptNet [20] is used for guiding dialogue generation. ConceptNet was already used in dialogue systems during previous research [24, 25, 27], but most of these researches are focused just on using knowledge in isolation rather than combining knowledge with other features.

This work introduces knowledge extraction algorithm which is used for adding knowledge to training dataset and to interlocutor messages during inference. BlenderBot [17] is used as baseline architecture for the model and is trained on knowledge augmented data. Such knowledge-enhanced version outperforms vanilla version of the model in terms of BLEU [13], ROUGE [10], diversity [9].

2 Methodology

2.1 Baseline

BlenderBot [17] model has been chosen as a baseline model for chatbot. Authors have two different versions of the model: generative and retrieve-and-refine [23]. Generative version showed better results, so they released this version. Therefore, BlenderBot will be referred to as its generative variant further in this paper.

BlenderBot is a seq2seq transformer [22] which main power is in the data it is fine-tuned on. Originally, it was pre-trained on 1.5B training examples from Reddit [1] and fine-tuned on the combination of following **datasets**: ConvAI2 [6], Empathetic Dialogues (ED) [15], Wizard Of Wikipedia (WoW) [5] and Blender Skill Talk (BST) [19]. In Section 3, these datasets will be referred to as "original".

2.2 Knowledge extraction and datasets augmentation

As Blenderbot is finetuned on these datasets without any explicit knowledge guidance, it can hallucinate knowledge that is implicitly saved in the model weights. To address this problem, this work proposes explicit knowledge guidance with knowledge extracted from ConceptNet [20]. ConceptNet is a knowledge graph that connects words and phrases of natural language (*terms*) with labeled edges (*relations*). Each $\langle term, relation, term \rangle$ triple is referred to as an *assertion*.

Each sample in aforementioned datasets is input message labeled with answer to that message. To incorporate knowledge in the dataset, each input message is appended with associated assertions which were extracted using custom knowledge extraction algorithm:

Algorithm 1 Retrieving assertions from message

```
Input: message, k  
Output: assertions  
1: for each sentence  $\in$  message do  
2:   Encode sentence into vector  
3:   for each token  $\in$  sentence do  
4:     Find all assertions for token in ConceptNet  
5:     Encode all assertions into vectors  
6:     Find cosine similarities between assertions vectors and sentence vector  
7:     Leave only top N similar assertions  
8:   end for  
9: end for
```

Steps of the developed algorithm are described below:

1. Given a dialogue message, it is splitted into sentences by sentence segmentation strategy. During dialogue it's not common to have complex sentence boundaries, that is why simple segmentation strategy that splits sentences by punctuation (./?) is used.
2. Each sentence is transformed into vector by sentence embedding model. Sentence embedding model is a neural network that converts sentences into vector representation in such a way that semantically similar sentences are close in vector space. It was decided to use Sentence-BERT model [16] for this step as it outperforms analogues models in this task.

3. Each sentence is tokenized, i.e. splitted into words or group of words. Group of words are more correct in cases when these words form an contextually meaningful expression, e.g. *“best man”*, *“flying colors”*, *“The Great Wall”*. In order to properly retrieve tokens, tokenizer algorithm and dependency parser [4] are used. Dependency parser builds a dependency tree which describes relationships between words in a sentence.
4. Each token is lemmatized and queried into the ConceptNet retrieving all assertions (<subject, relation, object>) connected to that token. These assertions compose main knowledge for each token, but there can be too much of them for inputting to the generative model. That is why it is necessary to filter them.
5. All assertions can be represented as small sentences (e.g. A net is used for catching fish), therefore they are vectorized using same sentence embedding model from step 2.
6. Cosine similarity score is calculated between sentence vector and all assertions vectors. All assertion vectors are ranked by similarity to sentence vector.
7. Only top k (where k is specified input parameter) similar assertions are added to the final assertions set.

Therefore, knowledge guided versions of four original datasets described in Section 2.1 were created using Algorithm 1. In Section 3, newly created datasets are marked as “with assertions”. Knowledge extraction is also used during real time conversation with new model. Assertions are extracted from user’s message and appended to input of the model, conditioning output not only on the user’s sentence, but on explicit knowledge. Current state-of-the-art model BlenderBot was finetuned on these new datasets to check hypothesis that general knowledge could improve performance of the dialog system.

2.3 Evaluation Metrics

Following evaluation metrics are used to evaluate the quality of generated responses: Perplexity (PPL) [18], BLEU [13], ROUGE [10] are used for measuring novelty, relevance and repetitiveness; Distinct-1, Distinct-2 [9] are used for diversity.

Perplexity explicitly measures the model’s ability to account for the syntactic structure of the dialogue (e.g. turn-taking) and the syntactic

structure of each utterance (e.g. punctuation marks). In dialogue, the distribution over the words in the next utterance is highly multi-modal, e.g. there are many possible answers, which makes perplexity particularly appropriate because it will always measure the probability of regenerating the exact reference utterance.

Distinct is an algorithm for evaluating the textual diversity of the generated text [9]. Distinct-n is calculated as the number of distinct n-grams divided either by total number of words across all generations (inter) or by number of words only within one sentence (intra). The larger the number of distinct n-grams, the higher the diversity of the generated text. This is useful in dialogue evaluation context as it can help to prove or reject the hypothesis that retrieved knowledge can help the model to be more diverse.

3 Experiments

BlenderBot model was released in 3 different sizes of parameters: 90M, 2.7B, 9.4B. Blenderbot implementation as well as all used datasets are provided by ParlAI framework [12]. Initial experiments are performed on the smallest model, but starting from Section 3.3, 2.7B version of the model will be used.

All algorithms and models used in experiments were implemented in Python programming language. Spacy [8] along with nltk [3] libraries were used for various text preprocessing operations: sentence segmentation, tokenization, dependency parsing, lemmatization. NetworkX [7] was used for performing operations with ConceptNet knowledge graph. All deep learning models are implemented with help of PyTorch [14].

90M model was fine-tuned using GeForce RTX 3070 GPU with 8 GB of video memory. 2.7B version of the model was trained using cluster with 2 V100 with total of 64 GB video memory.

3.1 Datasets performance comparison

Two fine-tuning sessions of the BlenderBot were ran during this initial experiment. First, it was necessary to reproduce results of the baseline paper[17] and finetune BlenderBot on original dataset described in Section 2.1. Second, BlenderBot was fine-tuned on newly created datasets with assertions specified in Section 2.2. Model was trained in multi-task fashion, meaning that during training there is an equal probability to have sample

from all of the used datasets (WoW, BST, ConvAI2) in a batch. During results estimation original validation split was used for each dataset. Results of the finetuning are provided in Table 1.

Table 1. Novelty (lower better) and Diversity (higher better) of Generative Blenderbot 90M trained on different datasets

Dataset	Novelty (↓)			Diversity (↑)	
	PPL	BLEU-1	ROUGE-1	InterDISTINCT-1	IntraDISTINCT-1
BST	16.1	0.1187	0.1654	0.0432	0.8263
BST (With assertions)	16.18	0.1201	0.1655	0.0432	0.8209
ConvAI2	12.66	0.1460	0.1819	0.0261	0.8486
ConvAI2 (With assertions)	13.34	0.1465	0.1818	0.0266	0.8363
Wizard of Wikipedia	18.72	0.1450	0.1931	0.0610	0.8322
Wizard of Wikipedia (With assertions)	17.23	0.1327	0.1768	0.0543	0.8424

Results show that there is no significant improvement on BST and ConvAI2 datasets, but perplexity and novelty are better when measured on Wizard of Wikipedia dataset. Reason for this can be due to the fact that WoW dataset contains a lot of factoids and having explicit knowledge in the input can help to model dependencies between utterance and general knowledge.

3.2 Increasing model size

Because of previous results, it was hypothesized that the size of the model (90M) does not allow to capture complex relationships between input data and extracted knowledge. It was decided to repeat previous experiment with 2.7B parameter model. Results provided in Table 2 show difference of BlenderBot sizes trained both on original datasets and datasets with assertions. Each metric value shown in the table is an average of value measured on all used datasets (e.g. perplexity is first measured on validation split of BST, ConvAI2, WoW, then averaged and show in the table).

Results show that increasing model size indeed increases difference between model trained on original data and model trained on knowledge enhanced data. BlenderBot 2.7B showed best results in both BLEU and ROUGE while fine-tuned on dataset with assertions.

Table 2. Comparison of different sizes of BlenderBot

Model	PPL	Novelty (↓)		Diversity (↑)	
		BLEU-1	ROUGE-1	InterDISTINCT-1	IntraDISTINCT-1
BlenderBot 90M	15.82	0.1365	0.1801	0.0434	0.8357
BlenderBot 90M (with assertions)	15.58	0.1331	0.1747	0.0413	0.8332
BlenderBot 2.7B	9.09	0.1474	0.1974	0.0295	0.8952
BlenderBot 2.7B (with assertions)	10.55	0.1279	0.1719	0.0264	0.9108

3.3 ConceptNet filtering

In order to improve quality of knowledge appended to datasets the knowledge base was filtered. ConceptNet contains a lot of infrequent relations which are hard to learn and often overspecific, and hence not useful for establishing high quality relations and paths between concepts. Therefore, a subset of the knowledge base that contains all assertions of the 13 most frequent relations is extracted: *RelatedTo*, *HasContext*, *IsA*, *FormOf*, *UsedFor*, *SimilarTo*, *AtLocation*, *HasSubevent*, *HasPrerequisite*, *CapableOf*, *Causes*, *MannerOf*, *PartOf*.

Now, when knowledge base contains small amount of possible relations, it is possible to convert each relation into a fixed special token (e.g. *RelatedTo*...) instead of treating it as a pure text. During this experiment original datasets were augmented in a similar way as described in Section 2.2, but filtered version of ConceptNet is used. BlenderBot was fine-tuned on a new dataset containing these special tokens. Performance of BlenderBot fine-tuned on original dataset, on unfiltered datasets with assertions and on datasets with special tokens are compared in the Table 3.

Table 3. Comparison of different versions of BlenderBot 2.7B

Model	PPL	Novelty (↓)		Diversity (↑)	
		BLEU-1	ROUGE-1	InterDISTINCT-1	IntraDISTINCT-1
BlenderBot 2.7B (original)	9.09	0.1474	0.1974	0.0295	0.8952
BlenderBot 2.7B (with assertions)	10.55	0.1279	0.1719	0.264	0.9108
BlenderBot 2.7B (with special tokens)	10.31	0.1276	0.1717	0.0265	0.9007

3.4 Dialogue Generation

Performance of the proposed method was measured with automated metrics, but also it was possible to see dialogues generated by the model via self-chat technique: during self-chat, two independent objects of the same model were created and tasked to generate responses to one another. Examples of dialogues generated by model trained on original data and by model trained on data with assertions are shown in Figures 1 and 2 respectively. Both dialogues have the same initial sentence (“Hi, how are you?”) and persona context (initial information about the bot: “I am a senior citizen, I like to read”) in order to be able to directly compare generated utterances.

```
Bot #1: Hi, how are you?
Bot #2: I'm doing well. How about you? What do you like to do in your free time?
-----
Bot #1: I'm good. Reading is one of my favorite things to do. Do you like reading?
Bot #2: Yes, I do. I like to read fiction and non-fiction. What is your favorite genre?
-----
Bot #1: I like historical fiction. It's a mix of real events and fictional ones.
Bot #2: What is the name of the documentary? I would love to watch it. I have never heard of it.
-----
```

Figure 1. Example self-chat of BlenderBot 2.7 fine-tuned on original data

```
Bot #1: Hi, how are you?
Bot #2: I'm doing well. How about you? What do you like to do in your free time?
-----
Bot #1: I'm good. Reading is one of my favorite things to do. Do you like reading?
Bot #2: Yes, I do. I like to read fiction and non-fiction. What is your favorite genre?
-----
Bot #1: I'm a big fan of science fiction. My favorite author is Isaac Asimov.
Bot #2: I have never read any of his books, but I have heard of him. He was born in 1948, right?
-----
```

Figure 2. Example self-chat of BlenderBot 2.7 fine-tuned on data with assertions

These examples are cherry-picked so they cannot be a proper indication of the results, but as it can be seen from the comparison of these dialogues, second (Figure 2) dialogue seems more meaningful. During first dialogue, bot#2 asks “What is the name of the documentary?” after the sentence “I like historical fiction” which seems like not an appropriate reaction.

4 Conclusion

Dialogue systems that are able to use knowledge about the world while having another conversational qualities are studied during this work. A

novel knowledge extraction algorithm was developed in order to augment existing dialogue datasets and fine-tune state-of-the-art chatbot model BlenderBot. During a comparison of original model and proposed model, results showed that model trained on knowledge augmented dataset tends to generate more novel (0.128 in BLEU, 0.172 in ROUGE-1) responses. Although there are no significant changes in diversity (0.91 in IntraDISINCT for augmented BlenderBot vs 0.89 for original BlenderBot). Current results can be improved by replacing knowledge extraction algorithm with more complex approach (e.g. predicting knowledge paths [2]), increasing model size or changing model architecture [25].

References

- [1] *The Pushshift Reddit Dataset*, Jan. 2020. Zenodo. doi: 10.5281/zenodo.3608135. URL <https://doi.org/10.5281/zenodo.3608135>.
- [2] M. Becker, K. Korfhage, D. Paul, and A. Frank. CO-NNECT: A framework for revealing commonsense knowledge paths as explicitations of implicit knowledge in texts. *CoRR*, abs/2105.03157, 2021. URL <https://arxiv.org/abs/2105.03157>.
- [3] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc., 2009.
- [4] D. Chen and C. Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1082. URL <https://aclanthology.org/D14-1082>.
- [5] E. Dinan, S. Roller, K. Shuster, A. Fan, M. Auli, and J. Weston. Wizard of wikipedia: Knowledge-powered conversational agents, 2018. URL <https://arxiv.org/abs/1811.01241>.
- [6] E. Dinan, V. Logacheva, V. Malykh, A. H. Miller, K. Shuster, J. Urbanek, D. Kiela, A. Szlam, I. Serban, R. Lowe, S. Prabhunoye, A. W. Black, A. I. Rudnicky, J. Williams, J. Pineau, M. S. Burtsev, and J. Weston. The second conversational intelligence challenge (conva2). *CoRR*, abs/1902.00098, 2019. URL <http://arxiv.org/abs/1902.00098>.
- [7] A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring network structure, dynamics, and function using networkx. In G. Varoquaux, T. Vaught, and J. Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.
- [8] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd. spaCy: Industrial-strength Natural Language Processing in Python. 2020. doi: 10.5281/zenodo.1212303.
- [9] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan. A diversity-promoting objective function for neural conversation models. *CoRR*, abs/1510.03055, 2015. URL <http://arxiv.org/abs/1510.03055>.
- [10] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/W04-1013>.
- [11] M. Matsuda. Politics 2028: Why artificial intelligence will replace politicians. <https://www.is.gd/oeQLi7>, 2018. accessed 2021-01-12.

- [12] A. H. Miller, W. Feng, A. Fisch, J. Lu, D. Batra, A. Bordes, D. Parikh, and J. Weston. Parlai: A dialog research software platform. *arXiv preprint arXiv:1705.06476*, 2017.
- [13] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. Bleu: a method for automatic evaluation of machine translation. page 8, 10 2002. doi: 10.3115/1073083.1073135.
- [14] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chil- amkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high- performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché- Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [15] H. Rashkin, E. M. Smith, M. Li, and Y. Boureau. I know the feeling: Learning to converse with empathy. *CoRR*, abs/1811.00207, 2018. URL <http://arxiv.org/abs/1811.00207>.
- [16] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. URL <https://arxiv.org/abs/1908.10084>.
- [17] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, K. Shuster, E. M. Smith, Y.-L. Boureau, and J. Weston. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*, 2020.
- [18] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models, 2016.
- [19] E. M. Smith, M. Williamson, K. Shuster, J. Weston, and Y. Boureau. Can you put it all together: Evaluating conversational agents' ability to blend skills. *CoRR*, abs/2004.08449, 2020. URL <https://arxiv.org/abs/2004.08449>.
- [20] R. Speer, J. Chin, and C. Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. *CoRR*, abs/1612.03975, 2016. URL <http://arxiv.org/abs/1612.03975>.
- [21] A. N. Vaidyam, H. Wisniewski, J. D. Halamka, M. S. Kashavan, and J. B. Torous. Chatbots and conversational agents in mental health: A review of the psychiatric landscape. *Can. J. Psychiatry*, 64(7): 456–464, July 2019.
- [22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017.
- [23] J. Weston, E. Dinan, and A. Miller. Retrieve and refine: Improved sequence generation models for dia- logue. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search- Oriented Conversational AI*, pages 87–92, Brussels, Belgium, Oct. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5713. URL <https://www.aclweb.org/anthology/W18-5713>.
- [24] T. Young, E. Cambria, I. Chaturvedi, M. Huang, H. Zhou, and S. Biswas. Augmenting end-to-end dialog systems with commonsense knowledge, 2018.
- [25] H. Zhang, Z. Liu, C. Xiong, and Z. Liu. Conversation generation with concept flow. *CoRR*, abs/1911.02707, 2019. URL <http://arxiv.org/abs/1911.02707>.
- [26] S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston. Personalizing dialogue agents: I have a dog, do you have pets too?, 2018.
- [27] H. Zhou, T. Young, M. Huang, H. Zhao, J. Xu, and X. Zhu. Commonsense knowledge aware conversation generation with graph attention. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4623–4629. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/643. URL <https://doi.org/10.24963/ijcai.2018/643>.

3D paviršiaus formavimo 2D vaizdų pagrindu algoritmų tyrimas

Arnas Nakrošis, Andrius Kriščiūnas

Kauno technologijos universitetas,
K. Donelaičio g. 73, Kaunas
arnas.nakrosis@ktu.lt, andrius.krisciunas@ktu.lt

Santrauka. Tobulėjant technologijoms pramonės ir viešajame sektoriuose informacijos atvaizdavimui vis dažniau yra naudojami trimačiai vaizdai, leidžiantys darbuotojams lengviau suprasti gautus rezultatus ar surinktą informaciją. Nors šiuo metu jau egzistuoja algoritmai leidžiantys standartinėmis priemonėmis surinktų 2D vaizdų pagrindu formuoti trimačius vaizdus, tačiau tokių algoritmų pateikiamų rezultatų kokybę dar vis smarkiai atsilieka nuo rinkos poreikio. Šio tyrimo metu buvo atliekama esamų algoritmų analizė siekiant pasiūlyti sprendimą, kuris leistų pasiekti geresnius rezultatus lyginant su literatūroje nagrinėjama atvejais. Rezultate autorių pasiūlyta algoritmu kombinacija leidžia suformuoti 3D paviršiu remiantis santykinai nedideliu 2D vaizdų kiekiu.

Raktiniai žodžiai: Taškinės nuotraukos, 3D paviršius, 3D taškų debesis, Poissono algoritmas, persidengiančios nuotraukos.

1 Įvadas

Tobulėjant technologijoms yra siekiama atrasti algoritmus, leidžiančius skaitmeninti objektus naudojant taškinius vaizdus. Objektų skaitmeninimas yra naudingas siekiant perteikti didesnę kiekį informacijos naudotojams viešajame ar pramonės sektoriuose. 2021 metais atlikti rinkos tyrimai pastebi, kad per ateinančius metus yra numatomas 20,9% metinis augimo tempas net rinką paveikus globaliems veiksniams [1], [2]. Atsižvelgiant į šiuos rinkos tyrimus yra matomas technologijos, kuri vis dar sparčiai tobulėja, poreikis. Siekiant metodologijas pritaikyti sparčiai besiplečiančiai rinkai ir spartėjančiam gamybos bei paslaugų tiekimo greičiui, tyrėjai nagrinėja skirtingas galimybes technologijų tobulinimui. 3D paviršiaus formavimo 2D vaizdų pagrindu generavimo algoritmus galima skirstyti į dvi pagrindines grupes: matematinėmis savybėmis ir mašininio mokymo algoritmais paremtus algoritmus.

Darbuose [3], [4] matematinėmis savybėmis paremti algoritmai naudoja 2.5D nuotraukas (naudojamas ~40 nuotraukų to paties objekto rinkinys),

kuriose kartu su vaizdine informacija yra pateikiamos gylio reikšmės bei parametras *delta*. Šiuo atveju algoritmas, pagal pateiktą vaizdų rinkinį erdviųjų vektorių pagalba apskaičiuoja atstumo reikšmes siekiant tolimesniuose žingsniuose sumažinti atminties sunaudojimą, bei skaičiavimams reikalingų resursų kiekį. Delta parametru valdomas erdviųjų vektorių sudarymas, kuris nulemia arčiau esančių detalių svarbą ir atstumo užtikrintumą, lyginant su geometriškai tolimesnėms detalėms. Taip rezultate algoritmas erdviiniams taškams priskiria ir užtikrintumo reikšmę, kuri tostant mažėja, bei kuomet elementas nėra matomas iš nagrinėjamos perspektyvos yra artima nuliui. Kitos studijos autoriai [5] pristatė algoritmą, kuris siekia sujungti privalumus iš retų 3D tinklus formuojančių algoritmų [6] ir tankius 3D tinklus formuojančių algoritmų [7]. Tai buvo pasiekta naudojant 2D nuotraukas (daugiau nei 26) ir 3D taškus. Pirmiausia, šis algoritmas, iš gautų nuotraukų išskiria atraminius taškus, kurie yra retai išdėstyti nuotraukų erdvėje, siekiant kad tie taškai turėtų kiek įmanoma didesnę reikšmę žiūrint į objektą iš skirtingų perspektyvų (nuotraukų). Taškų palyginimui skirtingose nuotraukose yra naudojamas ZNCC (angl. Zero – Mean Normalized Cross Correlation) metodas bei kryžminė patikra [8]. Taip yra sudaromas svarbių taškų sąrašas, kuris yra papildomas interpoliaciniais taškais. Šiuo procesu yra siekiama sutankinti formuojamą tinklą, gerinant paviršiaus tikslumą [9].

Atskira algoritmų grupė yra mašininio mokymusi paremti algoritmai. Darbe [10] autoriai pasiūlė algoritmą, kuris naudojant neuroninius tinklus sudarytų 3D paviršių iš retų magnetinio rezonanso medicininių vaizdų (toliau MRI) rinkinių. Šiam tikslui pasiekti yra naudojamos trys, skirtingų perspektyvų pjūvinės dalinai ortogonalios nuotraukos, kai MRI yra apdorojami gilioju konvoliuciniu neuroniniu tinklu (toliau CNN) [11] ir visų trijų MRI rezultatai yra bendrai apdorojami dviem pilnai sujungtais neuroninio tinklo sluoksniais. Gauti neuroninio tinklo rezultatai yra apdorojami pagrindinių komponentų analize (toliau PCA) [12], ir suformuojamas paviršius [13]. CNN yra apmokomas siekiant prognozuoti vidurki, poslinkį ir kovariacijos matricą. Apmokymo tikslo funkciją sudaro dvi pagrindinės dalys: paviršiaus prognozės patikimumas, siekiant maksimizuoti sąlyginę tikimybę, bei atstumo tarp prognozuoto ir tikrojo paviršiaus, matuojamo Kullback-Leibler pasiskirstymu [14]. Atlikus MRI apdorojimą naudojant CNN algoritmas apskaičiuoja tikimybes paviršiui, naudojant PCA įverčius. CNN yra sudarytas iš trijų šakų, skirtų kiekvieno MRI apdorojimui atskirai. Kiekvieną šaką sudaro

9 konvoliuciniai sluoksniai ir trys sutelkimo pagal maksimumą (angl. max pooling) sluoksniai. CNN šakų rezultatai yra sujungiami naudojant du neuroninio tinklo sluoksnius. Šis algoritmas yra pranašesnis už matematinius algoritmus tuo, kad formavimui naudoja nedidelį kiekį nuotraukų (priklausomai nuo objekto sudėtingumo reikia nuo 3 iki 9 nuotraukų) [10]. Tačiau norint apmokyti šį neuroninį tinklą yra reikalingas didelis kiekis apmokymui skirtų nuotraukų ir gautas apmokytas neuroninis tinklas nėra universalus (sunkus pritaikymas skirtingo tipo objektų formavimui) [10].

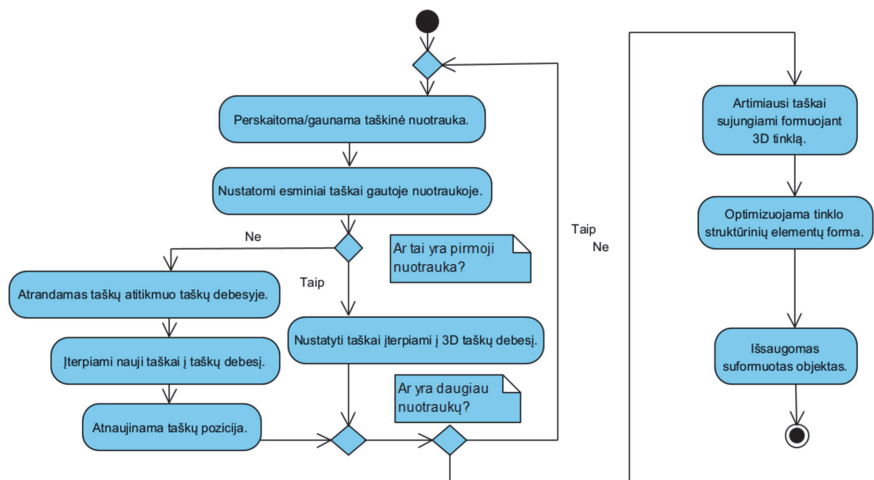
Apibendrinant 3D paviršių analizę galima teigti, kad dauguma algoritmų naudoja didelį kiekį įvesties nuotraukų, papildomų įvesčių arba yra pritaikyti tik vienai objektų grupei, kas smarkiai apriboja minimų algoritmų taikymą praktikoje. Tai lemia esamų sprendimų tobulinimo poreikį ir paklausą, kas tiesiogiai seka iš einamuoju momentu šioje srityje atliekamų tyrimų gausos. Šiame darbe autoriai nagrinėja matematinėmis savybėmis paremtą algoritmų kombinacijos sprendimą, kuri leidžia įvairių sričių objektams sudaryti 3D paviršius naudojant nedidelį kiekį taškių nuotraukų (priklausomai nuo formuojamo objekto formos sudėtingumo), svyruoja nuo 8 iki 16 vaizdų.

2 Glotnaus 3D paviršiaus formavimo algoritmas

Darbe sudaryto algoritmo skirto glotnaus 3D paviršiaus formavimui nedidelio kiekio įvesties vaizdų pagrindu, schema pateikta 1 pav. Šis algoritmas gali būti pritaikytas skirtingų formų objektams ir susideda iš trijų pagrindinių dalių: 3D taškų debesies formavimas, 3D debesies taškų optimizavimas, debesies sujungimas į 3D paviršių. 3D taškų debesies formavimas yra paremtas struktūros iš judėsio (angl. structure from motion) [15] algoritmu, kuris pasikartojančius kritinius taškus atrenka iš persidengiančių nuotraukų. Suformuotas taškų debesis yra optimizuojamas pašalinant taškus, esančius šalia kitų taškų ir esančius vienoje tiesėje su daugiau nei dviem taškais. Atlikus debesies optimizavimą, yra suformuojamas trimatis paviršius sujungiant taškus į trikampius naudojant Poissono [16] algoritmą.

3 Eksperimentiniai tyrimai

Sudaryto algoritmo analizei buvo pasirinktas formuoti sudėtingos ir paprastos struktūrų objektai (2 pav.).





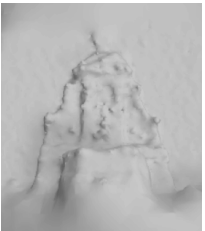
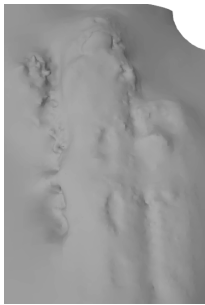


1 pav. Duomenų rinkinio pavyzdys



2 pav. Duomenų rinkinių pavyzdžiai

Pasirinkto objekto duomenų rinkinius sudarė 11 vaizdų aibės, kurių pagrindu buvo suformuotas 3D paviršius. 1 lentelėje pateikti 3D paviršiaus sudarymo rezultatai (laikas, ir nustatytų esminių taškų skaičius) taikant skirtingus algoritmus kartu pateikiant ir sudaryto paviršiaus vaizdo vizualizaciją.

1 lentelė. Algoritmų palyginimo lentelė, kai objektams sudaryti naudojama 11 nuotraukų/

Pavadinimas	Autorių siūlomas algoritmas		Žygiuojančių kubų algoritmas		TV-L	
Objektas 1						
	Formavimo laikas (s)	Esminių taškų skaičius	Formavimo laikas (s)	Esminių taškų skaičius	Formavimo laikas (s)	Esminių taškų skaičius
	62,32	8381	74,97	9345	69,02	7675
Objektas 2						
	Formavimo laikas (s)	Esminių taškų skaičius	Formavimo laikas (s)	Esminių taškų skaičius	Formavimo laikas (s)	Esminių taškų skaičius
	187,71	4738	192,56	5526	196,74	4257

Iš lentelėje pateiktų rezultatų yra matoma, kad autorių siūlomas algoritmų derinys vertinant vizualiai labiausiai atitinka objektus pateiktus 2 pav., bei suformuoja paviršių greičiausiai (5,66 s. vienos pirmo objekto nuotraukos apdorojimui) iš lygintų algoritmų. Antras objektas formuojamas beveik tris kartus ilgiau (17,06 s vienos nuotraukos apdorojimui), tačiau tokie rezultatai gaunami dėl to, kad nuotraukų dimensijos yra ~3 kartus didesnės.

4 Išvados

Darbe išanalizavus matematinėmis savybėmis bei mašininio mokymu paremtus 3D paviršiaus formavimo algoritmus, buvo pasirinkta matematinėmis savybėmis paremtų algoritmų klasė, ir apjungus struktūros iš judėsio ir Poissono algoritmus, pasiūlytas sprendimas kuris yra pritaikomas kiek įmanoma platesniame skirtingų objektų intervale naudojant santykinai nedidelį taškinių nuotraukų įvesties kiekį. Sudarytas algoritmas suformuoja glotnų paviršių naudojant nuo 6 iki 16 nuotraukų priklausomai nuo formuojamo objekto formos sudėtingumo. Darbe pateiktiems duomenų rinkinių pavyzdžiui algoritmas suformuoja trimatį paviršių vidutiniškai vienos nuotraukos apdorojimui užtrukdamas nuo 5,66 s iki 17,06 s priklausomai nuo naudojamų nuotraukų dimensijų.

Literatūra

- [1] "3D Mapping and Modeling Market Trends, Growth, Industry Analysis and Forecast 2022-2027." <https://www.imarcgroup.com/3d-mapping-modeling-market> (accessed Apr. 15, 2022).
- [2] "3d Mapping and 3d Modelling Market | 2022 - 27 | Industry Share, Size, Growth - Mordor Intelligence." <https://www.mordorintelligence.com/industry-reports/3d-mapping-and-3d-modelling> (accessed Apr. 15, 2022).
- [3] S. I. Young, D. B. Lindell, B. Girod, D. Taubman, and G. Wetzstein, "Non-Line-of-Sight Surface Reconstruction Using the Directional Light-Cone Transform," 2020, pp. 1407–1416. Accessed: Apr. 15, 2022. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Young_Non-Line-of-Sight_Surface_Reconstruction_Using_the_Directional_Light-Cone_Transform_CVPR_2020_paper.html
- [4] Curless B. and M. Levoy, "A Volumetric Method for Building Complex Models from Range Images." <https://graphics.stanford.edu/papers/volrange/> (accessed Jan. 13, 2021).
- [5] G. Zampokas, K. Tsiolis, G. Peleka, I. Mariolis, S. Malasiotis, and D. Tzovaras, "Real-Time 3D Reconstruction in Minimally Invasive Surgery with Quasi-Dense Matching," in *2018 IEEE International Conference on Imaging Systems and Techniques (IST)*, Oct. 2018, pp. 1–6. doi: 10.1109/IST.2018.8577106.
- [6] M. Kulawiak, "A Cost-Effective Method for Reconstructing City-Building 3D Models from Sparse Lidar Point Clouds," *Remote Sens.*, vol. 14, no. 5, Art. no. 5, Jan. 2022, doi: 10.3390/rs14051278.
- [7] W. Zeng, W. Ouyang, P. Luo, W. Liu, and X. Wang, "3D Human Mesh Regression With Dense Correspondence," 2020, pp. 7054–7063. Accessed: Apr. 15, 2022. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Zeng_3D_Human_Mesh_Regression_With_Dense_Correspondence_CVPR_2020_paper.html
- [8] J.-C. Yoo and T. H. Han, "Fast Normalized Cross-Correlation," *Circuits Syst. Signal Process.*, vol. 28, no. 6, p. 819, Aug. 2009, doi: 10.1007/s00034-009-9130-7.

- [9] P. Henderson, V. Tsiminaki, and C. H. Lampert, "Leveraging 2D Data to Learn Textured 3D Mesh Generation," 2020, pp. 7498–7507. Accessed: Apr. 15, 2022. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Henderson_Leveraging_2D_Data_to_Learn_Textured_3D_Mesh_Generation_CVPR_2020_paper.html
- [10] K. Tóthová *et al.*, "Probabilistic 3D Surface Reconstruction from Sparse MRI Information," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, Cham, 2020, pp. 813–823. doi: 10.1007/978-3-030-59710-8_79.
- [11] P. Dong, H. Zhang, G. Y. Li, I. S. Gaspar, and N. NaderiAlizadeh, "Deep CNN-Based Channel Estimation for mmWave Massive MIMO Systems," *IEEE J. Sel. Top. Signal Process.*, vol. 13, no. 5, pp. 989–1000, Sep. 2019, doi: 10.1109/JSTSP.2019.2925975.
- [12] B. M. S. Hasan and A. M. Abdulazeez, "A Review of Principal Component Analysis Algorithm for Dimensionality Reduction," *J. Soft Comput. Data Min.*, vol. 2, no. 1, Art. no. 1, Apr. 2021, Accessed: Apr. 15, 2022. [Online]. Available: <https://publisher.uthm.edu.my/ojs/index.php/jscdm/article/view/8032>
- [13] K. Tóthová *et al.*, "Uncertainty Quantification in CNN-Based Surface Prediction Using Shape Priors," in *Shape in Medical Imaging*, Cham, 2018, pp. 300–310. doi: 10.1007/978-3-030-04747-4_28.
- [14] Y. Huang, Y. Zhang, and J. A. Chambers, "A Novel Kullback–Leibler Divergence Minimization-Based Adaptive Student's t-Filter," *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5417–5432, Oct. 2019, doi: 10.1109/TSP.2019.2939079.
- [15] M. Cali and R. Ambu, "Advanced 3D Photogrammetric Surface Reconstruction of Extensive Objects by UAV Camera Image Acquisition," *Sensors*, vol. 18, no. 9, Art. no. 9, Sep. 2018, doi: 10.3390/s18092815.
- [16] H. Fang and F. Lafarge, "Connect-and-Slice: An Hybrid Approach for Reconstructing 3D Objects," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 13487–13495. doi: 10.1109/CVPR42600.2020.01350.

Įmonių bankroto prognozavimas naudojant gilųjį mokymą

Ieva Rizgeliienė

Kauno technologijos universitetas,
K. Donelaičio g. 73, Kaunas
paulaviciute.ieva@gmail.com

Santrauka. Šiame straipsnyje pristatomi sukurti giliojo mokymosi modeliai skirti įmonių bankroto prognozavimui. Tyrimo metu, naudojant Lietuvos įmonių duomenis, sukurti du modeliai: daugiasluoksnis perceptronas ir konvoliucinis neuroninis tinklas. Modelių apmokymui ir hiperparametrų validavimui, buvo naudojami subalansuoti duomenų poaibiai. Siekiant įvertinti modelių gebėjimą atskirti bankroto atvejus išbalansuotoje duomenų aibėje, modelių testavimui buvo naudojamas poaibis, kuriame buvo išlaikytas pradinio duomenų rinkinio klasių disbalansas. Gauti rezultatai parodė, kad sukurti giliojo mokymosi modeliai atpažįsta bankroto atvejus duomenų aibėje su dideliu klasių disbalansu.

Raktiniai žodžiai: gilusis mokymasis, dirbtiniai neuroniniai tinklai, konvoliucinis neuroninis tinklas, bankroto prognozavimas, daugiasluoksnis perceptronas, klasifikavimo modelis.

1 Įvadas

Įmonių bankrotas – tai neišvengiamas ekonomikos reiškinys. Įmonės bankroto prognozė gali padėti identifikuoti įmones, susiduriančias su sunkumais, o tai gali užkirsti kelią įmonės bankrotui arba leisti tinkamai tam pasiruošti, taip sumažinant galimus nuostolius.

Ekonomikos inžinieriai bankrotą pradėjo nagrinėti 1930 m. Pirmiausiai buvo susikoncentruota į bankrutavusių ir nebankrutavusių įmonių finansinių rodiklių palyginimą. 1968 m. Edward'as I. Altman'as pirmą kartą bankrotui prognozuoti pritaikė daugiamatės diskriminantinės analizės modelį [1]. Logistinė regresija pirmą kartą bankrotui prognozuoti buvo pritaikyta 1980 m. Ohlson'o [2]. Gerokai vėliau (21a.) buvo pritaikyti tokie modeliai kaip pagrindinių komponentų analizė, sprendimų medžiai ir atraminių vektorių klasifikatorius. Giliojo mokymosi algoritmai vis dar yra retesnis pasirinkimas bankroto prognozavimo uždaviniui spręsti, lyginant su kitais klasifikavimo metodais.

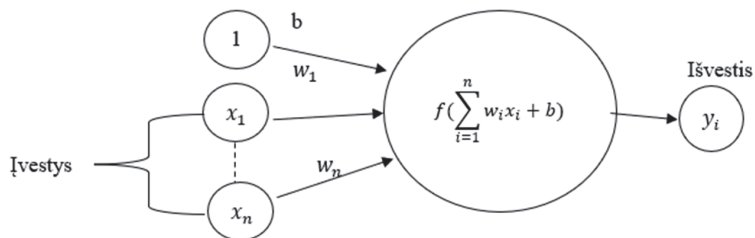
Gilūs mokymasis kiekviename sluoksnyje gali išmokti vis abstraktesnius duomenų bruožus, todėl parinkus tinkamą neuroninio tinklo architektūrą, galima gauti pakankamai tikslius prognozės rezultatus ir pranokti klasikinius bankroto prognozei taikomus algoritmus. Tyrėjai, bankroto prognozavimui pritaikę tiesioginio sklidimo neuroninius tinklus, pastebėjo, kad šio metodo klasifikavimo rezultatai rodo gerus rezultatus ir prilygsta klasikiniams klasifikavimo algoritmams [3]. Konvoliuciniai neuroniniai tinklai yra plačiai taikomi vaizdų atpažinimo uždaviniams spręsti, tačiau jau yra atlikta tyrimų ir įmonių bankroto prognozavimo tematika, kuriuose siūloma įmonių finansinius rodiklius išreikšti nespaltotų vaizdų pavidalu, pritaikius minėtą duomenų transformaciją, buvo pastebėta, kad toks metodas gali pranokti daugelį klasifikavimo algoritmų [4].

Šio tyrimo tikslas, naudojant giliuosius dirbtinius neuroninius tinklus, sukurti įmonių bankroto prognozės modelius, kurie prognozuotų įmonių bankrotą likus vienam mėnesiui iki šio įvykio. Tyrimo metu buvo išbandomi dviejų rūšių neuroniniai tinklai: daugiasluoksnis perceptronas ir konvoliucinis neuroninis tinklas. Kuriant neuroninių tinklų modelius, geriausios tinklo architektūros ir geriausių hiperparametrų paieškai buvo taikomas Bajeso optimizavimas, išbandomos 8 skirtingos aktyvacijos funkcijos ir 7 skirtingi optimizavimo algoritmai. Tyrimo metu naudojama atvirojo kodo programinės įrangos biblioteka TensorFlow ir Python programavimo kalba.

2 Metodai

2.1 Dirbtiniai neuroniniai tinklai

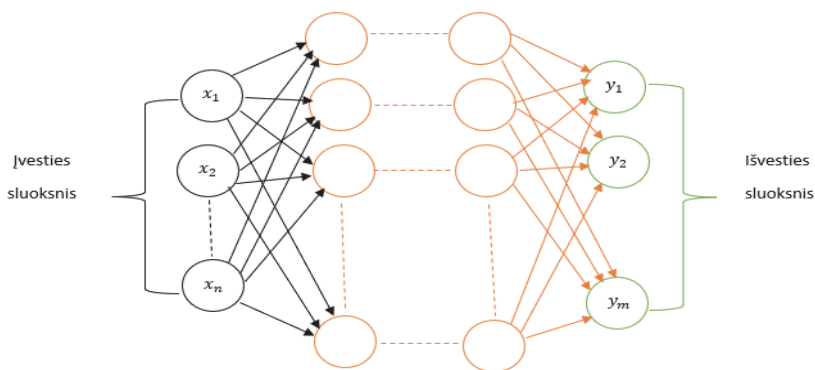
Dirbtiniai neuroniniai tinklai – tai informacijos apdorojimo sistemos, paremtos biologinio neurono veikimo principu. Kaip ir žmogaus smegenų atveju, dirbtiniai neuroniniai tinklai yra pagrįsti tarpusavyje sujungtais neuronais. Kiekvienai jungčiai priskirti koeficientai, vadinami neuronų svoriais. Į neuroninį tinklą įeinančios jungtys vadinamos įvestimis, išeinančios – išvestimis. Toliau pateikiama dirbtinio neurono schema.



1 pav. Dirbtinio neurono schema [5]

2.1.1 Tiesioginio sklidimo neuroniniai tinklai

Tiesioginio sklidimo neuroninio tinklo atveju, neuronai yra grupuojami į sluoksnius, o sluoksniai sujungiami taip, kad tinkle nesusidarytų ciklai. Šių neuroninių tinklų atveju, kiekvienais dirbtinis neuronas iš esamo sluoksnio yra sujungtas su kiekvienu dirbtiniu neuronu iš praeities sluoksnio. Šiame tinkle duomenys negali būti siunčiami į praeitus sluoksnius. Tiesioginio sklidimo neuroniniai tinklai, vadinami giliaisiais, tuomet, kai turi daugiau nei vieną paslėptąjį sluoksnį. Šiame tyrime yra išbandomi dviejų rūšių giliaji tiesioginio sklidimo neuroninio tinklai: daugiasluoksnis perceptronas ir konvoliucinis neuroninis tinklas, 2 pav. vaizduojamas daugiasluoksnio perceptrono modelis.



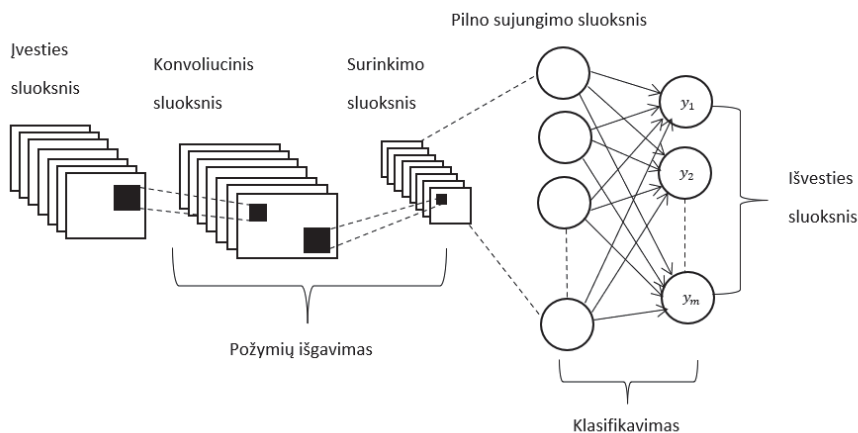
2 pav. Daugiasluoksnio perceptrono modelis [6]

2.1.2 Konvoliuciniai neuroniniai tinklai

Konvoliuciniai neuroniniai tinklai, tai tiesioginio sklidimo neuroniniai tinklai, turintys konvoliucinius sluoksnius. Konvoliuciniai neuroniniai tinklai yra plačiai taikomi vaizdams analizuoti. Trys pagrindiniai konvoliucinių neuroninių tinklų sluoksniai:

- konvoliucinis sluoksnis,
- surinkimo sluoksnis (angl. *pooling*),
- pilnai sujungtas sluoksnis.

Konvoliuciniuose neuroniniuose tinkluose, neuronai išsidėstę trijose dimensijose: plotis, aukštis, gylis. Toliau pateikiama konvoliucinio neuroninio tinklo schema.



3 pav. Konvoliucinio neuroninio tinklo schema [6]

Trumpai aptarsime 3 pav. schemoje vaizduojamus neuroninio tinklo sluoksnius. Į neuroninį tinklą perduodamos įvestys, turi trimatį pavidalą. Pagrindinis konvoliucinio sluoksnio tikslas, naudojant apmokomus filtrus (angl. *kernels*), išgauti duomenų požymius. Surinkimo sluoksnyje yra sumažinamas įvesčių dydis. Pilno sujungimo sluoksnyje, gauti duomenys, konvertuojami į vienmatę erdvę, kitaip tariant šiam sluoksniui perduodamos įvestys yra paverčiamos vektoriumi. Tuomet gautoms reikšmėms pritaikoma aktyvacijos funkcija. Išvesties sluoksnyje, pritaikius pasirinktą aktyvacijos funkciją, gaunamos išvestys. Gauti rezultatai yra suklasifikuojami.

2.1.3 Aktyvacijos funkcijos

Daugiasluoksnių perceptronų modeliuose, aktyvacijos funkcija, naudojama kiekviename sluoksnyje, tuo tarpu konvoliucinių neuroninių tinklų atveju, aktyvacijos funkcija naudojama konvoliuciniuose ir pilno sujungimo sluoksniuose. Šiame tyrime yra išbandomos šios aktyvacijos funkcijos:

1. ReLU (angl. *Rectified linear unit*) funkcija, apibrėžiama: $f(x) = \max(0, x)$.
2. Softsign funkcija, apibrėžiama: $f(x) = \frac{x}{1+|x|}$.
3. Softplus funkcija, apibrėžiama: $f(x) = \ln(1 + e^x)$.
4. Leaky ReLU (angl. *Leaky Rectified Linear unit*) funkcija, apibrėžiama: $f(x) = \max(0.1 \cdot x, x)$.
5. Sigmoidinė funkcija, apibrėžiama: $f(x) = \frac{1}{1+e^{-x}}$.

6. Hiperbolinio tangento funkcija, apibrėžiama: $\tan(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.

7. Elu (angl. Exponential Linear Unit) funkcija, apibrėžiama:

$$f(x) = \begin{cases} x & x > 0 \\ \alpha(e^x - 1) & x \leq 0 \end{cases}$$

Čia $\alpha < 0$. α – funkcijos hiperparametras, kuris kontroliuoja neigiamas įvestis. Ši funkcija mažina nykstančio gradiento efektą, taip pagreitinama giliųjų neuroninių tinklų mokymąsi ir gali užtikrinti didesnį klasifikavimo tikslumą [7].

8. Selu (angl. Scaled Exponential Linear Unit) funkcija, apibrėžiama:

$$f(x) = \lambda \begin{cases} x & x > 0 \\ \alpha e^x - \alpha & x \leq 0 \end{cases}$$

Ši funkcija, gali padėti gauti išvestis, pasiskirsčiusias pagal normalųji skirstinį [8]. Čia λ – standartizavimo konstanta, ši konstanta yra lygi 1.0507, α taip yra apibrėžta ir yra lygi 1.6732.

2.1.4 Dirbtinių neuroninių tinklų apmokymas

Daugiasluoksniui perceptronui, apmokymo metu, keičiami jungčių svoriai, konvoliuciniam neuroniniam tinklui, keičiami jungčių ir filtrų svoriai. Keičiant svorius, naudojamas atgalinės sklaidos metodas [9]. Šio metodu metu, pirmiausiai atsitiktinai sugeneruojami viso tinklo svoriai, tuomet iš apmokymo duomenų imties, parenkamas vienas elementas, kuris pateikiamas neuroniniam tinklui kaip įvesties sluoksnis. Atlikus neuroninio tinklo skaičiavimus, gaunamas išvesties sluoksnis, su pasirinkta klaidos funkcija apskaičiuojama klaidos reikšmė tarp gauto ir norimo rezultato. Tuomet, pasluoksniui, pradedant nuo išvesties sluoksnio, grįžtama atgal ir naudojant atgalinės sklaidos metodą, apskaičiuojami lokalūs gradientai. Pasirinktu optimizavimo metodu, ieškoma tikslo funkcijos optimumo – šiuo atveju, klaidos funkcijos minimumo.

2.1.4.1 Optimizavimo algoritmai

Tyrimo metu, ieškant labiausiai tinkančio optimizavimo algoritmo, išbandomi 7 skirtingi algoritmai:

1. Adam (angl. *Adaptive Moment Estimation*) algoritmas. Stochastinis optimizavimo metodas, kuris naudoja tik pirmos eilės gradientų reikšmes, taip sunaudodamas mažiau atminties resursų. Šis algoritmas, pirmojo ir antrojo gradientų įverčių skirtingiems parametrams, taiko

skirtingus mokymosi greičius. Adam algoritmas, sujungia dviejų kitų optimizavimo algoritmų privalumus: AdaGrad algoritmo, kuris rodo gerus rezultatus retiems gradientams ir RMSProp, kuri rodo gerus rezultatus nestacionariems gradientams [10], [11]. Šis algoritmas, mokymosi greitį, kiekvienam parametru pritaiko atskirai, todėl ir yra priskiriamas prie adaptivių metodų. Adam išsaugo eksponentiškai mažėjančius praeities kvadratinus gradientų vidurkius, tačiau skirtingai nei Adadelta. Ir RMSprop, šis algoritmas, taip pat išsaugo ir eksponentiškai mažėjantį praeities gradientų vidurkį [12].

2. Stochastinis gradientinis nusileidimas. Iteracinis optimizavimo metodas, kuris atlieka parametru atnaujinimą kiekvienam mokymo poaibiui.
3. RMSProp (angl. *Root mean squared propagation*) algoritmas. Gradientinio nusileidimo algoritmas su impulsu. Šis algoritmas padalija mokymosi greitį iš eksponentiškai mažėjančio gradiento kvadratų vidurkio [12].
4. Adadelta (angl. *An Adaptive Learning Rate*) algoritmas. Stochastinis optimizavimo algoritmas, mažinantis monotoniškai mažėjantį mokymosi greitį. Adadelta algoritmas, užuot neefektyviai kaupęs ankstesnius kvadratinus gradientus, rekursiškai susumuoja ankstesnius mažėjančius kvadratinus gradientų vidurkius [12].
5. Adagrad (angl. *Adaptive Gradient Algorithm*) algoritmas. Gradientinio nusileidimo algoritmas, kuris taiko didesnius atnaujinimus retiems parametrams ir mažesnius dažniems parametrams, tokiu būdu, šis metodas skirtingiems parametrams pritaiko skirtingą mokymosi greitį. Šio metodo privalumas yra tai, kad jis rodo gerus rezultatus, retai išsidėsčiusiems duomenims [12].
6. AdaMax (angl. *Adaptive Moment Maximum*) algoritmas. Adam algoritmo plėtinys, kurio metu, metodas apibendrinamas iki begalinės normos (maksimumo) ir gali padėti efektyviau optimizuoti kai kurias problemas [12].
7. Nadam (angl. *The Nesterov-accelerated Adaptive Moment*). Adam algoritmo plėtinys, kurio metu yra pridodamas Nestrovo momentas. Pridėtas Nestrovo momentas, leidžia atlikti tikslesnį žingsnį, pasirenkant gradiento kryptį [12].

2.1.4.2 Bajeso optimizavimas

Tinkamas hiperparametrų įvedimas yra itin svarbus modelio tikslumui. Yra daugybė optimizavimo metodų hiperparametrų nustatymui, pradedant nuo atsitiktinės paieškos iki sudėtingesnių metodų. Bajeso optimizavimas, tai globalaus optimizavimo metodas galintis optimizuoti sudėtingas, triukšmingas juodųjų dėžių funkcijas. Literatūroje, šis metodas yra išskiriamas iš kitų paieškos metodų, dėl savo lankstumo ir gebėjimo tirti neapibrėžtas funkcijas [13]. Dažniausiai naudojamas Bajeso optimizavimo metodas paremtas Gauso procesu. Tarkime turime parametrų aibe x ir tikslo funkciją f kurios pavidalo nežinome. Akivaizdu, kad f yra sudėtinga įvertinti, nes nežinome šios funkcijos formos, struktūros ir ypatybių. Taigi galime teigti, kad f yra juodoji dėžė. Tuomet kaip pakaitinis modelis yra naudojamas Gauso procesas. Pakaitinis modelis optimizuojamas tol kol yra patenkinama sustojimo sąlyga arba pasiekiamas konvergavimas. Bajeso optimizavimas kaip ir kiti hiperparametrų paieškos metodai, daug kartų apskaičiuoja modelį su skirtingomis hiperparametrų reikšmėmis, tačiau šio metodo pranašumas yra tai, kad vykdoma paieška nėra atsitiktinė, o paremta ankstesnės informacijos vertinimu, o tai leidžia atrasti modelį su didžiausiu tikslumu, sunaudojant mažiau resursų.

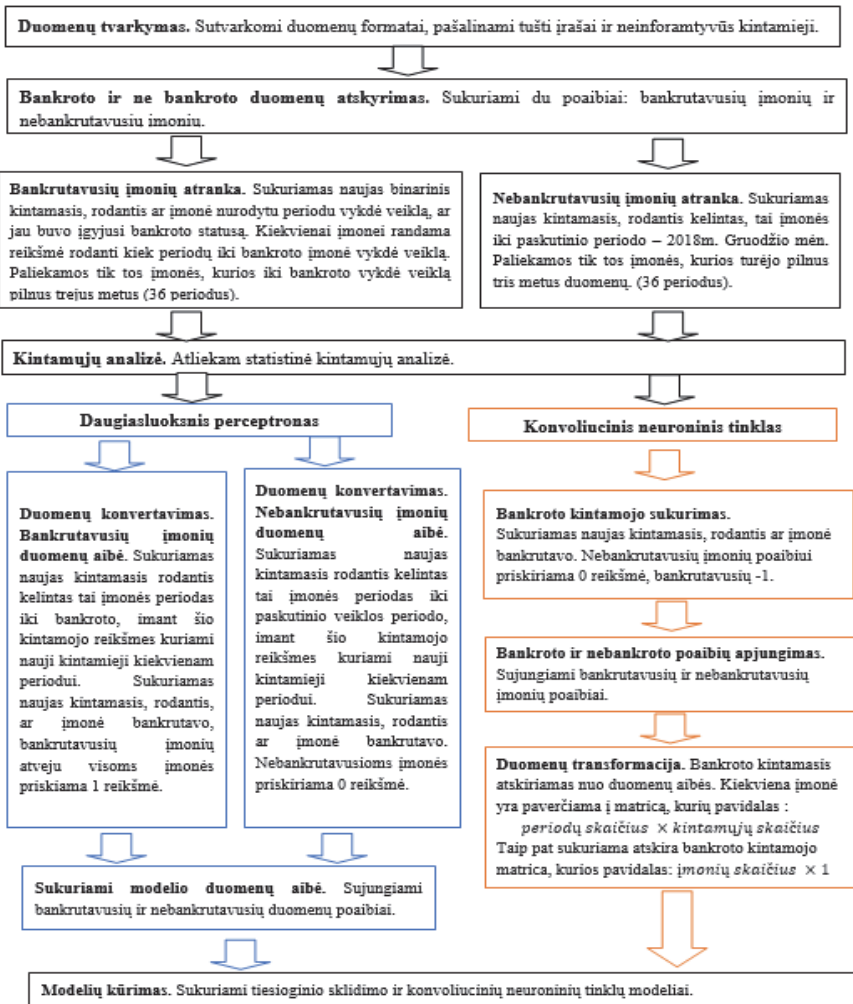
Tyrimas

2.2 Tyrimo planas

Tyrimą sudarė keli etapai, 4 pav. vaizduojamas tyrimo planas.

2.3 Duomenys

Tyrimo naudojami duomenys yra gauti iš valstybinės mokesčių inspekcijos ir Sodros. Duomenų rinkinį sudarė visų Lietuvos įmonių duomenys nuo 2013-01-01 iki 2018-12-01 (imtinai), surinkti iš PVM deklaracijų, pelno – nuostolių ataskaitų. Įmonių informacija nebūtinai buvo pateikiama visam nurodytam laikotarpiui, į duomenų rinkinį buvo įtrauktos įmonės ir turėjusios vos keletą mėnesių duomenų. Bendras visų duomenų dydis 8.17 GB, visų duomenų rinkinį sudarė 65 kintamieji. Pirmiausiai buvo atlikta duomenų kokybės analizė ir buvo atsisakyta tokių kintamųjų, kurie nesuteikia reikšmingos informacijos (tušti įrašai, kintamieji įgiję vieną reikšmę visu laikotarpiu). Taip pat buvo pašalintos įmonės, kuriose nagrinėjamu laikotarpiu buvo vykdoma reorganizacija. Atlikus literatūros analizę, buvo pastebėta, kad įmonės bankroto ženklus pradeda rodyti jau dveji metai iki bankroto [15].



4 pav. Tyrimo planas

Tam, kad modeliai galėtų tinkamai atskirti įmones, modelių sudarymui, buvo reikalingos tokios įmonės, kurios vykdė veiklą iki bankroto tokį pat laikotarpį kaip ir nebankrutavusios įmonės. Kadangi pateikti duomenys buvo penkerių metų laikotarpiui, buvo nuspręsta nagrinėti tokias įmones, kurios iki bankroto vykdė veiklą mažiausiai trejus metus. Tam, kad įsitikinti, kad trejų metų pakanka modelio sudarymui, buvo atlikta statistinė kintamųjų analizė, atskyrus bankrutavusias ir nebankrutavusias įmones. Statistinė analizė, buvo atliekama trejiems, dvejiems ir vieneriems metams iki bankroto. Buvo analizuojamos kintamųjų minimumų, maksimumų, standartinių nuokrypių, bei vidurkių reikšmės. Atlikus statistinę analizę, buvo pastebėta, kad bankrutavusių įmonių rodikliai nebuvo stabilūs jau dveji metai iki bankroto, o likus vieniems metams iki bankroto, kai kurie rodikliai rodė ir didelius skirtumus, lyginant su nebankrutavusių įmonių rodikliais.

Atlikus duomenų tvarkymo žingsnius, modelių sudarymui buvo palikti 45 kintamieji. Toliau yra pateikiami metiniai ir mėnesiniai kintamieji ir pagrindinės kintamųjų grupės.

- Bendra įmonės informacija:
 - įmonės amžius – 1 kintamasis;
 - ekonominė veiklos rūšis – 3 kintamieji;
 - geografinė įmonės informacija – 1 kintamasis;
 - teisinis statusas ir kodas – 1 kintamasis;
 - mokesčių mokėtojo informacija.
- Kintamieji iš metinių deklaracijų:
 - įmonių veiklos rodikliai iš pvm deklaracijų – 6 kintamieji;
 - informacija apie darbuotojus – 8 kintamieji;
 - darbo užmokesčio rodikliai – 2 kintamieji;
 - informacija apie įmonių nepriemokas – 2 kintamieji.

2.3.1 *Klasių disbalansas.*

Pastebėsime, kad nebankrutavusių ir bankrutavusių įmonių klasės buvo išbalansuotos. Bankrutavusių įmonių nagrinėjamoje imtyje buvo 1%, nebankrutavusių 99%.

Kuriant modelį, kuris turės veikti išbalansuotoje duomenų aibėje, svarbu duomenis subalansuoti taip, kad modelis pakankamai apsimokytų atpažinti mažumos klasę. Norint ištestuoti modelio gebėjimą klasifikuoti nesubalan-

suotoje duomenų aibėje, testavimo poaibyje yra tikslinga išlaikyti tokį klasių disbalansą, koks yra pradinėje duomenų imtyje. Todėl prieš duomenų subalansavimą, iš įmonių duomenų rinkinio, atsitiktinai atrenkama 15% įmonių ir sudaromas testavimo poaibis, kurį sudaro 12009 įmonių. Testavimo poaibyje išlaikomas klasių disbalansas – tik 1,3% įmonių turi bankroto statusą. Likę duomenys, naudojant dominuojančios klasės išmetimo metodą (angl. *Undersampling*), subalansuojami, taip, kad duomenų rinkinį sudarytų 15% bankrutavusių ir 85% nebankrutavusių įmonių. Subalansuotas duomenų rinkinys padalijamas į apmokymo ir validavimo poaibius, apmokymo poaibiui priskiriant 85% duomenų, validavimo 15% duomenų. Apmokymo poaibį sudarė 4799 įmonės, validavimo 854.

2.3.2 Duomenų konvertavimas

Tam, kad galėtume duomenis naudoti neuroninių tinklų modeliams, duomenis reikėjo konvertuoti. Daugiasluoksniu perceptrono atveju, buvo kuriami nauji kintamieji, periodams atgal. Atlikus duomenų konvertavimą, buvo gauti apmokymo, validavimo ir testavimo duomenų poaibiai, kuriuos sudarė 960 kintamųjų, šiuo atveju, vienas duomenų rinkinio įrašas atitiko vienos įmonės duomenis. Konvoliucinio neuroninio tinklo atveju, apmokymo, validavimo ir testavimo poaibiai buvo transformuojami į trimatę erdvę, kiekvieną įmonę paverčiant į matricą, kurios pavidalas – $36 \times 45 \times 1$. Po duomenų transformacijos, apmokymo, validavimo ir testavimo duomenų poaibiai turėjo tokį pavidalą: *įmonių skaičius* $\times 36 \times 45 \times 1$.

2.3.3 Modelių sudarymas

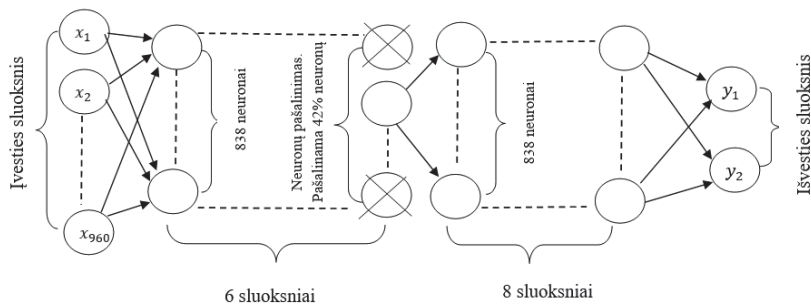
Konkrečiam uždaviniui spręsti tam, kad rasti geriausiai tinkantį neuroninio tinklo modelį, reikia rasti labiausiai tinkančią tinklo architektūrą ir tinkamiausius modelio hiperparametrus. Sudarius apmokymo, testavimo ir validavimo duomenų poaibius, kuriami neuroninių tinklų modeliai. Modelių apmokymui naudojami apmokymo, hiperparametrų validavimui – validavimo, modelių testavimui – testavimo duomenys. Geriausios neuroninio tinklo architektūros paieškai yra sukuriama funkcija, kuri sudaro neuroninio tinklo modelį. Ieškant geriausios tinklo architektūros, sukurta funkcija yra perduodama Bajeso optimizavimo metodui. Bajeso optimizavimo metu yra ieškoma maksimalaus modelio tikslumo – validavimo metu. Daugiasluoksniui perceptronui tiriami šie hiperparametrai: sluoksnių skaičius, neuronų

skaičius, aktyvacijos funkcija, optimizavimo algoritmas, mokymosi greitis, neuronų išmetimo sluoksnis ir išmetamų neuronų dydis, epochų skaičius, sluoksnių normalizavimas. Konvoliuciniam neuroniniam tinklui tiriami šie hiperparametrai: konvoliucinių sluoksnių skaičius, filtrų dydis, filtrų skaičius, aktyvacijos funkcija, neuronų išmetimo sluoksnis ir išmetimo dydis, neuronų skaičius, optimizavimo algoritmas. Daugiasluoksniui perceptronui išvesties sluoksnyje naudojama sigmoidinė aktyvacijos funkcija, konvoliucinio neuroninio tinklo modeliui – softmax funkcija.

3 Rezultatai

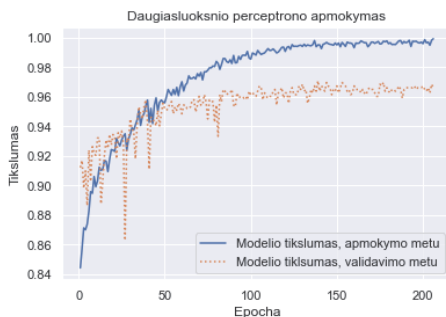
3.1 Daugiasluoksnis perceptronas

Atlikus hiperparametrų paiešką daugiasluoksniui perceptronui, buvo gautas neuroninis tinklas sudarytas iš 14 paslėptųjų sluoksnių. Paslėptuosius sluoksnius sudarė 838 neuronai, šeštame sluoksnyje pašalinama 42% neuronų. 5 pav. pateikiama daugiasluoksnių perceptrono modelio schema.

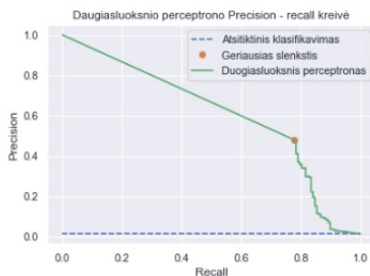
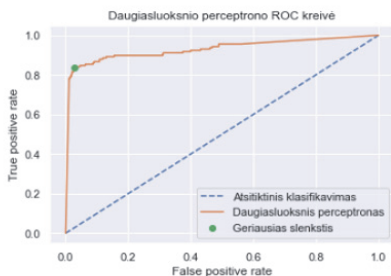


5 pav. Daugiasluoksnių perceptrono modelis

Gauto daugiasluoksnių perceptrono modelio 5 pav., paslėptuosiuose sluoksniuose buvo taikoma selu aktyvacijos funkcija. Modelio apmokymo metu taikomas Adadelta optimizavimo algoritmas. Apmokymo metu duomenys į priekį ir atgal buvo perduodami 206 kartus (epochų skaičius). Apmokymo metu gautas 99.9% modelio tikslumas, validavimo metu – 96.6%, testavimo metu – 97%. Toliau yra pateikiamas modelio apmokymo grafikas ir ROC, PRC kreivės.



6 pav. Daugiasluksnio perceptrono apmokymas.

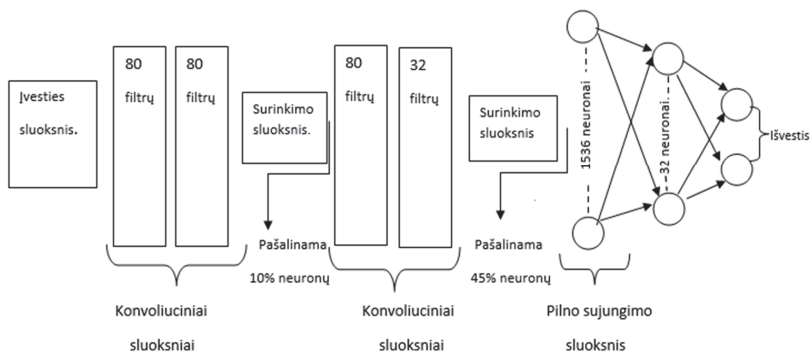


7 pav. Daugiasluksnio perceptrono ROC ir PRC kreivės.

Iš gautų ROC ir PRC kreivių 7pav., galime teigti, kad modelis geba atskirti klases. Pastebėsime, kad gautas pakankamai geras AUC įvertis, tačiau, kadangi modelis testuojamas nesubalansuotoje duomenų aibėje, aukštas AUC įvertis gali būti gaunamas, dėl gero daugumos klasės klasifikavimo, neatsižvelgiant į prastą mažumos klasės klasifikavimą, tačiau iš gauto F1 įverčio, kuris nėra labai prastas ir PRC kreivės, galime teigti, kad modelis geba prognozuoti bankroto atvejus.

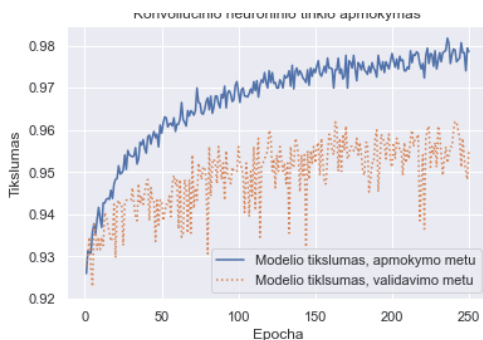
3.2 Konvoliucinis neuroninis tinklas

Atlikus hiperparametrų paiešką konvoliucinio neuroninio tinklo atveju, buvo gautas neuroninis tinklas sudarytas iš 8 paslėptųjų sluoksnių. Toliau pateiksim gautą konvoliucinio neuroninio tinklo schemą.



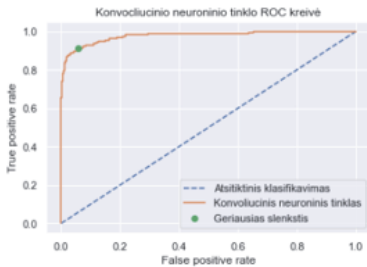
8 pav. Konvoliucinio neuroninio tinklo schema.

8 pav. pateikto modelio konvoliuciniuose sluoksniuose, buvo naudojama softsign aktyvacijos funkcija ir 3×3 filtrai. Surinkimo sluoksniuose, naudojama maksimumo surinkimo funkcija, o surinkimo žingsnis lygus 2. Pilno sujungimo sluoksnyje, naudojama elu aktyvacijos funkcija, o išvesties sluoksnyje softmax funkcija. Konvoliucinio tinklo apmokymui, naudojamas RMSprop optimizavimo algoritmas. Apmokymo metu duomenys buvo perduodi į priekį ir atgal 250 kartų. Apmokymo metu gautas 98.2% modelio tikslumas, validavimo metu – 96.3%, testavimo metu – 99%. Toliau pateikiama modelio apmokymo grafikas ir modelio ROC, PRC kreivės.



9 pav. Konvoliucinio neuroninio tinklo apmokymas.

Geriausias slenkstis=0.021930, AUC=0.977142



Geriausias slenkstis =0.999995, F1-įvertis=0.792



10 pav. Konvoliucinio neuroninio tinklo ROC ir PRC kreivės.

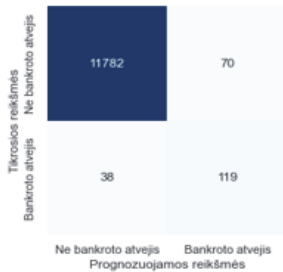
Iš gautų ROC ir PRC kreivių 10 pav., galime teigti, kad modelis gerai atskiria klases. Pastebėsime, kad gautas AUC įvertis yra labai arti idealaus klasifikavimo, taip pat pakankamai geras gaunamas ir F1 įvertis, vertinantis modelio gebėjimą klasifikuoti mažumos klasę.

3.3 Modelių palyginimas.

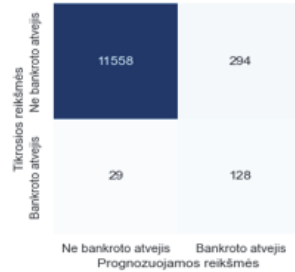
Norėdami palyginti modelių sumaišymo matricas ir klasifikavimo įverčius, klasifikavimui turime naudoti tą patį slenkstį. Iš gautų PRC ir ROC kreivių, pastebėjome, kad vertinant siūlomus slenksčius giliojo mokymosi modeliams, ROC kreivėse, gaunami slenksčiai yra pakankamai maži, tuo tarpu PRC kreivėse pakankamai dideli ir yra artimi maksimaliam galimam slenksčiui (vienetui). Didesnis slenkstis, mažintų prognozuojamų bankroto atvejų skaičių, o mūsų atveju, mums yra svarbiau identifikuoti bankroto atvejus, todėl yra geriau daugiau įmonių, kurioms negresia bankrotas, priskirti bankroto klasei, nei nepriskirti įmonių kurioms gresia bankrotas - bankroto klasei. Tačiau taip pat yra svarbu ir gerai atskirti nebankrutavusias įmones. Atsižvelgiant į tai, nusprendėme modelių klasifikavimo įverčių palyginimui, naudoti klasifikavimo slenkstį – 0.5. Vadinas, įmonės, kurioms buvo gauta bankroto tikimybė didesnė nei 0.5, buvo priskiriamos bankroto klasei.

Iš 11 pav. pateiktų sumaišymo matricų, pastebėsime, kad daugiasluoksnis perceptronas teisingai bankroto klasei priskyrė 9 įmonės daugiau, lyginant su konvoliuciniu neuroniniu tinklu. Tačiau, konvoliucinis neuroninis tinklas, klaidingai bankroto klasei priskyrė 4 kartus mažiau įmonių, lyginant su daugiasluoksnio perceptronu.

Konvoliucinis neuroninis tinklas



Daugiasluoksnis perceptronas



11 pav. Modelių sumaišymo matricos

1 lentelė. Modelių klasifikavimo įverčiai.

Modelis	Klasė	Precision	Recall	f1-score	Accuracy
Daugiasluoksnis perceptronas	Ne Bankroto	1	0.98	0.99	0.97
	Bankroto	0.3	0.82	0.44	
Konvoliucinis neuroninis tinklas	Ne Bankroto	1	0.99	1	0.99
	Bankroto	0.63	0.76	0.69	

Iš 1 lentelėje pateiktų precision įverčių, matome, kad modeliai puikiai prognozavo ne bankroto klasę. Vertinant, giliojo mokymosi metodais, prognozuojamus bankroto atvejus, galime teigti, kad 30% daugiasluoksnio perceptrono bankroto prognozių buvo teisingos, tuo tarpu konvoliucinio neuroninio tinklo atveju 63% prognozių buvo teisingos. Vertinant Recall įvertį, galime pastebėti, kad daugiasluoksnis perceptronas, ne bankroto klasei priskyrė 98% visų nebankroto atvejų, tuo tarpu konvoliucinis neuroninis tinklas – 99%. Vertinant kiek visų bankroto atvejų buvo priskirtų bankroto klasei, matome, kad daugiasluoksnis perceptronas teisingai priskyrė 82% įmonių, konvoliucinis neuroninis tinklas 76%. Vertinant f1 įvertį, kuris yra precision ir recall harmoninis vidurkis, galime teigti, kad konvoliucinis neuroninis tinklas klases atskyrė geriau. Vertinant bendrus modelių tikslumus, matome, kad konvoliucinio neuroninio tinklo bendras tikslumas buvo šiek tiek geresnis.

4 Išvados

Tyrimo metu sukurti giliųjų neuroninių tinklų modeliai, parodė gerus prognozavimo rezultatus. Gautų modelių bendri klasifikavimo tikslumai, tiek apmokymo, tiek testavimo metu buvo didesni nei 96%. Vertinant ROC ir PRC kreives, bei gautus klasifikavimo įverčius, galime teigti, kad bendrai įmonių bankrotą geriau prognozavo konvoliucinis neuroninis tinklas. Gautos sumaišymo matricos ir Recall įverčiai, rodo, tai, kad daugiasluoksnis perceptronas geriau atskyrė bankroto atvejus, bankroto klasei, tačiau daug daugiau klydo, atskirdamas bankroto atvejus, ne bankroto klasėje, lyginant su konvoliuciniu neuroniniu tinklu.

Bendrai paėmus, galime daryti išvadą, kad išbandyti giliojo mokymosi modeliai yra tinkami įmonių bankrotui prognozuoti ir geba atpažinti bankroto atvejus - duomenų aibėje su dideliu klasių disbalansu.

Literatūra

- [1] Altman E. (1968). Financial ratios discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance* 23(4) 589–609.”
- [2] Ohlson J. A. (1980). Financial ratios and the probabilistic prediction of bankruptcy. *Journal of Accounting Research*. (internetinė prieiga : Financial Ratios and the Probabilistic Prediction of Bankruptcy on JSTOR).
- [3] Shekar Shetty Mohamed Musa Xavier Brédart “Bankruptcy Prediction Using Machine Learning Techniques”.
- [4] Tadaaki Hosaka, Bankruptcy prediction using imaged financial ratios and convolutional neural networks, *Expert Systems with Applications*, Volume 117, 2019, Pages 287-299, ISSN 0957-4174, <https://doi.org/10.1016/j.eswa.2018.09.039>
- [5] Keiron O’Shea, Ryan Nash. 2015. An Introduction to Convolutional Neural Networks. <https://doi.org/10.48550/arXiv.1511.08458>
- [6] Azzouni, Abdelhadi & Boutaba, R. & Pujolle, Guy. (2017). NeuRoute: Predictive Dynamic Routing for Software-Defined Networks.
- [7] Clevert, Djork-Arné & Unterthiner, Thomas & Hochreiter, Sepp. (2015). Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). Under Review of ICLR2016 (1997).
- [8] Nguyen, A., Khoa Pham, Dat Thanh Ngo, Thanh Ngo and Lam Dang Pham. “An Analysis of State-of-the-art Activation Functions For Supervised Deep Neural Network.” 2021 International Conference on System Science and Engineering (ICSSE) (2021): 215-220.
- [9] Cilimkovic, Mirza. “Neural Networks And Back Propagation Algorithm.” (2010).
- [10] Kingma, Diederik P. and Ba, Jimmy. „Adam: A Method for Stochastic Optimization..“ *CoRR* abs/1412.6980 (2014).
- [11] Diederik P. Kingma, Jimmy Ba. 2017. ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION. <https://doi.org/10.48550/arXiv.1412.6980>
- [12] Ruder, Sebastian. (2016). An overview of gradient descent optimization algorithms.

- [13] Jasper Snoek Oren Rippel Kevin Swersky Ryan Kiros Nadathur Satish Narayanan Sundaram Md. Mostofa Ali Patwary Prabhat Ryan P. Adams (2015). Scalable Bayesian Optimization Using Deep Neural Networks. the Natural Sciences and Engineering Research Council of Canada. NSF IIS-1421780.
- [14] Flavio Barboza, Herbert Kimura, Edward Altman, Machine learning models and bankruptcy prediction, Expert Systems with Applications, Volume 83, 2017, Pages 405-417, ISSN 0957-4174.
- [15] Becerra-Vicario, Rafael & Alaminos, David & Llamas, Eva & Fernández-Gómez, Manuel. (2020). Deep Recurrent Convolutional Neural Network for Bankruptcy Prediction: A Case of the Restaurant Industry. Sustainability. 12. 5180. 10.3390/su12125180.

Pilvo aortos aneurizmos segmentavimas neuroniniais tinklais kompiuterinės tomografijos nuotraukose

Aušra Šubonienė

Vilniaus universitetas, Duomenų mokslo ir skaitmeninių technologijų institutas,
Akademijos g. 4, LT-08413, Vilnius
ausra.suboniene@gmail.com

Santrauka. Pilvo aortos aneurizmos diagnostikai ir stebėsenai dažnai naudojami kompiuterinės tomografijos vaizdai. Kontrastinėje tomografijos nuotraukoje kontrastas yra gerai matomas, tačiau automatinis trombo segmentavimas yra daug sudėtingesnė problema dėl aplink trombą esančių panašaus intensyvumo vaizdo taškų, vaizduojančių kitus vidaus organus. Šiame darbe atliekama pilvo aortos kontrasto ir trombo segmentacija naudojant tris neuroninius tinklus: ResNet-100, DeepLab V3 ir U-Net. Lyginami rezultatai, gauti naudojant skirtingas nuostolių funkcijas, taip pat taikomas programiškai išskaičiuotų trombo žymėjimų paruošimas naudojant morfologines operacijas. Taip pat pritaikomas atsitiktinis neuronų išmetimas ir klasifikavimas naudojant sumažinto lango kompiuterinės tomografijos vaizdus.

Raktiniai žodžiai: Pilvo aortos aneurizma, segmentacija, neuroniniai tinklai, kompiuterinė tomografija

1 Įvadas

Pilvo aortos aneurizma yra lėtinė liga, kuri negydoma sukelia pilvo aortos plyšimo komplikaciją [1]. Pilvo aortos aneurizma pasireiškia pilvo aortos išplatėjimu nusilpus jos sienelei, kurios diagnozei ir stebėsenai naudojama kompiuterinė tomografija. Susilpnėjusi aortos sienelė išsiplėčia, ji pasidengia kraujo komponentų sluoksniu, kuris vėliau susiformuoja į trombą. Aortos sienelės išsiplėtimas gali būti 1,5 ar daugiau kartų didesnis už normalų aortos skersmenį [2]. Išsiplėtusi pilvo aorta gali plyšti ir sukelti vidinį kraujavimą. Plyšus pilvo aortos aneurizmai mirtingumas siekia 85–90 % [2]. Plyšus pilvo aortos aneurizmai ir spėjus atlikti operaciją mirtingumas vis tiek išlieka aukštas, apie 33–36 % [1]. Tačiau jei plyšimo išvengiama atliekant operaciją, mirtingumas ženkliai sumažėja iki 1–6 % [3]. Todėl ankstyva ir greita pilvo aortos aneurizmos diagnostika yra ypač svarbi.

Tačiau vienas iš diagnostiką apsunkinančių veiksnių yra tai, kad šiuo metu diagnostikoje naudojami metodai ilgai užtrunka, ir trūksta automati- nių aneurizmos analizavimo ir stebėjimo įrankių [4]. Dažnai gydytojai, ana- lizuodami kompiuterinės tomografijos nuotraukas automatiniais ar pusiau automatiniais metodais, turi nurodyti pradinę aortos ir trombo vietą ar zoną, kurioje jie yra gerai matomi. Siekiant geresnės segmentavimo koky- bės šis procesas kartojamas beveik kiekvienai kompiuterinės tomografijos nuotraukai. Naudojant šiuos diagnostikos įrankius sukuriamas aortos kon- trasto trimatis modelis, tačiau į jį dažnai neįtraukiamas trombas. Turint tokį trimatį modelį sunku įvertinti aortos išsiplėtimą ir trombo dydį. Remiantis kraujagyslių chirurgijos draugijos praktikos gairėmis [4] geresnis šių proce- sų standartizavimas leistų geriau nustatyti ligos progresavimo laipsnį bei greitį, tačiau šiuo metu tam trukdo naudojamas rankinis aortos skersmens matavimas, kuomet aortos skersmuo matuojamas rankiniu būdu pagal tam tikrą vartotojo pasirinktą vaizdą.

Dažnai pilvo aortos segmentacija yra neautomatinė ar pusiau automa- tinė, kuomet vartotojas turi pažymėti pradinį kontrasto ar trombo tašką ar užimamą regioną kompiuterinės tomografijos nuotraukoje. Dažnai se- gmentacija atliekama naudojant regiono plėtimo metodus ar morfologi- nes operacijas, nors tiriami ir kiti būdai, tokie kaip matematinų modelių, tinklėlio algoritmų naudojimas ar aktyvusis mokymas [5], [6], [7]. Taip pat taikomi ir neuroniniai tinklai, tačiau dažniausiai atliekama visos aneurizmos segmentacija [8], nes trombo regionas dėl netaisyklingos formos ir neišsi- skiriančio vaizdo taškų intensyvumo yra sunkiai segmentuojamas. Šio dar- bo tikslas yra ištirti pilvo aortos aneurizmos (atskirai kontrasto ir trombo) segmentavimo metodus naudojant neuroninius tinklus.

2 Literatūros apžvalga

Literatūroje aprašyta pilvo aortos aneurizmos analizė atliekama naudojant įvairius duomenis. Dažnai naudojami kompiuterinės tomografijos angiogra- mos vaizdai (CTA) [9], [8]. Taip pat gali būti naudojami magnetinio rezonanso vaizdai (MRI) [10], ultragarso nuotraukos [7] ar net dirbtinai sukurti vaizdai [11]. Dažniausiai pilvo aortos aneurizmos segmentacijos metodai buvo ku- riami ir testuojami atrenkant tik priešoperacinius neplyšusios aneurizmos vaizdus, tačiau kartais tiriami ir pooperaciniai pilvo aortos aneurizmos vaiz- dai (pasiektas $0,82 \pm 0,07$ Dice panašumo koeficientas [12]). Pooperacinių vaizdų segmentavimas yra sudėtinga užduotis dėl netaisyklingos aortos for-

mos. Trombo segmentavimas (net ir priešoperacinis) taip pat turi daug iššūkių, nes vaizduose trombas labai panašus į šalia esančius organus. Todėl nors kiti aprašyti tyrimai gavo geresnius aortos aneurizmos segmentacijos rezultatus, [12] tyrime pasiekti rezultatai taip pat yra geri atsižvelgiant į tai, kad buvo vertinta vien tik trombo segmentacija.

Pilvo aortos aneurizmos vaizdų analizėje taikomi įvairūs metodai kompiuterinių tyrimų metu gautų vaizdų segmentacijai atlikti. Taškų intensyvumu paremti metodai yra lengvai implementuojami aortos ir trombo segmentacijos metodai. Tokių metodų pavyzdys yra Otsu metodas [13], naudotas aortos segmentacijai [14], kurio metu vaizduose esantys objektai atskiriami atliekant klasės skirtumų minimizaciją (ang. *minimisation of intra-class variances*). Rezultatai, gauti taikant šiuos metodus, yra nepastovūs ir gali suprastėti pasikeitus tyrimo aparatui, tyrimo parametrų ar aortos ar venos užsipildymu kontrastine medžiaga, todėl šie metodai dažnai jungiami su kitais metodais: naudojamos ir intensyvumo ir gradiento paieškos erdvės [15] ar taškų intensyvumo analizė derinama su grafo pjūviais paremtais metodais [16].

Regiono plėtimo metodas ir įvairūs jo variantai (dar vadinami aktyvaus kontūro metodais) yra dažnai naudojamas pilvo aortos aneurizmos segmentavimo metodas [17], [18]. Taikant šį metodą galima pritaikyti erdvinis apribojimus segmentuojamiems objektams, ko negalima padaryti naudojant pikselių intensyvumo histograma grįstus segmentavimo metodus, neapdorojančius erdvinės informacijos. Dažnai naudojant regiono plėtimo metodus reikalingas vartotojo įsikišimas segmentuojant nuotraukas. Gali būti prašoma nurodyti pirmą ir paskutinį sekos vaizdus, apribojančius aortos išplatėjimą. Taip pat prašoma vartotojo atlikti pradinį žymėjimą netoli aortos vidinių taškų [7]. Galimas regiono plėtimo metodo taikymas su papildomais kitų algoritmų išvestimi gautais apribojimais siekiant pagerinti tikslumą [19].

Pilvo aortos aneurizmos segmentavimą atliekant su neuroniniais tinklais dažniausiai naudojamas U-Net ar pilnai sujungtas konvoliucinis tinklas (ang. *deep convolutional neural network, DCNN*), nors galimos ir kitos tinklų struktūros [20], [21], [22]. Įprastai U-Net neuroninio tinklo mokymo metu kaip įvestis pateikiamas vienas vaizdas, tačiau galimas ir 3D U-Net variantas [23]. Be U-Net pilvo aortos aneurizmai segmentuoti taip pat yra naudojami ir giliai konvoliuciniai neuroniniai tinklai (DCNN), paremti pilniai sujungta konvoliucinių tinklų architektūra [12], [24]. DetecNet [12] aptikimo tinklas buvo pritaikytas dominančio regiono (ang. *region of interest*) aptikimui. Tuomet buvo panaudota nauja pilnai sujungtų konvoliucijų tinklų architektūra ir tinklas buvo testuotas naudojant 13 pooperacinių pacientų vaizdų sekų [12].

Autoriai pasiekė 0.82 Dice panašumo koeficientą pooperaciniams pacientų vaizdams. Kitame tyrime [24] buvo gauti panašūs rezultatai, kur Dice panašumo koeficientas pooperaciniams pacientų vaizdams buvo $0,837 \pm 0,062$, priešoperaciniams: $0,790 \pm 0,102$.

Nors naudojant giliojo mokymo metodus buvo pasiekti geri segmentavimo rezultatai, ypač pooperacinių pacientų vaizdams, šių metodų generalizacija yra sudėtingas uždavinys radiologinių tyrimų vaizdams [25]. Atsiradus neatitikimams dėl įrangos, kuria gaunami vaizdai, pakeitimo, pasikeitusių vaizdo gavimo algoritmų kompiuterinės tomografijos metu, ar tyrimus atliekant kitoms demografinėms pacientų grupėms tokio segmentavimo rezultatai blogėja. Toks generalizacijos trūkumas gali reikšmingai pakenkti modelių klinikiniam taikymui, nes modelio tikslumas gali sumažėti atliekant tyrimus su kita įranga, nei buvo naudojama mokymo duomenims gauti [23].

3 Aortos ir trombo aptikimo metodai

ResNet yra semantinės segmentacijos uždaviniams spręsti skirtas neuroninis tinklas, sukurtas siekiant sumažinti nykstančio gradiento problemą giliuose neuroniniuose tinkluose [26]. Iki ResNet įprastas būdas pagerinti klasifikavimo ar segmentacijos tikslumą naudojant konvoliucinius tinklus buvo didinti jų sudėtingumą pridendant papildomų sluoksnių. Tačiau tokios neuroninių tinklų architektūros dažnai susidurdavo su nykstančio gradiento problema. Tinklo mokymo metu naudojant klaidos skidimą atgal į pradinius sluoksnius, gradientas dažnai tapdavo nepastebimai mažas dėl didelio kiekio daugybės operacijų. Apmokant tokius gilius neuroninius tinklus dažnai galima pastebėti, kad gilėjant neuroniniam tinklui jo pasiektas tikslumas dažnai sustoja arba net pradeda blogėti. ResNet neuroninis tinklas siekia išvengti šios problemos naudodamas tapatumo jungtį (ang. *identity connection*), kuri apeina vieną ar daugiau sluoksnių, sujungdama sluoksnius į likučių bloką (ang. *residual block*). Tapatumo sąryšio įvedimas neprideda papildomų tinklo parametru, todėl skaičiavimo sudėtingumas yra beveik toks pat, kaip ir paprastų neuroninių tinklų. Likučių blokas dažnai sujungia kelias konvoliucines operacijas. Klaidos sklidimo atgal metu gradiento sklidimas vyksta naudojant naujai įvestą tapatumo ryšį, todėl apeinami likutinio bloko vidiniai svoriai ir sumažinama nykstančio gradiento problema. Naudojant ResNet tapatumo ryšius ir konvoliucinius sluoksnius, sujungtus į likučių blokus, buvo sėkmingai apmokyti 50 ir 100 sluoksnių gylio konvoliuciniai neuroniniai tinklai [26].

DeepLab V3 gilusis neuroninis tinklas yra neuroninis tinklas, pritaikytas segmentacijos uždaviniui spręsti, nes dėl skylėtų konvoliucijos operacijų požymių žemėlapis (ang. *feature map*) nesumažinamas tiek, kiek kituose pilnai konvoliuciniuose tinkluose, todėl prarandama mažiau informacijos apie vaizduose esančių objektų vietą [27]. DeepLab V3 neuroninis tinklas naudoja išplėstų konvoliucijų (ang. *dilated convolutions*) idėją iš DilatedNet konvoliucinio tinklo [28], bet DeepLab V3 atveju vienodai išplečiami visi filtro taškai. DeepLab V3 autoriai taip pat taiko skylėtos erdvinės piramidės grupavimą (ang. *trous spatial pyramid pooling*) konvoliucijos blokuose [27]. Tai yra skylėta SPP versija, naudojama SPPNet tinkle [29]. Keli skylėtų konvoliucijų filtrai su skirtingais skylėtumo koeficientais yra taikomi konvoliucijos sluoksnio išvesčiai, kurių rezultatai vėliau apjungiami. Šis metodas pagerina objektų, kurių skalės vaizduose gali daug skirtis, aptikimą.

U-Net neuroninio tinklo architektūra buvo sukurta remiantis pilno konvoliucinio tinklo architektūra [30] ir buvo skirta semantinei segmentacijai pagerinti medicininiuose vaizduose. Tinklo architektūra sudaryta iš dviejų kelių. Pirmasis kelias yra duomenų sutraukimo kelias, taip pat vadinamas enkoderiu (ang. *encoder*), kuris naudojamas vaizdo duomenų kontekstui rasti. Antrasis kelias yra išplėtimo kelias, kuris dar yra vadinamas dekoderiu (ang. *decoder*), ir yra simetriškas enkoderio tinklui. Dekoderis suteikia gali-mybę gauti tikslią objektų lokalizaciją pradiniam vaizde, ne tik apibendrin-tą informaciją, kuri yra gaunama kaip enkoderio išvestis. Dekoderis naudo-ja transponuotas konvoliucijos operacijas ir yra nuo pradžios iki galo pilnai konvoliucinis tinklas (ang. *fully convolutional network, FCN*). U-Net tinklą su-daro tik konvoliuciniai sluoksniai be pilnai sujungto sluoksnio (ang. *dense layer*). Todėl U-Net tinklas kaip įvestį gali priimti bet kokio dydžio vaizdą.

4 Segmentavimo vertinimo metodai

Segmentavimo rezultatams vertinti buvo pasirinkti du metodai: Dice koefi-cientas bei sankirtos ir sąjungos santykis (IoU) (ang. *intersection over union*). Dice panašumo koeficientas [31] naudojamas kiekybiškai įvertinti dviejų imčių panašumą. Vaizdų semantinės segmentacijos atveju Dice koeficien-tas interpretuojamas kaip panašumas tarp tikros objekto segmentacijos ir segmentacijos spėjimo, gauto naudojant segmentacijos modelį. Dice koefi-cientas apskaičiuojamas kaip dvigubo sankirtos dydžio ir abiejų aibių dydžių sumos santykis:

$$Dice = \frac{2|X \cap Y|}{|X| + |Y|}.$$

Sprendžiant vaizdų segmentacijos uždavinį dviejų klasių atveju $|X \cap Y|$ yra teisingai teigiamai klasifikuotų vaizdo taškų skaičius, $|X|$ ir $|Y|$ yra atitinkamai tikrų teigiamos klasės vaizdo taškų ir prognozuotų teigiamos klasės vaizdo taškų skaičius.

Jaccard panašumo koeficientas [32], dar žinomas kaip sankirta virš sąjungos, kuri dažnai trumpinama kaip IoU (ang. *intersection over union*), yra dar vienas dažnas vaizdų segmentacijoje naudojamas modelių rezultatų įvertinimo metodas. Naudojant Jaccard panašumo koeficientą segmentacija vertinama skaičiuojant santykį tarp bendrų dviejų klasių aibių taškų ir dviejų klasių sąjungos taškų. Jaccard panašumo koeficientas taikomas vaizdų segmentacijoje skaičiuojant santykį tarp tikros ir prognozuotos klasifikacijos aibių sankirtos ir sąjungos:

$$IoU = \frac{|X \cap Y|}{|X \cup Y|}.$$

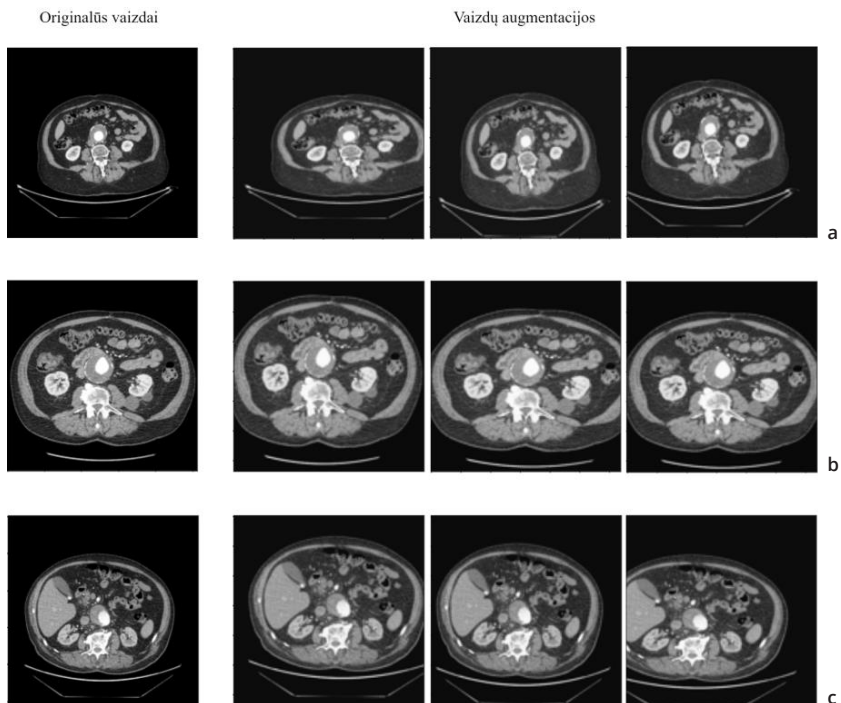
5 Duomenys

Šiame darbe buvo naudotos 17 pacientų kompiuterinės tomografijos vaizdų sekos, kurioms gauti buvo naudota kontrastinė medžiaga. Originaliuose duomenyse buvo gautos 30 pacientų vaizdų sekos [8], tačiau dėl trūkstamų sužymėjimų darbe buvo naudoti tik 17 pacientų duomenys. Pacientų kompiuterinės tomografijos vaizdai su diagnozuota pilvo aortos aneurizma buvo surinkti nuo 2015 m. sausio mėn. iki 2019 m. gruodžio mėn. Gdansko medicinos universiteto Radiologijos departamente. Vidutinis pacientų amžius yra 70,6 metai. Iš 30 pacientų atrinkti 23 vyrų ir 7 moterų vaizdai.

Kompiuterinės tomografijos vaizdus žymėjo trijų medikų komanda: kraujagyslių chirurgas bei du kraujagyslių chirurgijos rezidentai. Iš pradžių buvo atliekama automatinė kontrasto segmentacija naudojant aktyvaus kontūro metodą nuo distalinės krūtinės aortos lygio iki bendrųjų šlaunikaulio arterijų. Taip pat buvo atliktas automatinis pagrindinių pilvo aortos išsišakojimų (celiakijos kamieno, viršutinės mezenterinės ir inkstų arterijų) pradžių sužymėjimas naudojant aktyvaus kontūro metodą. Antrame žymėjimo žingsnyje buvo atliekamas rankinis pilvo aortos aneurizmos žymėjimas rankiniu būdu pažymint išorinę aortos sieną kiekviename vaizde. Vėliau žymėjimai buvo programiškai sujungti pirmenybę teikiant rankinio žymėjimo rezultatams. Pacientų vaizdų sekas sudarė 450–1300 vaizdų, iš kurių 40–60

% reikėjo rankinio sužymėjimo. Sužymėjimo pabaigoje vaizdai buvo anonimizuoti [8].

Atlikus pradinį vaizdų apdorojimą ir suvienodinus kontrasto ir trombo žymėjimus, vaizdai iš DICOM formato buvo konvertuoti į PNG atliekant HU transformaciją (lango plotis 400, lango lygis 35). Tuomet vaizdai buvo normalizuoti į [0, 1] intervalą taikant min-max normalizaciją. Atlikus duomenų paruošimą buvo taikoma duomenų augmentacija, siekiant padidinti turimų vaizdų skaičių. Buvo taikomas posūkis ir stačiakampių ar kvadratų iškirpimas vaizde, kurių ilgio ir pločio santykis yra nuo 3/4 iki 4/3. Tuomet iškirpti stačiakampiai buvo transformuojami į pradinio vaizdo dydį. Gautas vaizdas galėjo apimti nuo 75 % iki 100 % pradinio vaizdo ploto. Šios duomenų augmentacijos buvo taikytos atsitiktiniu būdu kiekvienam vaizdai ir jo žymėjimui kiekvienos epochos pradžioje. Duomenų augmentacijos pavyzdžiai pavaizduoti 1 pav.



1 pav. Trijų pacientų duomenų augmentacijų pavyzdžiai.

6 Rezultatai

Aortos ir trombo segmentacijai buvo taikyti trys neuroninių tinklų modeliai: ResNet-101, DeepLab V3 ir U-Net. Dėl ribotų skaičiavimo resursų vaizdai buvo sumažinti iki 128x128 vaizdo taškų. 1 lentelėje pavaizduoti neuroninių tinklų testavimo rezultatai aortos kontrasto segmentacijai. Geriausi rezultatai po fiksuoto epochų skaičiaus pasiekti naudojant U-Net neuroninį tinklą su mišria nuostolių funkcija, apjungiančia kryžminę entropiją ir Dice panašumo koeficientą.

Trombo segmentaciją atlikti sudėtinga, nes aplink trombą yra panašios geometrijos struktūros, žymimos panašaus intensyvumo vaizdo taškais. Siekiant palengvinti trombo segmentaciją pradinis vaizdas buvo apkarpomomas iki 312x312 pašalinant kompiuterinės tomografijos vaizdo taškus. Taip pat anksčiau gautus rezultatus bandyta pagerinti taikant morfologines operacijas trombo žymėjimams. Dėl programiškai gautų trombo žymėjimų kartais gauti žymėjimai turi pavienių vaizdo taškų ar nesujungtų regionų, priskiriamų trombui, kurie galimai atsiranda dėl nevienodo aortos sienos identifikavimo kompiuterinės tomografijos vaizduose. Problemą bandyta spręsti taikant išplėtimo ir erozijos operacijas, siekiant sujungti atskirus trombą žyminčius regionus su smulkiais tarpais.

1 lentelė. Aortos segmentavimo rezultatai naudojant skirtingus neuroninius tinklus ir nuostolių funkcijas.

Nuostolių funkcija	Kryžminė entropija		Dice koeficientas		Kryžminė entropija ir Dice koeficientas	
	Dice	IoU	Dice	IoU	Dice	IoU
Segmentavimo tikslumo vertinimas						
ResNet-101	0,26774	0,17900	0,80633	0,68460	0,63592	0,50207
DeepLab V3	0,29460	0,21513	0,80479	0,68563	0,73527	0,60535
U-Net	0,81529	0,71421	0,80576	0,70532	0,87554	0,79734

Analizuojant po vaizdo apkarpyto gautus grafinius rezultatus buvo pastebėta, kad tinkle atsiranda permokymas (ang. *overtraining*). Siekiant to išvengti, buvo pritaikytas atsitiktinis svorių išmetimas (ang. *dropout*) tarp trijų U-Net tinklo dekoderio sluoksnių. Nors dėl atsitiktinio svorių išmetimo šiek tiek pablogėjo aortos kontrasto segmentavimas, tačiau trombo segmentacija pagerėjo maždaug 1,6 karto. Segmentacijos rezultatai atlikus pradinių vaizdų apkarpyimą ir atsitiktinį svorių pašalinimą pateikiami 2 lentelėje.

2 lentelė. Segmentavimo rezultatai taikant vaizdų apkarpyimą ir atsitiktinį svorių pašalinimą

Segmentuotas regionas	Tik su vaizdo apkarpymu		Su vaizdo apkarpymu ir atsitiktiniu svorių pašalinimu	
	Dice	IoU	Dice	IoU
Kontrastas	0,91799	0,86600	0,86610	0,79662
Trombas	0,22655	0,15133	0,38257	0,29939

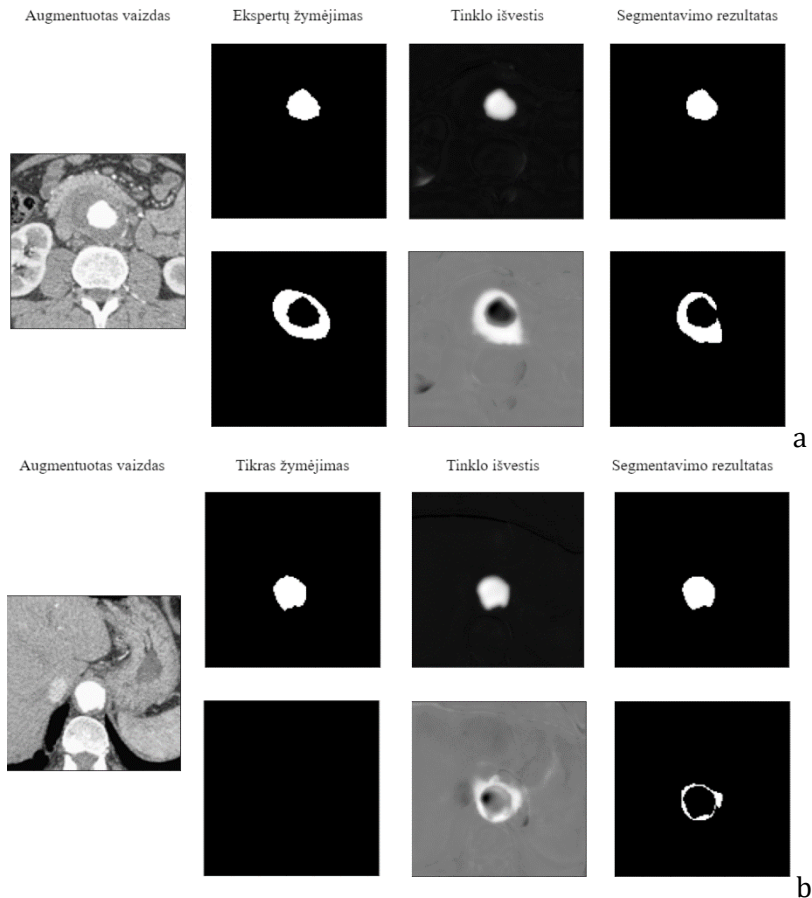
Apmokius U-Net neuroninį tinklą siekta pagerinti rezultatus apjungiant du skirtingai apmokytus neuroninius tinklus. Duomenyse yra trombo žymėjimo problemų, taip pat trombo regionas yra sudėtingai identifikuojamas dėl panašaus intensyvumo vaizdo taškų, esančių aplink trombą bei netaisyklingos jo formos. Todėl buvo siekta palengvinti segmentavimo užduotį apmokant du skirtingus neuroninius tinklus. Pirmasis tinklas buvo apmokytas atlikti tik kontrasto segmentaciją, kitas tinklas segmentuoja tik bendrą kontrasto ir trombo regioną. Trombo regionas gautas atėmus segmentuotą kontrasto regioną iš bendro regiono. Mokymas ir testavimas atliktas keliais būdais: naudojant visus turimus pacientų vaizdus ir naudojant tik tuos vaizdus, kuriuose trombas yra matomas.

3 lentelė. Rezultatai apmokius du neuroninius tinklus skirtingiems regionams segmentuoti

Regionas	Du segmentavimo tinklai, testuoti su visais vaizdais		Du segmentavimo tinklai, testuoti tik su matomo trombo vaizdais		Du segmentavimo tinklai, apmokyti tik matomo trombo vaizdais, testuoti su visais		Du segmentavimo tinklai, apmokyti ir testuoti tik matomo trombo vaizdais	
	Dice	IoU	Dice	IoU	Dice	IoU	Dice	IoU
Kontrastas	0,9058	0,8582	0,9066	0,8662	0,9399	0,8941	0,9458	0,9054
Trombas	0,4123	0,3325	0,5820	0,4694	0,3906	0,3202	0,5514	0,4521
Bendras	0,8444	0,7612	0,8296	0,7453	0,7931	0,7132	0,7725	0,7017

Testavimo rezultatai pavaizduoti 3 lentelėje. Kontrasto segmentavimo rezultatai yra geriausi tuomet, kai du tinklai buvo apmokyti ir testuoti tik su matomo trombo vaizdais, nors skirtumas tarp testavimo vaizdų variantų nėra toks didelis kaip tarp mokymo vaizdų variantų. Trombo segmentavimo testavimo rezultatai yra daug geresni tais atvejais, kai buvo testuojama naudojant tik trombą vaizduojančius vaizdus. Tai galima paaiškinti analizuojant

atliktų segmentacijų pavyzdžius. Pav. 2b matoma, kad testavimui naudojant ir tuos vaizdus, kuriuose nėra trombo, trombas yra vis tiek surandamas. 2b vaizde matoma, kad kartais aplink kontrastą esančios kitos anatomicinės struktūros, tokios kaip raumenys, yra identifikuojamos kaip trombas. Tokių klaidų išvengiama, kai testavimas atliekamas tik su vaizdais, kuriuose matomas trombas. Pav. 2a pavaizduotas trombo segmentavimo pavyzdys, kai rastos trys iš keturių trombo pusių.



2 pav. Kontrasto ir trombo segmentacijų pavyzdžiai, testuojant ir apmokant su visais duomenimis.

7 Išvados

Apmokius tris neuroninius tinklus su skirtingomis nuostolių funkcijomis pasiekti geri kontrasto segmentavimo rezultatai, tačiau trombo segmentavimo rezultatai nėra tokie geri. Tai gali lemti per mažas tinklo mokymui turimas duomenų kiekis, žymėjimų netikslumai vaizduose ir trombo erdvinės savybės. Turint mažai pacientų įvyksta tinklo persimokymas, kurio iš dalies galima išvengti taikant atsitiktinių svorių pašalinimą.

Literatūroje aprašyto segmentacijų tikslumo, gauto su turimais duomenimis, gauti ar pagerinti nepavyko, tačiau pavyko gauti panašius kontrasto segmentavimo rezultatus: [8] pasiekė 0,94 Dice panašumo koeficientą aneurizmos segmentacijai, kur kontrasto ir trombo regionas naudojamas kaip viena klasė ir skaičiuojamas trijų neuroninių tinklų rezultatų svorinis vidurkis segmentacijoms rasti. Šiame darbe sprendžiama problema yra sudėtingesnė, nes siekiama atskirai segmentuoti trombo regioną. Mažesnis segmentacijos tikslumas gaunamas ir dėl mažesnio duomenų kiekio, kuris naudojamas apmokyti. Šiame darbe buvo naudojami 17 iš 30 pacientų vaizdai. Siekiant atskirti skirtingų pacientų vaizdus mokymo, testavimo ir validavimo duomenyse, naudojamas daug mažesnis pacientų skaičius tinklo mokymui ir validavimui, kas lemia mažesnę vaizdų įvairovę.

Gautus pradinis rezultatus pavyko pagerinti naudojant du atskirus neuroninius tinklus, kurie buvo apmokyti klasifikuoti tik vieną regioną (kontrastą arba trombą). Taip supaprastinus mokymo užduotį galima pasiekti gerų rezultatų net ir naudojant beveik du kartus mažesnę duomenų kiekį nei [8] darbe.

Naudojant 2D neuroninius tinklus prarandama erdvinė informacija, todėl tampa sunkiau identifikuoti anatomines struktūras, turinčias mažą skerspūvio plotą. Objekto segmentacijos sekimas 3D duomenyse galėtų palengvinti ilgų anatominių struktūrų, turinčių mažą skerspūvio plotą, segmentavimą, todėl maži pavieniai trombo regionai galėtų būti geriau atskiriami. Dėl nepakankamo pacientų kompiuterinės tomografijos vaizdo sekų kiekio pilnas 3D U-Net tinklo mokymas yra negalimas, tačiau galima tinklą mokyti 3D objekto segmentais kaip tinklo įvestį naudojant tris ar penkis iš eilės einančius kompiuterinės tomografijos sekos vaizdus.

Literatūra

- [1] Ramkumar, N., Suckow, B. D., Arya, S., Sedrakyan, A., Mackenzie, T. A., Goodney, P. P., & Brown, J. R. (2020). Association of Sex With Repair Type and Long-term Mortality in Adults With Abdominal Aortic Aneurysm. *JAMA Network Open*, 3(2), e1921240–e1921240.
- [2] Kent, K. C. (2014). Abdominal aortic aneurysms. *New England Journal of Medicine*, 371(22), 2101–2108.
- [3] Greenhalgh, R. M., Brown, L. C., Epstein, D., Kwong, G. P. S., Powell, J. T., & Sculpher, M. J. (2005). Endovascular aneurysm repair and outcome in patients unfit for open repair of abdominal aortic aneurysm (EVAR trial 2): randomised controlled trial. *Lancet*, 365(9478), 2187–2192.
- [4] Chaikof, E. L., Dalman, R. L., Eskandari, M. K., Jackson, B. M., Lee, W. A., Mansour, M. A., Mastracci, T.M., Mell, M., Murad, M.H., Nguyen, L.L. and Oderich, G.S. & Starnes, B. W. (2018). The Society for Vascular Surgery practice guidelines on the care of patients with an abdominal aortic aneurysm. *Journal of vascular surgery*, 67(1), 2–77.
- [5] Maiora, J., Ayerdi, B., & Graña, M. (2014). Random forest active learning for AAA thrombus segmentation in computed tomography angiography images. *Neurocomputing*, 126, 71–77.
- [6] Dolgov, V. Y., Klyshnikov, K. Y., Ovcharenko, E. A., Glushkova, T. V., Batranin, A. V., Agienko, A. S., ... & Kutikhin, A. G. (2019). Finite element analysis-based approach for prediction of aneurysm-prone arterial segments. *Journal of Medical and Biological Engineering*, 39(1), 102–108.
- [7] Cercenelli, L., Lodi, S., Bortolani, B., Mascoli, C., Gargiulo, M., & Marcelli, E. (2020). EVAR-eaSE: An Easy-to-Use Software for Planning Sac Embolization in EndoVascular Aneurysm Repair Procedure. *Applied Sciences*, 10(18), 6252.
- [8] Dziubich, T., Białas, P., Znaniecki, Ł., Halman, J., & Brzeziński, J. (2020, August). Abdominal aortic aneurysm segmentation from contrast-enhanced computed tomography angiography using deep convolutional networks. In *ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium* (pp. 158–168). Springer, Cham.
- [9] Rengarajan, B., Wu, W., Wiedner, C., Ko, D., Muluk, S. C., Eskandari, M. K., ... & Finol, E. A. (2020). A comparative classification analysis of abdominal aortic aneurysms by machine learning algorithms. *Annals of biomedical engineering*, 48(4), 1419–1429.
- [10] Wang, Y., Tian, B., Xiong, F., Kao, E., Zhang, Y., Liu, X., ... & Saloner, D. (2021). Computer-aided quantification of non-contrast 3D black blood MRI as an efficient alternative to reference standard manual CT angiography measurements of abdominal aortic aneurysms. *European Journal of Radiology*, 134, 109396.
- [11] Jalahmadi, G., Linte, C., & Helguera, M. (2017, March). A numerical framework for studying the biomechanical behavior of abdominal aortic aneurysm. In *Medical Imaging 2017: Biomedical Applications in Molecular, Structural, and Functional Imaging* (Vol. 10137, p. 101372A). International Society for Optics and Photonics.
- [12] López-Linares, K., Aranjuelo, N., Kabongo, L., Maclair, G., Lete, N., Ceresa, M., ... & Ballester, M. A. G. (2018). Fully automatic detection and segmentation of abdominal aortic thrombus in post-operative CTA images using deep convolutional neural networks. *Medical image analysis*, 46, 202–214.
- [13] Otsu, N. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1), 62–66, 1979.

- [14] Lessard, S., Kauffmann, C., Pfister, M., Cloutier, G., Thérasse, É., de Guise, J. A., & Soulez, G. (2015). Automatic detection of selective arterial devices for advanced visualization during abdominal aortic aneurysm endovascular repair. *Medical engineering & physics*, 37(10), 979–986.
- [15] Siriapisith, T., Kusakunniran, W., & Haddawy, P. (2019). 3D segmentation of exterior wall surface of abdominal aortic aneurysm from CT images using variable neighborhood search. *Computers in biology and medicine*, 107, 73–85.
- [16] Freiman, M., Esses, S. J., Joscowicz, L., & Sosna, J. (2010, April). An iterative model-constrained graph-cut algorithm for abdominal aortic aneurysm thrombus segmentation. In *2010 IEEE international symposium on biomedical imaging: from nano to macro* (pp. 672–675). IEEE.
- [17] Kim, H. C., Park, S. W., Nam, K. W., Choi, H., Choi, E. J., Jin, S., ... & Sun, K. (2010). Determination of accurate stent graft configuration in abdominal aortic aneurysm using computed tomography: a preliminary study. *Clinical imaging*, 34(4), 255–262.
- [18] van Disseldorp, E. M. J., van Dronkelaar, J. J., Pluim, J. P. W., van de Vosse, F., van Sambeek, M. R. H. M., & Lopata, R. G. P. (2016, September). Automatic segmentation and registration of abdominal aortic aneurysms using 3D ultrasound. In *2016 IEEE International Ultrasonics Symposium (IUS)* (pp. 1–4). IEEE.
- [19] Macía, I., de Blas, M., Legarreta, J. H., Kabongo, L., Hernández, Ó., Egaña, J. M., ... & Graña, M. (2016). Standard and fenestrated endograft sizing in EVAR planning: description and validation of a semi-automated 3D software. *Computerized Medical Imaging and Graphics*, 50, 9–23.
- [20] Maiora, J., & Graña, M. (2012, March). A hybrid segmentation of abdominal CT images. In *International Conference on Hybrid Artificial Intelligence Systems* (pp. 416–423). Springer, Berlin, Heidelberg.
- [21] Maiora, J., & Graña, M. (2012, June). Abdominal CTA image analysis through active learning and decision random forests: Application to AAA segmentation. In *The 2012 international joint conference on neural networks (IJCNN)* (pp. 1–7). IEEE.
- [22] Shum, J., DiMartino, E. S., Goldhammer, A., Goldman, D. H., Acker, L. C., Patel, G., ... & Finol, E. A. (2010). Semiautomatic vessel wall detection and quantification of wall thickness in computed tomography images of human abdominal aortic aneurysms. *Medical physics*, 37(2), 638–648.
- [23] Lu, J. T., Brooks, R., Hahn, S., Chen, J., Buch, V., Kotecha, G., ... & Tenenholz, N. A. (2019, October). DeepAAA: clinically applicable and generalizable detection of abdominal aortic aneurysm using deep learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 723–731). Springer, Cham.
- [24] López-Linares, K., Kabongo, L., Lete, N., Maclair, G., Ceresa, M., García-Familiar, A., ... & Ballester, M. A. G. (2017). Dcnv-based automatic segmentation and quantification of aortic thrombus volume: Influence of the training approach. In *Intravascular Imaging and Computer Assisted Stenting, and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis* (pp. 29–38). Springer, Cham.
- [25] Zech, J. R., Badgeley, M. A., Liu, M., Costa, A. B., Titano, J. J., & Oermann, E. K. (2018). Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS medicine*, 15(11), e1002683.
- [26] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

- [27] Chen, L. C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587.
- [28] Yu, F., & Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122.
- [29] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9), 1904–1916.
- [30] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234–241). Springer, Cham.
- [31] Dice, Lee R. (1945). „Measures of the Amount of Ecologic Association Between Species“. *Ecology*. 26 (3): 297–302. doi:10.2307/1932409. JSTOR 1932409.
- [32] Jaccard, Paul (1912). „The Distribution of the Flora in the Alpine Zone.1“. *New Phytologist*. 11 (2): 37–50. doi:10.1111/j.1469-8137.1912.tb05611.x. ISSN 0028-646X.

Apgaulingi šablonai įtaigoje kompiuterijoje

Artūras Tarasenka

Vilniaus universitetas, Matematikos ir informatikos fakultetas
Didlaukio g. 47, LT-08303 Vilnius
arturas.tarasenka@mif.stud.vu.lt

Santrauka. Apgaulingi šablonai (angl. *dark patterns*) – tai vartotojo sąsajos šablonai, kuriuose išnaudojant psichologinius dėmesio pritraukimo ir išlaikymo ypatumus vartotojai verčiami priimti paslaugų tiekėjams naudingus sprendimus, užtikrinančius verslo pelningumą. Šie šablonai yra vadinami apgaulingais, kadangi naudotojai dažnai nepastebi jų lūkesčius neatitinkančių manipuliacijų. Šiame tyrime identifikuojami apgaulingų šablonų atributai ir jų pagrindu sudaroma vartotojo sąsajos inspektavimo metodika, skirta įvertinti vartotojo sąsajos atitikimą etinio projektavimo gairėms ir identifikuoti manipuliatyvius elementus. Metodikoje siekiama nustatyti teisės į informaciją, panaudojamumo, saugumo, lankstumo, optimizavimo, žmogaus gerovės ir nuosavybės teisės principų pažeidimus.

Raktiniai žodžiai: apgaulingi šablonai, įtaigi kompiuterija, etikos principai, panaudojamumas, panaudojamumo eistikos, naudotojo potyrių projektavimo gairės.

1 Įvadas

Apgaulingas šablonas yra piktybiška vartotojo sąsaja, verčianti vartotojus priimti jiems nenaudingus sprendimus [1]. Juose siekiama padidinti verslo efektyvumą išnaudojant vartotojų psichologijos savybes vartotojo sąsajose įgyvendinant apgaulingą funkcionalumą [2]. Apgaulingi šablonai kuriami remiantis įtaigosios kompiuterijos principais (angl. *persuasive computing*). Įtaigioji kompiuterija, dar vadinama kaptologija, apima kompiuterinių sistemų projektavimo principus, skirtus įtakoti vartotojų elgesį ir nuostatas [3]. Nors įtaigūs projektavimo metodai savaime nėra kenksmingi, tačiau sumanus pažinimo mokslo taikymas suteikia galimybę jais piktnaudžiauti, pavyzdžiui, paslepiant vartotojui svarbią informaciją. Tai paskatino mokslininkus tirti vartotojo potyrių (angl. *user experience*) projektavimo etikos problemą.

Apgaulingus šablonus galima analizuoti tiek informatikos teisės, tiek vartotojo potyrių projektavimo požiūriu. Šiame darbe apgaulingų šablonų analizuojami vartotojo potyrių projektavimo požiūriu, o teisės ir elektroninių nusikaltimų analizė nėra šio tyrimo objektas.

Apgaulingų šablonų kontekste tarp amoralumo ir neteisėtumo yra siaura riba. Pavyzdžiui, informacijos slėpimas ar daiktų įtraukimas į vartotojų krepšelį be jų sutikimo laikomas ne tik neetišku, bet ir neteisėtu. Tuo tarpu apsunkintas paskyros ištrynimasis ar prenumeratos atšaukimas nėra neteisėtas, bet vis tiek yra neetiška.

Sunku pamatuoti, ar projektavimo strategija yra neetiška, ar pažeidžia įstatymus, todėl daugelyje tyrimų apgaulingi šablonai yra identifikuojami ir kategorizuojami. Pavyzdžiui, Moser ir kt. [4] išanalizavo 200 populiariausių el. prekybos svetainių, kurioje nustatė manipuliatyvius vartotojo sąsajos elementus, skatinančius pirkimus. Panašiai Mathur ir kt. [5] nustatė, kad 11 proc. populiariausių el. prekybos programų naudoja apgaulingus šablonus. Be to, autoriai sukūrė taksonomiją, kurioje identifikuoti apgaulingi šablonai suskirstyti į agresyvius, vidutiniškus ir švelnius.

Gerai suprojektuotos vartotojo sąsajos turi palengvinti naudotojų sąveiką su sistemomis, padėti jiems priimti sprendimus ir pasiekti savo tikslą. Apgaulingų projektavimo šablonų naudojimas verslo požiūriu gali būti pelningas, tačiau jie yra ant moralės ir teisėtumo ribos, dažnai prieštarauja vartotojų norams. Todėl tokios vartotojo sąsajos giliai paveikia žmonių emocijas ir elgesį [2]. Geriausiu atveju apgaulingi šablonai erzina vartotojus. Blogiausiu – klaidina ir apgauna vartotojus, pvz., sukeldami finansinius nuostolius, apgaulingai skatindami atiduoti asmeninius duomenis arba paskatindami priverstinį ir priklausomybę sukeliantį elgesį tiek suaugusiems, tiek vaikams [5]. Tai rimta šiuolaikinės visuomenės problema. Ją spręsti galima dviem būdais:

- a) siekiant sumažinti jos poveikį, interneto vartotojai būtų mokomi atpažinti apgaulingus šablonus jų naudojamose svetainėse ir programose. Tai padarytų apgaulingus šablonus neveiksmingus, nes tam, kad galėtų pasiekti savo tikslą, manipuliacija turi likti nepastebėta.
- b) pertvarkyti apgaulingus šablonus, kad būtų išvengta etikos pažeidimų ir manipuliacijų. Tačiau po pertvarkymo jie turėtų išlikti pajėgūs pasiekti konkrečius verslo tikslus.

Apgaulingų šablonų klasifikacija yra jau sukurta, tačiau vis dar reikia išgryninti atributus, kurie leistų nustatyti, ar kuriamą sąsają galima priskirti

apgaulingam šablonui. Be to, nėra metodikos, kuri galėtų padėti kūrėjams sukurti vartotojo sąsają, kuri padėtų kūrėjams pasirinkti etiškas apgaulingų šablonų alternatyvas, nemanipuliuodama ir neapgaudama naudotojų.

Tyrimo tikslas yra nustatyti atributus, apibrėžiančius apgaulingus šablonus, ir sukurti vertinimo metodiką sistemos kūrėjams, kuri padėtų identifikuoti etinio projektavimo gairių pažeidimus. Antrame skyriuje pateikiama susijusios literatūros apžvalga. Toliau nagrinėjamas vertinimo metodas. Galiausiai pateikiamos išvados.

2 Literatūros apžvalga

Įtaigių produktų kūrimo įrankiai tampa vis labiau prieinami – internetinių vaizdo įrašų, socialinių tinklų, metrikos ir kitos naujovės [6]. Dėl to daugiau asmenų ir organizacijų projektuoja patirtis, kurios jie tikisi, turės įtakos žmonių elgesiui. Tačiau daugelis įtaigaus projektavimo bandymų yra nesėkmingi, nes projektuotojai nesupranta, kokie veiksniai lemia elgesio pokyčius.

Susidūrus su etinėmis dilemomis, susijusiomis su technologijų kūrimu, dažnas metodas yra etinių gairių ar principų kūrimas [3]. Juose siūlomi profesinės etikos principai, kurių pagrindinis teigia, kad įtikinėjimas turi būti etiškas, auditorija turi būti informuota apie įtikinėtojų ketinimus. Du kiti principai akcentuoja įtaigius projektuotojo ketinimus: jie turi būti laikomi etiškais ir netechnologiniame kontekste, o technologijų kūrėjai neturėtų siekti įtikinti kitų tuo, kuo jie patys nepatikėtų. Kiti principai yra susiję su įtaigios technologijos poveikiu privatumui: jos kūrėjai turėtų saugoti naudotojų privatumą lyg savo ir technologijos, perduodančios informaciją trečiosioms šalims, turėtų būti ypač kruopščiai tikrinamos. Galiausiai, du principai yra susiję su rezultatais: pirma, numatyti rezultatai turi būti tokie, kad jie būtų etiški ir be įtikinimo; antra, įtaigių technologijų kūrėjai turi numatyti ir prisimti atsakomybę už manipuliacijomis pasiektus rezultatus.

Nors principai yra naudingi, jų dažnai nepakanka. Daugelis problemų atsiskleidžia tik stebint naudotojų sąveiką su sistema. Taip atskleistas aštuntasis reikalavimas – minimalus blaškymas. Mokslo ir technologinių studijų tyrimai akcentuoja projektavimo etikos charakteristikas ir jų įtaką suprojektuotų rezultatų vertei. Nors sukurta įvairių vertybėmis grindžiamų metodų, jie nėra plačiai naudojami projektuotojų ir neišku, kaip šiuos metodus pritaikyti konkrečiame projekte. Vertybėmis grindžiamas projektavimas (angl. *value-sensitive design*) yra tokių metodų pavyzdys [7]. Pabrėžiami būdai,

kuriais projektuotojai vertybes taiko savo darbuose bei keliami klausimai į kokias ir kieno vertybes reikėtų atsižvelgti [8]. Etiško projektavimo principai apibendrina kuriamų sistemų etinius ir panaudojamumo atributus [9]:

1. Teisė į informaciją – rodoma visa naudotojui reikalinga informacija (E1).
2. Panaudojamumas – naudotojas skatinamas sėkmingai atlikti užduotis (E2).
3. Saugumas – atsižvelgiama į naudotojo privatumą ir sistemos saugumą (E3).
4. Lankstumas – suteikiama galimybė naudotojui atlikti užduotį keliais būdais. (E4)
5. Optimizavimas – projektavimo komanda gali naudoti skirtingas vertybes, kad pasiektų konkretų projektavimo rezultatą. Projektuotojai yra atsakingi už vertybių susiejimą su galimais sprendimais (E5).
6. Žmogaus gerovė – ši vertybė nusako fizinės, materialinės ir psichologinės naudotojų gerovės užtikrinimo sistemoje klausimus (E6).
7. Nuosavybės teisė yra susieta su teise į informacija ir jo nuosavus objektus (E7).

Apibendrinus apgaulingus šablonus, jie kategorizuoti į šias grupes [1], [10], [11]:

1. Kabinėjimasis (angl. *nagging*) – daugkartinis atitraukimas nuo einašios vartotojo užduoties.
2. Trukdymas (angl. *obstruction*) – našumo mažinimas blokuojant užduočių eigą, pavyzdžiui, apsunkinant kainų palyginimą arba verslui nenaudingą vartotojo paskyros panaikinimą.
3. Slėpimas (angl. *sneaking*) – slepiamos papildomos išlaidos, įtraukimas prekių į krepšelį be vartotojo žinios, priverstinis mokamos prenumeratos tęstinumas.
4. Sąveikos trukdžiai (angl. *interface interference*) – vartotojo sąsajos manipuliacijos, slepia
5. Priverstinis veiksmas (angl. *forced action*) – vertimas pridėti draugus, pateikti privačius duomenis ir pan.

Literatūros šaltinių analizėje nustatyti apgaulingi šablonai ir etikos principai, kuriais remiantis yra sukurti vertinimo kriterijai.

3 Apgaulingų šablonų panaudojamumo principų pažeidimo analizė

Standarte ISO 9241:2018 apibrėžta, kad panaudojamumas – tai naudotojo veiklos efektyvumas, našumas ir jaučiamas malonumas, su kuriais konkretus naudotojas gali pasiekti konkrečių tikslų konkrečiose naudojimo aplinkose. Remiantis šiuo apibrėžimu apgaulingiems šablonams vertinti suformuotas šis kriterijų rinkinys: efektyvumas (P1) – pasiekiamų tikslų užbaigtumas ir tikslumas; suteikiamos tinkamos funkcijos ir tinkama informacija: išmokstamumas (P1a) - mokymosi arba užduoties trukmė, įsimenamumas (P1b), klaidos (P1c) – skaičius, dažnis, tipas; našumas (P2) – kiek pastangų ir resursų reikalauja tikslo pasiekimas bei jaučiamas malonumas (P3) – nuomonė apie sistemą, pakartotiniai apsilankymai.

Literatūros analizėje surinkti apgaulingi šablonai yra nagrinėjami etikos ir panaudojamumo principų požiūriu, siekiant nustatyti, ką konkrečiai apgaulingi šablonai pažeidžia (1 lentelė).

1 lentelė. Etikos principų siejimas su etikos vertybėmis

Šablonas	Etikos principai								Panaudojamumo principai					
	E1	E2	E3	E4	E5	E6	E7	E8	P1	P1a	P1b	P1c	P2	P3
1. Kabinėjimasis	○	●	○	○	○	●	○	○	●	●	○	●	●	●
2. Trukdymas														
a) Tarpinė valiuta	○	●	○	○	○	●	○	○	○	○	○	○	●	●
b) Kainų palyginimo prevencija	○	●	○	○	○	●	○	○	●	○	○	○	●	●
c) „Roach Motelis“	○	●	○	○	○	●	○	○	●	○	○	○	●	●
3. Slėpimas														
a) Viliojimas ir sukeitimas slapta	○	●	○	○	○	●	○	○	○	○	○	○	○	●
b) Paslėptos išlaidos	○	●	○	○	○	●	○	○	●	○	○	○	○	●
c) Slaptas įtraukimas į krepšelį	○	●	○	○	○	●	○	○	●	○	○	○	○	●
d) Priverstinis tęstinumas	○	●	○	○	○	●	○	○	●	○	○	○	○	●

Šablonas	Etikos principai								Panaudojamumo principai						
	E1	E2	E3	E4	E5	E6	E7	E8	P1	P1a	P1b	P1c	P2	P3	
4. Sąveikos trukdžiai															
a) Paslėpta informacija	●	●	○	○	○	○	○	○	○	○	○	○	○	●	●
b) Išankstinis pasirinkimas	●	●	○	●	○	●	○	○	○	○	○	○	○	○	●
c) Emocinis manipuliavimas	○	●	○	○	○	○	○	○	●	○	○	○	○	○	●
d) Netikra hierarchija	○	●	●	○	○	○	○	○	●	○	○	○	○	○	●
e) Užmaskuota reklama	●	○	○	○	○	●	○	○	○	○	○	○	○	●	●
f) Kėblūs klausimai	○	●	○	○	○	○	○	○	●	○	○	○	○	●	●
5. Priverstinis veiksmas															
a) Socialinė piramidė	●	○	○	○	○	○	○	○	○	○	○	○	○	●	○
b) Privačių duomenų išviliojimas	○	●	●	○	○	○	○	○	○	○	○	○	○	○	●
c) Žaidimai	●	○	○	○	○	○	○	○	○	○	○	○	○	●	○

Išanalizavus apgaulingų šablonų etikos, panaudojamumo principų ir euristikų [12] ir naudotojų potyrių gairių [13] pažeidimus, gautas apgaulingų šis apgaulingų šablonų nustatymą padedančių klausimų rinkinys:

1. Klausimai susiję su etikos principais ir vertybėmis:

- E1. Ar naudotojams teikiama informacija apie sistemos kūrėjų ketinimus?
- E2. Ar sistemos kūrėjų ketinimai būtų etiški ir netechnologiniame kontekste?
- E3. Ar sistemos kūrėjai nebando įtikinti naudotojų tuo, kuo patys nepatikėtų?
- E4. Ar naudotojų privatumas saugomas nemažiau nei sistemos kūrėjų?
- E6. Ar numatyti sistemos rezultatai būtų etiški ir be įtikinimo?

2. Klausimai susiję su panaudojamumo principais:
 - P1. Ar naudotojui suteikiamos tinkamos funkcijos ir informacija tikslui pasiekti?
 - P1a. Ar naudotojo užduoties atlikimas trunka ne ilgiau nei to tikisi naudotojas?
 - P1c. Ar dažnai pasitaiko naudotojų klaidos?
 - P2. Ar naudotojo tikslo siekimas reikalauja daugiau pastangų ir resursų nei reikia?
 - P3. Ar dažni naudotojų pakartotiniai apsilankymai?
3. Klausimai susiję su panaudojamumo euristikomis:
 - H1. Ar sistema nuolat naudotojui teikia grįžtamąjį ryšį apie sistemos būseną ir jos pasikeitimus?
 - H2. Ar informacija pateikiama naudotojui suprantama kalba, žodžiais, sąvokomis ar frazėmis, nenaudodama sistemos terminų?
 - H3. Ar naudotojui suteikiama galimybė ištaisyti padarytą klaidą be papildomų dialogo langų?
 - H4. Ar skirtingose sistemos vietose pasikartojantys žodžiai reiškia tą pačią funkciją?
 - H5. Ar naudotojas yra apsaugotas nuo suklydimų?
 - H6. Ar objektai, veiksmai ir pasirinkimai yra aiškiai matomi ir naudotojui nereikia jų prisiminti?
 - H7. Ar sistema leidžia naudotojams pritaikyti naudotojo sąsają jų poreikiams?
 - H8. Ar sistemos dialogo languose nėra naudotojui neaktualios informacijos?
 - H9. Ar klaidų pranešimai yra suprantami, tiksliai nurodant problemą ir pateikiant sprendimą?
4. Klausimai susiję su naudotojų potyrių projektavimo gairėmis:
 - G1. Ar sistemai yra taikoma tinkamiausia konfigūracija pagal numatytuosius nustatymus?
 - G2. Ar sistema automatiškai įsimena, kas naudotojui patinka ir automatiškai tai atkartoja kitą kartą?
 - G4. Ar sistema automatiškai atlieka už naudotojus tai, ką įmanoma atlikti?
 - G6. Ar naudotojo atliekama užduotis nėra pertraukiama verčiant ją papildomai mąstyti?
 - G7. Ar iš naudotojo nėra prašoma perteklinio patvirtinimo?

4 Išvados

Darbe apžvelgti apgaulingi šablonai: kabinėjimasis, trukdymas, slėpimas, sąsajos trukdžiai, priverstinis veiksmas. Identifikuoti šių šablonų atributai, padedantys identifikuoti apgaulingus elementus.

Literatūros analizėje nustatyta, kad apgaulingi šablonai pažeidžia šias etikos vertybes ir principus: teisė į informaciją, panaudojamumas, saugumas, lankstumas, optimizavimas, žmogaus gerovę ir nuosavybės teises. Kuriant vertinimo gaires pastebėta, kad visi tirti apgaulingi šablonai pažeidžia bent vieną etikos ir bent vieną panaudojamumo principą. Darbe tirti apgaulingi šablonai pažeidžia daugumą panaudojamumo euristicų. Visi šablonai pažeidžia bent vieną panaudojamumo euristicą, kas atskleidžia glaudų panaudojamumo ir etiško projektavimo principų santykį.

Literatūra

- [1] Di Geronimo, L., L. Braz, E. Fregnan, F. Palomba & A. Bacchelli. (2020). UI Dark Patterns and Where to Find Them: A Study on Mobile Applications and User Perception. CHI ,20: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems,
- [2] Maier, M. Dark patterns – An end user perspective. Human-Computer Interaction and Social Media, 2019. <<http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1330920&dswid=-2205>>
- [3] Davis, J. (2009). Design Methods for Ethical Persuasive Computing. Persuasive ,09: Proceedings of the 4th International Conference on Persuasive Technology. Article No.: 6, 1–8. <<https://dl.acm.org/doi/pdf/10.1145/1541948.1541957>>
- [4] Moser, C., S. Y. Schoenebeck & P. Resnick. (2019). Impulse Buying: Design Practices and Consumer Needs. CHI '19: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Article no. 242, 1–15. <<https://dl.acm.org/doi/10.1145/3290605.3300472>>
- [5] Mathur, A. , G.A car, M. J. Friedman, E. Lucherini, J. Mayer, M. Chetty & A. Narayanan. (2019). Dark Patterns at Scale: Findings from a Crawl of 11K Shopping Websites. Proceedings of the ACM on Human-Computer Interaction. Article no. 81, 1–32. <<https://doi.org/10.1145/3359183>>
- [6] Fogg, B.J. (2009). A Behavior Model for Persuasive Design. Proceedings of the 4th International Conference on Persuasive Technology (Persuasive '09), Article no. 40, 1–7. <https://dl.acm.org/doi/abs/10.1145/1541948.1541999?casa_token=jzvU3an21SQAAAAA:Xp_Esj5hqv-AsLwFv6v-01D7VQqGhaZ2jaEK6sMoswisKdGQWAGLqLesmDum9oqGXpfDmNm8GkA>
- [7] Fordyce, S. (2020). Value Sensitive Design: Shaping Technology with Moral Imagination: by Batya Friedman, and David G. Hendry, Cambridge, MA: The MIT Press, 2019, 229 pp. ISBN 9780262039536. \$40.
- [8] Gray C. M. & E. Boling. (2016). Inscribing ethics and values in designs for learning: a problematic. Educational Technology Research and Development: ETR & D., 64, 969–1001. <<https://doi.org/10.1007/s11423-016-9478-x>>

- [9] Chivukula, S. S., C. M. Gray & J. A. Brier. (2019) Analyzing Value Discovery in Design Decisions Through Ethicography. Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Article no. 77, 1–12. <<https://doi.org/10.1145/3290605.3300307>>
- [10] Chromik, M., Eiband, M., Völkel, S. T., & Buschek, D. (2019, March). Dark Patterns of Explainability, Transparency, and User Control for Intelligent Systems. In IUI workshops (Vol. 2327). <<http://ceur-ws.org/Vol-2327/IUI19WS-ExSS2019-7.pdf>>
- [11] Mathur, A. (2020). Identifying and measuring manipulative user interfaces at scale on the web. Princeton University, PhD dissertation <<https://dataspace.princeton.edu/handle/88435/dsp012f75rc09f>>
- [12] Nielsen, J. (1994). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods, John Wiley & Sons, New York, NY.
- [13] Platt. D. (2016). The Joy of UX: User Experience and Interactive Design for Developers. Addison-Wesley Professional.

Viršelio dailininkė *Jurga Tėvelienė*
Maketuotoja *Vida Vaidakavičienė*

Vilniaus universiteto leidykla
Saulėtekio al. 9, LT-10222 Vilnius
info@leidykla.vu.lt, www.leidykla.vu.lt
Knygos internete: www.knygynas.vu.lt
Mokslo periodikos žurnalai: www.zurnalai.vu.lt

