Steganografija dirbtinių neuroninių tinklų parametruose ir jos aptikimas mašininio mokymosi metodais

Pranas Žeromskas, Viktor Medvedev

Vilniaus universitetas, Matematikos ir informatikos fakultetas,
Duomenų mokslo ir skaitmeninių technologijų institutas,
Akademijos g. 4, LT-08412, Vilnius, Lietuva
pranas.zeromskas@mif.stud.vu.lt

Santrauka. Dirbtinių neuroninių tinklų modeliais plačiai dalijamasi viešosiose saugyklose, todėl kyla grėsmė, kad jų parametrų failai gali būti išnaudoti kenkėjiškam turiniui slėpti ir platinti. Šiame straipsnyje analizuojamas erdvinės srities steganografijos pritaikymas dirbtiniuose neuroniniuose tinkluose, vertinant tiek informacijos paslėpimo, tiek jos aptikimo galimybes. Nustatyta, kad vaizdų klasifikavimo modeliuose galima modifikuoti bent pusę jų talpos reikšmingai nepabloginant klasifikavimo tikslumo. Įterpto turinio aptikimui taikyti mašininio mokymosi metodai, su kuriais pasiektas 90,2 % tikslumas modifikacijų aptikimo užduotyje. Rezultatai rodo, kad tinklų parametruose galima paslėpti informaciją, tačiau tokios modifikacijos gali būti aptinkamos.

Raktiniai žodžiai: neuroninių tinklų steganografija, modelių parametrai, LSB steganografija, steganografijos aptikimas, neuroninių tinklų saugumas.

1 Įvadas

Per pastaruosius metus dirbtinis intelektas tapo neatsiejama kasdienybės dalimi tiek profesinėje, tiek asmeninėje aplinkoje. Dirbtiniais neuroniniais tinklais paremti modeliai plačiai naudojami informacijos apdorojime, medicinoje, ekonomikoje ir daugelyje kitų sričių. Kadangi modelių apmokymas yra daug laiko ir skaičiavimo resursų reikalaujanti procedūra, įvairiose viešosiose saugyklose galima rasti ir atsisiųsti jau apmokytų modelių parametrus, o tai atveria duris juos išnaudoti kaip kenkėjiškos programinės įrangos nešiklius per modelių tiekimo grandines (angl. *supply chain attacks*).

Šio darbo tikslas – ištirti, kiek neuroninių tinklų parametrų bitų gali būti modifikuota, nepaveikiant modelio funkcionalumo, bei patikrinti, ar modifikuoti modeliai gali būti aptinkami naudojant mašininio mokymosi metodus.

DOI: https://doi.org/10.15388/LMITT.2026.35

2 Literatūros apžvalga

Steganografija – informacijos įterpimas kitame informacijos pakete taip, kad paslėptos informacijos buvimas nebūtų akivaizdus. Perduodama paslaptis (angl. *payload*) yra įterpiama į nešiklį (angl. *cover object)*, nepakeičiant pradinio nešiklio funkcionalumo. Tam išnaudojami duomenų formatai, turintys daug perteklinės informacijos, pavyzdžiui, paveikslai, vaizdo ar garso įrašai. Nešiklyje esanti nereikšminga informacija yra keičiama slaptąja žinute [1]. Tokia praktika dažnai naudojama teisėtiems tikslams, pavyzdžiui, autorinių teisių apsaugai, tačiau ją taip pat galima naudoti ir kenkėjiškomis paskatomis – perduodant kodą, failus ar kenkėjišką programinę įrangą (angl. *malware*) į aukos kompiuterį, išvengiant antivirusinių programų aptikimo.

Praktikoje aptinkamos erdvinės srities (angl. *spatial-domain*) arba transformacijos srities (angl. *transform-domain*) steganografijos rūšys. Erdvinės srities metodai tiesiogiai keičia duomenis binarinėje duomenų reprezentacijoje. Dažnai naudojamas mažiausiai reikšmingų bitų (angl. *LSB – Least Significant Bit*) metodas, keičiantis paskutiniuosius bitus, minimaliai paveikiančius galutinį rezultatą. Pavyzdžiui, naudojant paveikslėlius kaip nešiklį, pikselių spalvų binarinės reprezentacijos paskutiniai bitai koreguojami, keičiant juos į pernešamo turinio fragmentus [2]. Transformacijos srities metodai nešiklį transformuoja, pavyzdžiui, į dažnių sritį ir turinį įterpia į koeficientus. Tokie metodai yra sudėtingesni, tačiau jie yra žymiai sunkiau aptinkami, atsparesni nešiklio modifikacijoms [2].

Nors dalis modernių dirbtinių neuroninių tinklų įgyvendinimų naudoja kvantuotus (angl. *quantized*) parametrų formatus, siekiant sumažinti modelio dydį, tradiciškai jų reikšmės dar dažnai saugomos 32 bitų slankiojo kablelio formatu (*IEEE* 754). Kadangi neuroninių tinklų svoriai po apmokymo dažniausiai yra arti nulio, dauguma jų bitų saugo trupmeninę skaičiaus dalį. Nustačius pirmąjį parametro baitą (ženklo bitą ir 7 iš 8 laipsnio rodiklio bitų), skaičiaus reikšmė apribojama siaurame intervale, o keičiant paskutiniuosius trupmeninės dalies bitus, skaičiaus reikšmė kinta labai mažose ribose.

Šios savybės leidžia dirbtinių neuroninių tinklų parametrus išnaudoti kaip steganografijos nešiklius. Įvairios modifikacijų strategijos leidžia įterpti nuo keleto iki 24 bitų kiekviename arba konkrečių sluoksnių modelio parametruose, visiškai nepaveikdamos arba minimaliai neigiamai paveikdamos modelių funkcionalumą [3, 4, 5].

Vieni dažniausiai naudojamų steganografijos nešiklių – paveikslėliai – pasižymi tuo, kad juose saugoma informacija pasižymi aiškia struktūra ir sti-

pria gretimų pikselių koreliacija, todėl erdvinės srities steganografija palieka statistinius požymius, kuriuos galima aptikti įvairiais analizės metodais, o kartais – ir plika akimi [6]. Tuo pačiu, paveikslėlių failų užimama vieta diske yra sąlyginai maža, ypač naudojant stipriai suspaustus formatus (pavyzdžiui, *JPEG*), taigi ir pernešamo turinio dydis negali būti didelis.

Tuo tarpu dirbtinių neuroninių tinklų parametrų reikšmės po apmokymo turi didelę entropiją, nėra ryškių pasiskirstymo požymių, taigi paprastais statistiniais metodais aptikti atliktas modifikacijas gali būti sunkiau. Kadangi modelių parametrų failai įprastai yra kelis šimtus ar daugiau kartų didesni už paveikslėlius, atliekamos modifikacijos gali būti konservatyvesnės, t. y. modifikuojama maža parametro reikšmės dalis, greta esantys turinio baitai/bitai įterpiami nutolusiuose modelio parametruose ir taip toliau. Dėl to steganografijos, naudojančios dirbtinių neuroninių tinklų parametrus kaip nešiklį, aptikimas tampa sudėtingesne užduotimi.

3 Metodinė dalis

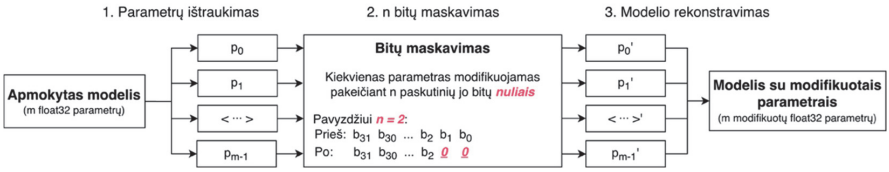
Tyrimui pasirinktas platus vaizdų klasifikavimo modelių spektras: klasikiniai konvoliuciniai neuroniniai tinklai (KNT, angl. *Convolutional Neural Networks, CNN*), liekanų KNT (angl. *Residual CNN*), daugiašakiai KNT (angl. *Multibranch CNN*), efektyvieji KNT (angl. *Efficient CNN*), tankieji KNT (angl. *Dense CNN*), modernūs KNT (angl. *Modern CNN*), vaizdų transformeriai (angl. *Vision transformers*), Swin transformeriai (angl. *Swin transformers*). Visų modelių įgyvendinimai ir iš anksto apmokyti parametrai paimti iš *PyTorch* bibliotekos *torchvision.models* paketo[1].

Parametrų modifikavimo įtakai įvertinti modeliai testuojami naudojant 2012–2017 m. *ImageNet* vaizdų atpažinimo konkurso (angl. *Large Scale Visual Recognition Challenge, ILSVRC*) duomenų aibę[2], sudarytą iš 50 tūkst. paveikslų, suskirstytų į 1 tūkst. klasių.

3.1 Neveiksnių bitų aptikimas

Siekiant nustatyti, kurie parametrų bitai gali būti modifikuojami nepaveikiant modelio veikimo, analizuojama parametrų bitų maskavimo įtaka klasifikavimo tikslumui. Visų modelio svorių *n* paskutiniai bitai keičiami į nulius (žr. 1 pav.), pradedant nuo mažiausiai reikšmingų. 32 bitų slankiojo kablelio

[1] *PyTorch torchvision.models* dokumentacija. https://docs.pytorch.org/vision/stable/models.html
[2] *ImageNet ILSVRC* duomenų aibė. https://image-net.org/challenges/LSVRC/



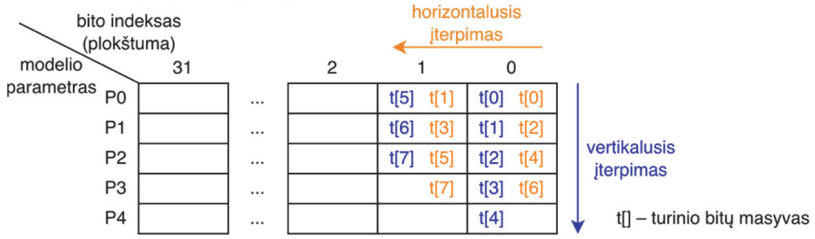
1 pav. Modelio parametrų bitų maskavimo procedūra.

formatu (*IEEE* 754) saugomuose parametruose nuosekliai maskuojama nuo 1 iki 32 bitų, o po kiekvieno pakeitimo vertinamas modelio klasifikavimo tikslumas.

Parametro bitas laikomas neveiksniu, jei jį ir visus mažiau reikšmingus bitus pakeitus į nulį, modelio Top1 tikslumas sumažėja ne daugiau kaip 1 procentiniu punktu, lyginant su originaliu modeliu.

3.2 Turinio įterpimo procedūra

Turinio įterpimo į neuroninius tinklus metodų galima sugalvoti labai daug – išskleisti skirtingomis kryptimis, naudoti įvairius maišos metodus, slėpti tik konkrečiuose modelių sluoksniuose ar jų dalyse. Šiame darbe nagrinėjamos dvi paprastos turinio įterpimo procedūros: vertikalioji ir horizontalioji (žr. 2 pav.).



2 pav. Turinio įterpimo procedūros.

Vertikalioji procedūra veikia paeiliui keičiant paskutinį kiekvieno parametro bitą turinio bitu. Jei turinio bitų skaičius viršija modelio parametrų skaičių, užpildžius paskutinius visų parametrų bitus, pradedami keisti prieš-paskutiniai bitai. Procesas kartojamas, kol nebelieka neįterptų turinio bitų.

Horizontalioji įterpimo procedūra pirmiausia apskaičiuoja, kiek kiekvieno parametro bitų reikia pakeisti: k = turinio bitų skaičius / modelio parame-

trų skaičius. Tuomet einama per kiekvieną modelio parametrą, o paskutiniai *k* jo bitai pakeičiami turinio bitais.

3.3 Įterpto turinio aptikimas

3.3.1 Bitų sekų klasifikavimas

Pirmojoje aptikimo užduotyje siekiama nustatyti, ar įmanoma atskirti įprastų failų segmentus nuo iš dirbtinių neuroninių tinklų svorių ištrauktų bitų segmentų, naudojant bitų lygmens savybes. Šį eksperimentą galima apibūdinti taip: turime dirbtinį neuroninį tinklą ir žinome, kokia turinio įterpimo procedūra galėjo būti panaudota. Jeigu atliktume įterpimui atvirkštinį veiksmą ir iš svorių ištrauktume bitų eilutę, ar būtų įmanoma nustatyti, ar ši bitų eilutė atitinka natūralias modelio mokymo metu susiformavusias reikšmes, ar ji labiau panaši į failo iškarpą? Užduočiai sudarytas duomenų rinkinys, kurį sudaro 2000 bitų ilgio iškarpos iš paveikslėlių failų, tekstinių dokumentų, Windows vykdomųjų programų (angl. *Windows executable files, exe*) bei iš dirbtinių neuroninių tinklų svorių ištrauktos bitų sekos, taikant vertikaliojo ir horizontaliojo ištraukimo procedūras. Kiekvienam iš šių įrašų apskaičiuotos šios savybės:

- vienetų bitų dalis visoje sekoje;
- bitų kitimo dažnis;
- vienodų bitų sekų ilgio statistikos (vidurkis, standartinis nuokrypis, ilgiausia seka, entropija);
- bitų pasiskirstymo entropija;
- vienetų tankio svyravimai slenkančiuose languose;
- dažninių komponentų charakteristikos (diskrečioji Furjė transformacija);
- 2 ir 3 bitų šablonų dažniai.

Gautas požymių vektorius naudojamas apmokyti atsitiktinių miškų (angl. *Random Forest*), papildomų medžių (angl. *Extra Trees*), logistinės regresijos (angl. *Logistic Regression*) ir *XGBoost* klasifikatorius. Klasifikatoriaus užduotis – požymių vektoriui priskirti vieną iš keturių klasių: paveikslėlis, tekstinis dokumentas, programa arba neuroninis tinklas.

3.3.2 Modelio parametrų grupių klasifikavimas

Kadangi praktinėje situacijoje įterpimo procedūra dažniausiai nėra žinoma, papildomai nagrinėjamas tiesioginis modelio parametrų klasifikavimas. Šio-

je užduotyje iš modelio parenkamos 1024 iš eilės einančių parametrų grupės ir siekiama nustatyti, ar jos buvo modifikuotos. Jei grupė modifikuota, papildomai nustatomas įterpto turinio tipas ir naudota įterpimo procedūra.

Paruoštą duomenų rinkinį sudaro atsitiktinės parametrų grupės iš įvairių paveikslėlių klasifikavimo modelių (aprašytų skyriaus pradžioje). Dalis grupių paliekamos nemodifikuotos ir sudaro švarių duomenų klasę, o likusiose grupėse įterpiamas failų turinys taikant skirtingas įterpimo procedūras. Nagrinėjamos vertikalioji ir horizontalioji procedūros, kai į kiekvieną parametrą įterpiama po 1, 4 arba 16 bitų. Vieno bito atveju vertikalioji ir horizontalioji procedūros sutampa, todėl gaunami penki skirtingi modifikacijos variantai.

Kiekviena parametrų grupė interpretuojama kaip 32×1024 bitų matrica, kur eilutės atitinka bito poziciją *IEEE* 754 reprezentacijoje, o stulpeliai – nuoseklius modelio parametrus.

Kiekvienam duomenų rinkinio įrašui apskaičiuojamos 3.3.1 skyrelyje aprašytos bitų sekų savybės, jas pritaikant ir papildant, papildomai įvertinant jų pasiskirstymą skirtingose bito pozicijose ir tarp gretimų parametrų, siekiant išnaudoti dvimatę įrašo struktūrą. Gautas požymių vektorius naudojamas apmokyti 3.3.1 skyrelyje pateiktus klasifikatorius. Šie modeliai sprendžia modifikuotų parametrų aptikimo, įterpto turinio tipo bei naudotos įterpimo procedūros identifikavimo užduotis.

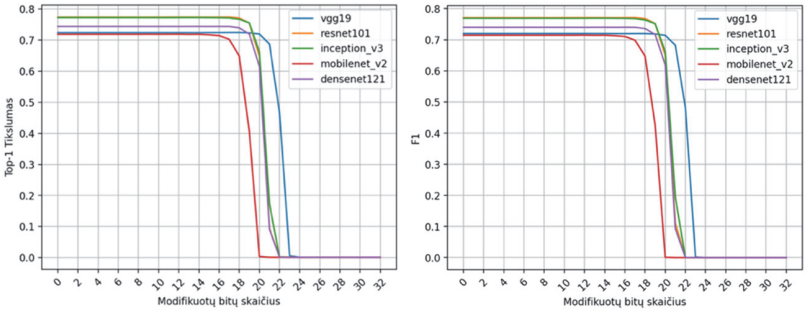
Ta pati užduotis sprendžiama ir naudojant konvoliuciniais neuroniniais tinklais grįstus klasifikatorius. Kiekvienas duomenų rinkinio įrašas (32×1024 dydžio bitų matrica) interpretuojama kaip vieno kanalo paveikslas, kurio pikselių reikšmės yra 0 arba 1.

Naudojamos 3 skirtingos architektūros:

1. bazinis KNT modelis (286 tūkst. parametrų), sudarytas iš kelių nuoseklių konvoliucinių sluoksnių, normalizacijos, aktyvacijos funkcijų ir sutelkimo (angl. *pooling*) sluoksnių, po kurių seka pilnai sujungtas klasifikatorius;
2. asimetriškas KNT modelis (1,448 mln. parametrų), kuriame naudojami daugiau eilučių nei stulpelių turintys konvoliucijų branduoliai, pavyzdžiui, 3×9, 3×7 ir 3×5, pritaikyti stačiakampės struktūros įrašams;
3. liekaniniais blokais paremta KNT architektūra (1,522 mln. parametrų), naudojanti trumpąsias jungtis tarp sluoksnių gilesniam modeliui stabiliai mokyti.

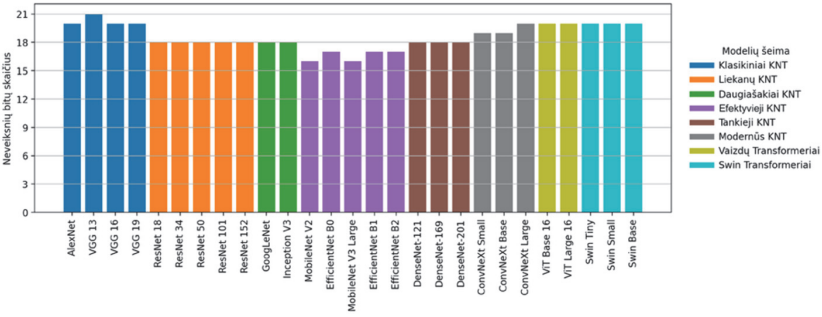
4 Eksperimentai ir rezultatai

Iš pradžių patikrinta detali klasifikavimo metrikų priklausomybė nuo modifikuotų parametrų bitų kiekio keliems modeliams. Grafikuose (žr. 3 pav.) pateikiamos klasifikavimo kokybę atspindinčios tikslumo ir F1 mato metrikos. Taip pat įvertintos preciziškumo ir atkūrimo metrikos rodo analogišką tendenciją. Skirtingos linijų spalvos atspindi konkrečius bandytus modelius.



3 pav. Modelio klasifikavimo tikslumo ir F1 priklausomybė nuo modifikuotų parametrų bitų skaičiaus.

Pirminis eksperimentas rodo, kad modifikuojant iki pusės parametro bitų klasifikavimo tikslumas, preciziškumas, atkūrimas ir F1 metrika beveik nekinta, tačiau kiekvienam modeliui galima identifikuoti lūžio tašką, kai vieno papildomo bito modifikavimas stipriai neigiamai paveikia visas klasifikavimo metrikas. Neveiksnių bitų skaičius modeliuose pateikiamas 4 pav.



4 pav. Neveiksnių bitų skaičius kiekvienam modeliui.

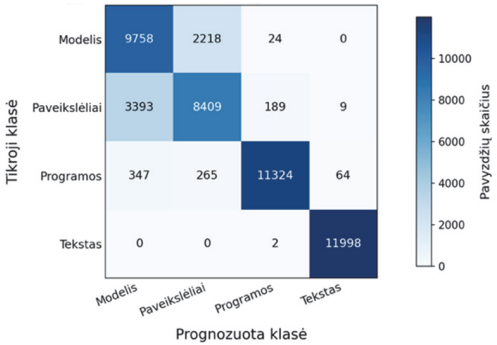
Visiems modeliams pavyko pakoreguoti bent 16 bitų prieš pastebint reikšmingą klasifikavimo tikslumo suprastėjimą. Mažiausią pakeistų bitų skaičių toleravo kompaktiškos architektūros mobilūs KNT modeliai (16–17). Galima pastebėti (žr. 4 pav.), kad kiekviena modelių šeima turi panašius neveiksnių parametro bitų skaičius, nepriklausančius nuo modelio parametrų skaičiaus, taigi didesnę įtaką daro architektūra, o ne modelio dydis.

Rezultatai rodo, kad ištirtuose modeliuose galima saugiai modifikuoti bent pusę svorio bitų (16 iš 32) ir naudoti juos steganografija paremtuose paslėpto turinio pernešimo metoduose, reikšmingai nepaveikiant modelio klasifikavimo tikslumo. Tai leidžia neuroninių tinklų parametrus naudoti kaip didelės talpos steganografijos nešiklį. Tirtuose modeliuose neveiksnioji talpa siekia nuo 7 iki 760 megabaitų.

4.1 Įterpto turinio aptikimas

4.1.1 Bitų sekų klasifikavimas

Šiame eksperimente vertinama 3.3.1 skyrelyje aprašyta bitų sekų klasifikavimo užduotis, kai įterpimo procedūra laikoma žinoma ir iš modelio svorių ištraukiamos bitų sekos. Klasifikatoriai apmokyti naudojant numatytuosius hiperparametrus, išskyrus keletą modifikacijų. Logistinei regresijai taikytas duomenų standartizavimas (*StandartScaler*), iteracijų kiekis padidintas iki 2000. Medžių ansamblių metodams (atsitiktinių miškų, papildomų medžių ir *XGBoost*) naudota 500 medžių. *XGBoost* papildomai nustatytas maksimalus gylis 6 ir mokymosi žingsnis 0,05. Visiems klasifikatoriams naudota fiksuota atsitiktinė pradžia (*random_state* = 42), duomenų rinkinio klasės subalansuotos.



5 pav. Testavimo rinkinio klasifikavimo matrica su XGBoost klasifikatoriumi.

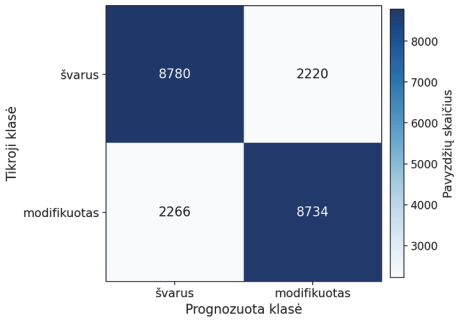
Rezultatai parodė, kad, naudojant išskirtas bitų lygmens savybes, galima gana patikimai klasifikuoti įrašus į 4 klases. Geriausią rezultatą pasiekė *XGBoost* klasifikatorius, kurio tikslumas testavimo duomenų aibei siekė 86,44 %. Atsitiktinių miškų klasifikatorius pasiekė 85,65 %, papildomų medžių – 85,24 %, o logistinės regresijos tikslumas siekė 80,91 %.

Klasifikavimo matrica (žr. 5 pav.) rodo, kad lengviausiai atskiriami tekstinių dokumentų ir vykdomųjų programų segmentai, o didžiausias persidengimas stebimas tarp paveikslėlių ir neuroninio tinklo svorių bitų sekų.

4.1.2 Aptikimas iš modelio parametrų grupių

Toliau vertinama 3.3.2 skyrelyje aprašyta užduotis, kai analizuojamos modelio parametrų grupės. Eksperimentuose vertinamos trys užduotys: modifikuotos grupės aptikimas (binarinė klasifikacija), įterpto turinio tipo klasifikavimas ir įterpimo procedūros identifikavimas. Metodinėje dalyje aprašytas duomenų rinkinys padalytas į mokymo, validavimo ir testavimo aibes santykiu 80/10/10, užtikrinant, kad tie patys modeliai ir įterpiami failai nepatektų į skirtingas aibes.

Iš klasikinių klasifikatorių (naudoti tie patys hiperparametrai kaip ir bitų sekų klasifikavimo uždavinyje), geriausi rezultatai visoms užduotims pasiekti su *XGBoost* modeliu. Binarinio atskyrimo užduotyje pavyko pasiekti 79,6 % tikslumą (klasifikavimo matrica pateikiama 6 pav.).



6 pav. Binarinės užduoties testavimo rinkinio klasifikavimo matrica su *XGBoost* klasifikatoriumi.

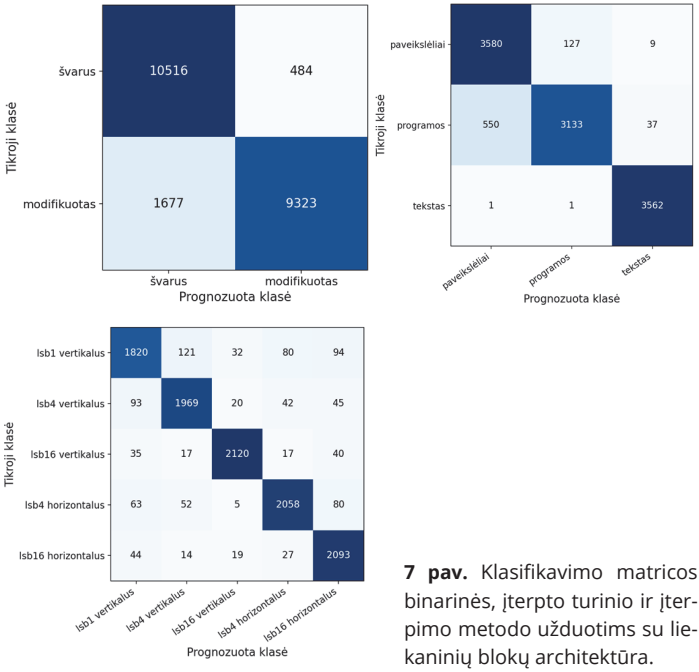
Tiksliausiai identifikuojamos grupės su įterptais tekstiniais dokumentais (1,8 % priskirti ne tai klasei), prasčiau klasifikuotos grupės su įterptomis programomis (17 %), o sunkiausiai aptinkamos grupės su įterptais paveikslėliais

(50,8 % priskirti švarių modelių klasei). Šie rezultatai atitinka tai, ką gavome ankstesniu eksperimentu: lengviausiai atskirti tekstinius dokumentus, sunkiausiai – paveikslėlius. Taip pat pastebėta, kad sudėtingiausia aptikti modifikacijas, kai keičiama mažiau parametrų bitų.

Klasifikuojant modifikuotų įrašų įterpto turinio tipą pasiektas 91,2 % tikslumas, o identifikuojant naudotą įterpimo procedūrą – 80,8 %.

KNT klasifikatoriai visose užduotyse pasiekė geresnius rezultatus nei klasikiniai klasifikatoriai. Modeliai buvo mokomi iki 40 epochų, naudojant 256 dydžio paketus. Optimizavimui naudotas *Adam* optimizatorius. Pradinis mokymosi greitis 0,0007 ir svorių mažinimo parametras (angl. *weight decay*) – 0,0001. Mokymosi greitis automatiškai mažinamas per pusę, jei tikslumas validacijos duomenų aibei negerėjo 3 epochas, su minimalia riba ties 10^{-6}. Taikytas ankstyvas stabdymas, jei rezultatai negerėjo 7 epochas.

Geriausius rezultatus demonstravo daugiausiai parametrų turinti liekaninių blokų architektūra (klasifikavimo matricos pateikiamos 7 pav.), pasiekusi 90,2 % tikslumą binarinio klasifikavimo užduotyje, 93,4 % klasifikuojant įterpto turinio tipą ir 91,5 % identifikuojant įterpimo procedūrą.



7 pav. Klasifikavimo matricos binarinės, įterpto turinio ir įterpimo metodo užduotims su liekaninių blokų architektūra.

Asimetrinė architektūra pasiekė 87,6 %, 93,3 % ir 91,2 % tikslumus atitinkamai. Bazinis KNT modelis pasiekė 83,8 % tikslumą binarinėje užduotyje, 92,6 % klasifikuojant turinio tipą ir 85,4 % identifikuojant įterpimo procedūrą.

Kaip ir klasikinių klasifikatorių atveju, sunkiausiai identifikuojamos grupės su įterptais paveikslėliais ir naudojančios mažiausiai bitų modifikuojančias procedūras.

KNT klasifikatorių pranašumui, tikėtina, įtaką turėjo tai, kad modeliai mokosi tiesiogiai iš parametrų bitų erdvinės struktūros. Tuo tarpu klasikiniai klasifikatoriai remiasi iš anksto apskaičiuotais požymiais, todėl dalis informacijos prarandama. Plečiant ir tobulinant požymių rinkinį, tikėtina, pavyktų pasiekti ir geresnių rezultatų, artimesnių konvoliuciniams neuroniniams tinklams, tačiau ateityje tiriant kitas įterpimo procedūras, dabar įvertinamų požymių nebeužtektų, taigi konvoliuciniai neuroniniai tinklai, nors reikalauja didesnio skaičiavimo resursų kiekio, ypač apmokymo metu, yra universaliau pritaikomi įterpto turinio aptikimo užduočiai.

5 Išvados

Šiame darbe tirtas dirbtinių neuroninių tinklų panaudojimas pernešant paslėptą informaciją erdvinės steganografijos principu. Nustatyta, kad visuose analizuotuose paveikslų klasifikavimo modeliuose galima modifikuoti bent 16 iš 32 parametro bitų reikšmingai nepabloginant klasifikavimo tikslumo. Šie nereikšmingi bitai gali būti keičiami į norimą pernešti turinį ir leidžia neuroninius tinklus išnaudoti kaip didelės talpos steganografijos nešiklius. Įvertinta, kad maksimalus paslėpto turinio dydis gali siekti nuo kelių iki kelių šimtų megabaitų viename modelyje priklausomai nuo architektūros ir parametrų kiekio.

Atliktos erdvinės srities modifikacijos, naudojančios vertikalųjį ir horizontalųjį įterpimo metodus, gali būti sėkmingai aptinkamos naudojant mašininio mokymosi metodus. Klasifikuojant analizuojamo modelio parametrų grupes į modifikuotas ir nemodifikuotas, klasikiniai mašininio mokymosi algoritmai pasiekė 79,6 % tikslumą, o konvoliuciniais neuroniniais tinklais grįsti modeliai – iki 90,2 %. Taip pat tiksliai pavyksta nustatyti ir įterpto turinio tipą bei naudotą įterpimo strategiją.

Šioms užduotims KNT klasifikatoriai buvo efektyvesni nei nuo ištrauktų statistinių požymių rinkinio priklausantys klasikiniai klasifikatoriai, nes išnaudojama pilna parametrų bitų erdvinė struktūra. Klasikiniams klasifikatoriams reikiamų požymių kiekis sparčiai didėja plečiant tiriamas įterpi-

mo procedūras, tačiau dėl gerokai trumpesnio mokymo ir vykdymo laiko jie galėtų būti naudojami kaip pirminės patikros modeliai, prieš taikant tikslesnius KNT klasifikatorius. Svarbu paminėti, kad šiame darbe klasifikatoriai nebuvo optimizuojami siekiant maksimalaus aptikimo tikslumo – atlikti eksperimentai buvo skirti įvertinti, ar įterptas turinys apskritai gali būti aptinkamas pagal parametrų bitų struktūrą.

Gauti rezultatai patvirtina, kad neuroninių tinklų parametrai gali būti naudojami kaip efektyvus ir talpus paslėptos informacijos perdavimo kanalas. Parodyta, kad net paprastesni mašininio mokymosi modeliai gali aptikti įterptą turinį, todėl tolimesniuose tyrimuose tikslinga nagrinėti sudėtingesnes įterpimo strategijas ir mažesnio intensyvumo modifikacijas, ir kitas klasifikatorių architektūras. Taip pat svarbu įvertinti aptikimo metodų veikimą klasifikuojant kitų paskirčių modelius, skirtingas architektūras bei kvantuotus parametrų formatus. Šie rezultatai pabrėžia poreikį kurti modelių patikros mechanizmus jų tiekimo grandinėse ir įvertinti realias siūlomų metodų taikymo galimybes, pavyzdžiui, taikant automatizuotą mašininio mokymosi modeliais pagrįstą parametrų analizę prieš modelių platinimą ar naudojimą.

Literatūra

[1] Provos, N., & Honeyman, P. (2003). Hide and seek: An introduction to steganography. *IEEE security & privacy*, *1*(3), 32–44. doi: 10.1109/MSECP.2003.1203220

[2] Eid, W. M., Alotaibi, S. S., Alqahtani, H. M., & Saleh, S. Q. (2022). Digital image steganalysis: current methodologies and future challenges. IEEE Access, 10, 92321–92336. doi: 10.1109/ACCESS.2022.3202905

[3] Liu, T., Liu, Z., Liu, Q., Wen, W., Xu, W., & Li, M. (2020, December). Stegonet: Turn deep neural network into a stegomalware. In Proceedings of the 36th Annual Computer Security Applications Conference (pp. 928–938). doi: 10.1145/3427228.3427268

[4] Wang, Z., Liu, C., & Cui, X. (2021, September). Evilmodel: hiding malware inside of neural network models. In 2021 IEEE symposium on computers and communications (ISCC) (pp. 1–7). IEEE. doi: 10.1109/ISCC53001.2021.9631425

[5] Wang, Z., Liu, C., Cui, X., Yin, J., & Wang, X. (2022). Evilmodel 2.0: bringing neural network models into malware attacks. Computers & Security, 120, 102807. doi: 10.1016/j.cose.2022.102807

[6] Fridrich, J., & Goljan, M. (2002). Practical steganalysis of digital images: state of the art. Security and Watermarking of Multimedia Contents IV, 4675, 1–13. doi: 10.1117/12.465263