

Electronics and electrical engineering Elektronika ir elektros inžinerija

KLASIFIKATORIAUS VIETA DAIKTŲ INTERNETO KRAŠTŲ KOMPIUTERIOJE

Julius SKIRELIS*

Vilnius Gedimino technikos universitetas, Vilnius, Lietuva

Gauta 2018 m. birželio 13 d.; priimta 2018 m. birželio 20 d.

Santrauka. Tradicinė daiktų interneto debesų kompiuterija yra palengva keičiama kraštų kompiuterijos technologija. Kraštų kompiuterijos santvarka sprendžia brangių duomenų, perpildytų duomenų centrų ir jų efektyvumo problemas. Šiame straipsnyje apžvelgiamos ir palyginamos debesų ir kraštų kompiuterijos esminės savybės, atskleidžiami jų tarpusavio ryšiai struktūriniu aspektu. Apžvelgus kraštų kompiuterijoje taikomas technologijas, techninės įrangos, metodų ir programinių priemonių kontekste išaiškėjo poreikis integruoti klasifikatorių. Siekiant pabrėžti klasifikatoriaus kraštų kompiuterijoje privalumus, iširtos jų taikymo sritys, įvardyti esami sprendimai ir juose taikomi klasifikatoriai. Išsiaiškinus kraštų kompiuterijoje taikomus klasifikavimo metodus ir populiariausius klasifikatorių tipus nustatyta, kad kraštų kompiuterijoje nepakankamai išnagrinėtas saviorganizuojančių klasifikatorių taikymas, egzistuoja poreikis atlikti papildomus mokslinius tyrimus. Galiausiai apžvelgti galimi įgyvendinimo būdai remiantis esamais sprendimais, suskirsčius būdus į tris kategorijas – programinį, aparatinį ir mišrų.

Reikšminiai žodžiai: kraštų kompiuterija, daiktų internetas, saviorganizuojantis klasifikatorius.

Įvadas

Daiktų interneto technologijos tampa vis aktualesnės kasdienėje veikloje. Efektyvus paskirstytų duomenų apdorojimas tarp daiktų interneto įrenginių būtinas tiek mokslinių tyrimų, tiek pramoninio taikymo aspektu (Zalieskaitė ir Žilinskis, 2015). Paskirstytojo dirbtinio intelekto, duomenų bazių ir debesų kompiuterijos poreikis motyvuoja spręsti didelių paskirstytųjų duomenų perdavimo ir apdorojimo efektyvumo, taip pat ir saugomo problemas.

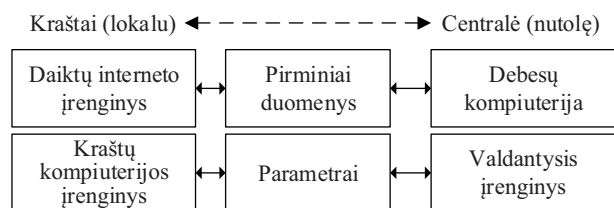
Centralizuota debesų kompiuterijos technologija iš dalies atitinka didelių duomenų apdorojimo kriterijų, tačiau natūrali daiktų internete vyraujančių duomenų dinamiskumo savybė užgožiama dėl neefektyvaus ir brangaus duomenų perdavimo modelio (Kubota, Fukuta, Yoshihide ir Kenichi, 2016). Šiai problemai spręsti taikomas decentralizuotas kraštų kompiuterijos (angl. *edge computing*) modelis, pasižymintis skaičiavimo išteklių paskirstymo, pigaus ir saugaus duomenų perdavimo savybėmis (Georgakopoulou, Jayaraman, Fazia, Villari ir Ranjan, 2016).

Didėjant interneto ryšiu pasiekiamų įrenginių skaičiui, 2020 m. prognozuojama per 50 milijardų įrenginių, tampa aktuali savalaikiškumo sąvoka (Weisong ir Dustdar, 2016), kai, pritaikius kraštų kompiuterijos modelį bent daliai šių

įrenginių, galima jų funkcijų sąrašą papildyti paskirstytųjų duomenų skaičiavimo (Sabella, Vaillant, Kuure, Rauschenbach ir Giust, 2016) ir perdavimo metodais. Kadangi didžioji dalis mobiliųjų įrenginių įprastinėmis funkcijomis neišnaudoja visų skaičiavimo išteklių, atveriamą erdvę įgyvendinti pigią duomenų infrastruktūrą, nedarant įtakos įrenginio specializuotoms funkcijoms.

Kraštų kompiuterijos sąvoka yra palyginti nauja daiktų interneto srityje, o jos įvedimą lėmė realiojo laiko tinklinių sistemų įgyvendinimo mobiliosiose sistemose poreikis (Tran, Hajisami, Pandey ir Pompili, 2017), pvz., išmaniuosiuose automobiliuose, išmaniuosiuose namuose (Taleb, Dutta, Ksentini, Iqbal ir Flinck, 2017) ar biomedicinoje (Wadood et al., 2017). Pastebėta, kad ši technologija taip pat sprendžia statinių ryšio tinklų problemas – nutolusių duomenų centrų funkcionalumas perkeliamas į lokalius kraštų kompiuterijos įrenginius. Modelis taip pat užtikrina didesnę duomenų saugumo lygį išlaikant tą pačią duomenų kainą (Vahid Dastjerdi ir Buyya, 2016), lyginant su debesų kompiuterijos sprendimais – tai grįžta supratimu, kad pirminiais duomenimis tarp įrenginių keičiamasi lokaliai, duomenys nėra perduodami geografiškai smarkiai nutolusiam centrui. Pagrindinis kriterijus, lemiantis mažą delną ir

*Autorius susirašinėti. El. paštas julius.skirelis@vgtu.lt



1 paveikslas. Debesų ir kraštų kompiuterijos modelio palyginimas

Figure 1. Comparison of Cloud and Edge computing models

1 lentelė. Debesų ir kraštų kompiuterijos savybių palyginimas
Table 1. Cloud and edge computing features comparison

Parametras	Debesų kompiuterija	Kraštų kompiuterija
Vėlinimas	Didelis	Mažas
Vėlinimo svyravimas	Didelis	Labai mažas
Mobilumas	Ribotas	Labai mobilus
Atakų tikimybė	Didelė	Labai maža
Vieta	Internetas	Lokalus tinklas
Nuotolis	Daug mazgų	Bent vienas
Geografinis skirstymas	Centralizuotas	Paskirstytas
Mazgų skaičius	Keletas	Labai daug
Santykinė kaina	Labai didelė	Maža

geresnę paslaugos kokybę (angl. *Quality of Service*) (Alra-wais, Althothaily, Hu ir Cheng, 2017; Intharawijitr, Member, Iida ir Member, 2017), yra kraštų kompiuterijos įrenginio valdymas būsenų parametrais. Fizinė tinklo struktūra iš esmės nekinta (žr. 1 paveikslą), tačiau, pritaikius kraštų kompiuterijos modelį, tinklo efektyvumas padidėja, taigi techninės sąlygos atitinka abiejų modelių pritaikymą.

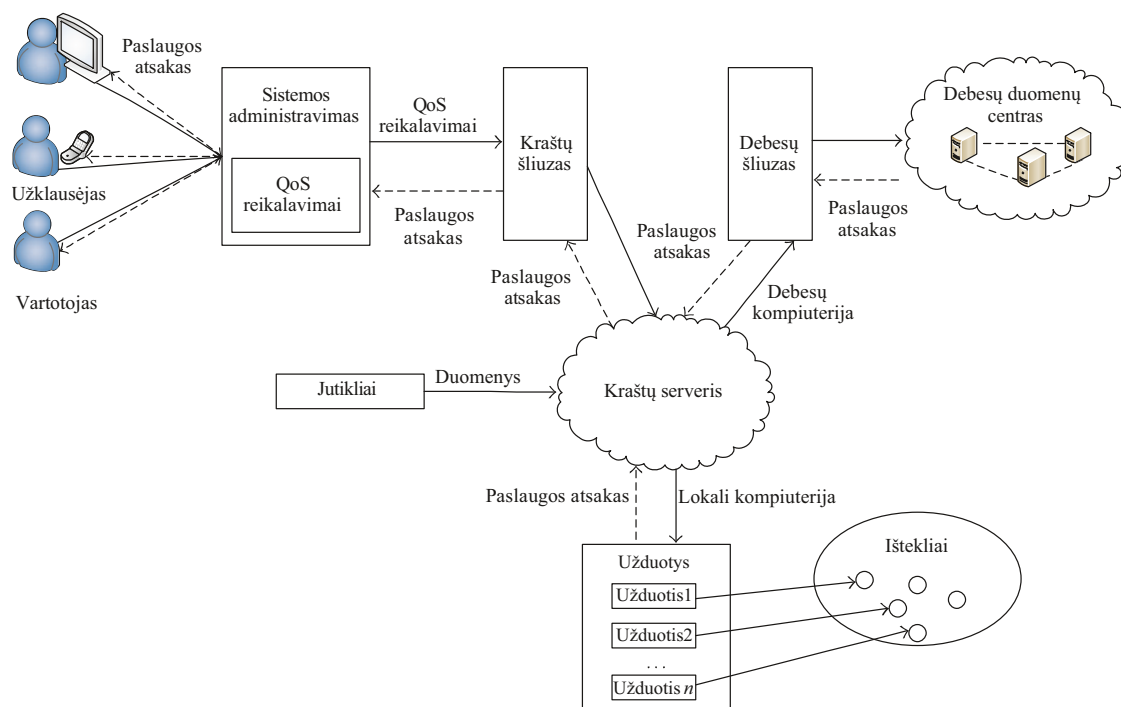
Siekiant optimaliai įgyvendinti kraštų kompiuteriją, būtini intelektiniai įrenginių valdymo sprendimai, vienas jų – dirbtinių neuronų tinklais grįsti klasifikatoriai.

Iš 1 lentelėje pateikto debesų ir kraštų kompiuterijų savybių palyginimo matyti, kad kraštų kompiuterijos technologija geriau atitinka šiuolaikinius poreikius aktualiais techniniais ir kokybiniais aspektais.

Šiuo straipsniu siekiama atskleisti klasifikatoriaus vietą daiktų interneto kraštų kompiuterijoje, todėl yra apžvelgiamos taikomos technologijos, jų privalumai ir trūkumai, išaiškinamos klasifikatoriaus taikymo sritys ir klasifikavimo metodai. Pateikiami galimi klasifikatorių įgyvendinimo kraštų kompiuterijoje būdai.

1. Kraštų kompiuterijoje taikomos technologijos

Iš esmės kraštų kompiuterija yra debesų kompiuterijos optimizacijos modelis (Barcelo et al., 2016), kai duomenys apdorojami kraštiniuose įrenginiuose. 2 paveikslas iliustruoja kraštų kompiuterijos struktūrą, jos sąveika su debesų kompiuterija, taip pat pabrėžia duomenų lokalumo savybę. Tokia pertvarka įgyvendinama techninės ir programinės įrangos ir jose taikomų metodų pokyčiais.



2 paveikslas. Daiktų interneto kraštų kompiuterijos struktūra (Li, Ota ir Dong, 2018)
Figure 2. IoT Edge computing structure (Li, Ota ir Dong, 2018)

Techninė įranga. Fizinės įrangos struktūra dažniausiai yra pavaldima iš tradicinės debesų kompiuterijos struktūros, jeigu technologija įgyvendinta belaidžiam tinkle, klasikinės belaidės priegigos taškas (WAN) programiškai optimizuojamas į programinio įsiterpimo belaidį priegigos tašką (SD-WAN) (Satya, 2016). Aparatinis kraštų kompiuterijos įgyvendinimo būdas remiasi koncentratoriumi – tarpiniu integriniu grandyniu, lokaliai atliekančiu tarpinių skaičiavimų užduotis. Pagrindinis kraštų kompiuterijos diegimo privalumas – duomenų apdorojimo įrenginiai tampa nepriklausomi nuo geografinės padėties (Skirelis ir Navauskas, 2017), o tai lemia mažesnius siunčiamų į centrą duomenų srautus, kartu greitesnį bendrą sistemos atsaką.

Vienas populiariausių kraštų kompiuterijos įgyvendinimo atvejų siejamas su mobiliojo ryšio pastočių funkcionalumu, anglų kalba vadinamu *Mobile-Edge Computing* (MEC). Pagrindinė MEC užduotis techninės įrangos lygmeniu – dinamiškai (Mao, Zhang, Song ir Letaief, 2017; El-Sayed et al., 2017) paskirstyti apkrovą tarp įrenginių (Mao et al., 2017), užtikrinant paslaugos kokybę visiems nutolusiems įrenginiams.

Metodai. Kraštų kompiuterija apibūdina tarpinius skaičiavimus, duomenų saugojimą ir papildomas tinklo paslaugas. Kadangi duomenų šaltiniai ir juos apdorojantys įrenginiai yra lokaliūs hierarchiniu požiūriu, dažnai pasitaiko paslaugos kokybės vertinimo metodas, remiantis Markovo grandies prognozavimu (Al-Sayed, Khatib ir Omara, 2016) – taikant paskirstytųjų įvykių tikimybių modelį ir lyginant jį su realiomis sistemos būsenomis. Klaidų matricių metodas taikomas paslaugos kokybei vertinti tiesiogiai (Din, Paul, Ahmad, Gupta ir Rho, 2018).

Kraštų kompiuterijos modelis puikiai tinka gilaus mokymo konvoliucinių neuronų tinklams įgyvendinti kraštų serveriuose (Din et al., 2018; Li et al., 2018). Esminė problema išlieka sinchronizavimo tarp įrenginių būdas, jai spręsti dažniausiai taikomi grafų metodai (Chen, Shi, Yang ir Xu, 2018). Kraštų kompiuterijos tinklo topologijos dažniausiai aprašomos grafais, populiariausias modelis – *Waxman*. Populiariausi grafais tinklo topologiją aprašantys standartai yra šie:

- *Boston Representative Internet Topology Generator* (BRITe) – iki šiol labiausiai paplitęs standartas, turintis grafinės aplinkos įrankį, įvertina grafų galios taisykles;
- *IGen* – iš esmės skirtas maršrutatoriaus lygio topologijoms aprašyti;
- *Inet* – grafų galios taisyklėmis grįstas standartas, įvertinantis tarpinių mazgų skaičių, kelio ilgį, klasterizavimo koeficientą ir kitas daiktų interneto metrikas.

Bendruoju atveju kraštų kompiuterijoje duomenų perdavimo laikas T aprašomas taip:

$$T = \frac{S}{t_m} \times \frac{1}{1 + (n-1)g},$$

čia S – duomenų dydis; t_m – teorinis modelio duomenų pralaidumas; n – mazgų skaičius; g – empiriškai nustatoma įrenginių lygiagretumo stiprinimo reikšmė. Ši išraiška rodo, kad, modelyje didinant kraštų kompiuterijos įrengi-

nių skaičių, ne tik suteikiama galimybė įgyvendinti mobiliuosius duomenų centrus, perkeltiant centralizuoto centro galimybes, tačiau leidžia sumažinti duomenų perdavimo vėlinimą, o kartu ir jų kainą.

Programinės priemonės. Dauguma programinių įrenginių siejama su techninės įrangos gaminiu, kai programinė terpė yra skirta tikrinei užduočiai techninėje įrangoje atlikti. Techninę įrangą su jai skirtu programiniu paketu siūlo tokios korporacijos, kaip *Intel*, *DELL*, *Cisco* ir kitos. Egzistuoja kompiuteriuose naudojamos programinės įrangos paketai: *OpenFlow*, *OpenStack*, *Docker*, *KVM* branduoliu grįstos virtualios mašinos. Juos naudojant siekiama susisteminti virtualiosios realybės, viešosios erdvės, jutiklių, industrijos 4.0 duomenų srautus, diegti kraštų kompiuterijos technologijas lokaliai su duomenų šaltiniu, siekiant išvengti neesminių duomenų vyravimo interneto tinkle.

Programiniam imitavimui naudojami *iFogSim*, *CloudSim*, *CloudAnalyst*, *CloudExp*, *EdgeCloudSim* programiniai paketai. Šios imitacinės programinės priemonės pasižymi galimybe imituoti realių duomenų srautą tinklo modelyje, įvertinti delsą, kainą, taip pat imituoti įvairius tinklo trukdžius, tokius kaip ryšio kanalo pralaidos sumažėjimas, laikini ryšio trūkiai. Šios priemonės įgyvendintos naudojant *Java* programavimo aplinką, pasiekiami pilni išeities kodai. Verta paminėti, kad daugeliui imitacinių įrenginių *CloudSim* išeities kodas naudojamas kaip pamatinis, leidžiantis išplėsti bazinio imitatoriaus savybes tiksliniams poreikiams. Didžiausias minėtų programinių paketų trūkumas – visa veiksmų virtualizacija, imituojama aplinka yra ribojama parametrais, neįtraukiamos nenumatytos sistemos būsenos ir tinklų anomalijos.

Antroji imitavimo programinių priemonių kategorija – įrankiai, įgyvendinti *MATLAB* aplinkoje, kurių populiariausi yra *SimEvents*, *NS-2*, *Complex Networks Package*. Pastarieji suderinami tik su senesnėmis *MATLAB* versijomis (≤ 7.0), todėl 2017 m. buvo sukurta nauja platforma daiktų interneto įrenginiams imituoti ir tirti – *ThingSpeak* papildinys. Ši aplinka sudaryta iš *MATLAB* integruojamos nutolusios duomenų bazės. Privalumai yra tokie: nutolusi aplinka leidžia vizualizuoti ir tirti duomenis autonomiškai; galima įkelti duomenis į duomenų bazę tiesiogiai iš jutiklių arba per *MATLAB* aplinką; duomenys perduodami šifruota *REST API* jungtimi; gali būti naudojama ne tik imitavimo, bet ir realioms sistemos įgyvendinti. Trūkumai šie: nemokama tik namų vartotojams (su apribojimais) ir akademinėi bendruomenei, komercinei veiklai reikia pirkti licenciją; duomenys saugomi *ThingSpeak* duomenų bazėje, o tai gali kelti saugumo kriterijui abejonių; atviruoju kodu grįstas tik *MATLAB* papildinio išeities kodas.

Buvo nagrinėtos atvirojo kodo imitacinės programinės priemonės, leidžiančios įsiterpti į programos vykdymą, todėl yra galimybė papildyti kraštų kompiuterijos simuliacinio modelį klasifikatoriumi ir atlikti tyrimus.

Taip pat nagrinėta uždarojo kodo populiarūs daiktų interneto platforma – *Microsoft Azure*. Ši infrastruktūra grįsta virtualiojo šliuzo struktūra, kurio pagrindinis privalumas – suderinamumas tiek su *Windows*, tiek su *Linux*

operacinėmis sistemomis. *Linux* operacinės sistemos palaikymas lemia platformos populiarumą, nes suteikia galimybę nesudėtingai perkelti klasikines debesų kompiuterijos struktūras link kraštų. Virtualus šliuzas atlieka komunikaciją su *Microsoft* serveriais, siekiant sinchronizuoti būsenas ir prireikus naudoti nutolusias duomenų bazes duomenims saugoti. Platformos naujumą atspindi analogiškų sistemų nenaudojamas modelis – virtualioji talpykla (angl. *Cloudlet*) (Ai, Peng ir Zhang, 2017) yra konfigūruojama ir atnaujinama tiesiogiai iš valdymo skydo. Didžiausias privalumas – daugybės programavimo kalbų palaikymas, virtuali talpykla gali būti programuojama *JavaScript*, *Java*, *.NET* programavimo paketais. Ši gausa leidžia pritaikyti itin populiarią *NodeJS* aplinką šliuze, taip pat leidžia programuoti įvairias kraštų kompiuterijos užduotis *Visual Studio* įrankiu. Daug dėmesio skiriama saugumui, todėl visi duomenų kanalai, išskyrus jutiklių, yra šifruojami *X.509* raktais. Virtualiosios talpyklos yra apmokestinamos, ir tai yra didžiausias jų trūkumas.

Atlikus kraštų kompiuterijoje taikomų technologijų apžvalgą, išsikelti tokie tikslai:

- nustatyti svarbiausias aparatines ir programines klasifikatoriaus taikymo kraštų kompiuterijoje sritis, reikalaujančias papildomų mokslinių tyrimų;
- susisteminti ir apibūdinti kraštų kompiuterijoje taikomus klasifikavimo metodus.

2. Klasifikatoriaus taikymo kraštų kompiuterijoje sritys

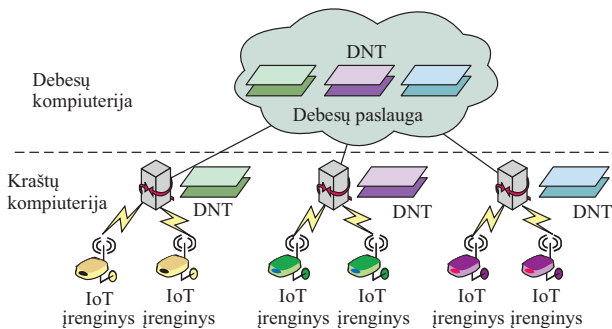
Kraštų kompiuterijoje klasifikatorius taikomas įvairiais aspektais, keli iš jų – anomalijų paketuose aptikimas (Pasca, Dama, Sathya ir Kuchi, 2017), brukalų (angl. *spam*) detektavimas (Zhang et al., 2017). Aptikus anomalijas dirbtinių neuronų tinklu grįstas klasifikatorius integruojamas į duomenų paketų maršrutizavimo funkciją (Hodo et al., 2016). Tokia klasifikatoriaus išraiška įgyvendinama taikant moko daugiasluoksnio perceptrono arba saviorganizuojančio žemėlapiu (angl. *SOM*) klasifikatorių. Tai užtikrina daugiau nei 99 % kenkėjiškų atakų – paketų – aptikimą.

Klasifikatorius taip pat taikomas dideliems duomenims struktūruoti (Diallo et al., 2017), klasifikatoriaus parametrai parenkami pagal duomenų tipą. Atlikus visą arba dalinį duomenų struktūravimą, sumažinamas duomenų perdavimo vėlinimas.

Klasifikatorius taikytinas autorių teisių saugos srityje (Alrawais et al., 2017) ir pateikiamas programiniu lygmeniu, čia paskirstyti kraštų kompiuterijos įrenginiai atlieka autorinės programinės įrangos serijinio numerio autentiškumo patikrą.

Pasiūlytas kraštų kompiuterijos įrenginių pertvarkymas į jungtinių dirbtinio neuronų tinklo klasifikatorių (Li et al., 2018), kai kiekvienas įrenginys atlieka vieno neurono funkciją (Sheng, Wang ir Zhao, 2016), o interneto tinku perduodami sinchronizavimo ir neuronų tinklo parametrai.

Viena pagrindinių kraštų kompiuterijos taikymo sričių – protingų miestų ir namų infrastruktūra (Danner, Wills, Ruiz ir Lerner, 2016). Čia orientuojamasi į vaizdų apdorojimą, todėl susiduriama su didelių duomenų



3 paveikslas. Kraštų kompiuterijos modelis gilaus mokymo tinklui (Li et al., 2018)

Figure 3. Edge computing model for Deep Neural Network (Li et al., 2018)

efektyvaus realaus atsako problemomis. Sudėtingi gilaus mokymo dirbtiniai neuronų tinklai, tokie kaip *AlexNet*, *GoogLeNet*, *VGG* ir kiti, reikalauja daug skaičiavimo išteklių. Šių tinklų lygiagretnimas kraštų kompiuterijos modelyje (3 paveikslas) sudaro galimybes atlikti kompleksinę analizę bei priimti sprendimus realiuoju laiku.

3. Kraštų kompiuterijoje taikomi klasifikavimo metodai

Klasifikuojami duomenys yra įvairios prigimties ir įvairių dimensijų, atitinkamai parenkamas klasifikatoriaus tipas ir jo parametrai (Zaliejkaitė ir Žilinskas, 2015). Egzistuoja keli požiūriai, kur galima taikyti klasifikavimą kraštų kompiuterijoje: klasifikuojami duomenys, siekiant pakeisti jų dimensiją (Holden et al., 2006) ir sugrupuoti; klasifikatorius naudojamas kaip tarpinis įrenginys, koncentruojantis duomenis, atliekantis pradinį jų apdorojimą; visa daiktų interneto kraštų kompiuterijos infrastruktūra atitinka klasifikatorių, o jos įrenginiai – klasifikatoriaus elementus.

Tradiciškai dideli jutiklių duomenys apdorojami duomenų centruose – klasteriuose ar debesų kompiuterijoje. Duomenų sumažinimas iki jų patekimo į duomenų centrus yra pagrindinė kraštų kompiuterijos ir joje taikomų klasifikatorių užduotis. Egzistuoja daugybė glaudinimo metodų, taikomų duomenims, siunčiamiems į klasterius glaudinti, tačiau tai neišsprendžia problemos, kad nutolusiuose duomenų centruose duomenys dubliuojami siekiant užtikrinti jų pasiekiamumą. Šiai problemai spręsti taikomi išankstiniai jutiklių duomenų apdorojimo metodai, tokie kaip semantinė analizė, klasterizavimas ir klasifikavimas.

Pagal iškeltą užduotį taikomi mokomieji ir saviorganizuojantys klasifikatoriai. Dažniausiai naudojami mokomieji klasifikatoriai: neraiškioji logika; tiesiniai; atraminių vektorių (angl. *SVM*); *k* artimiausių kaimynų (angl. *kNN*). Iš saviorganizuojančių klasifikatorių dažniausiai naudojami saviorganizuojantys žemėlapiai (Hammadi-Mesmoudi ir Korczak, 1995).

Atliekant literatūros apžvalgą nustatyta, kad saviorganizuojančių klasifikatorių taikymas kraštų kompiuterijoje nepakankamai išnagrinėtas, todėl reikalauja detalesnių mokslinių tyrimų.

4. Kraštų kompiuterijoje tinkamų klasifikatorių įgyvendinimo būdai

Technologijos lankstumas leidžia įterpti klasifikatorių įvairiais būdais, iš kurių optimalus parenkamas pagal klasifikavimo uždavinį ir klasifikuojamus duomenis.

Programinis. Paprastas būdas išnaudoti esamą fizinę daiktų interneto infrastruktūrą – įgyvendinti klasifikatorių programiniu būdu (Farris, Militano, Nitti, Atzori ir Iera, 2016). Toks įgyvendinimo būdas pakeičia interneto tinklo sklindančių duomenų rūšį, taip pat pakeičia modelį ir jo įrenginių funkcines reikšmes. Priklausomai nuo klasifikatoriaus paskirties klasifikatorius taikomas duomenims arba tinklo pertvarkai. Siekiant tenkinti lokalumo kriterijus ir esant sąlygai, kad galinis įrenginys yra išmanusis, programinis klasifikatorius dažniausiai įgyvendinamas pačiame įrenginyje (Alippi, Fantacci, Marabissi ir Roveri, 2016).

Aparatinis. Įgyvendinimas grindžiamas naudojant aparatinę įrangą kaip duomenų tarp daikto ir centralės skyriklių, realizuojama naudojant koncentruojamąjį serverį ar kitokią įrangą, kuri apdoroja klasifikavimo pobūdžio duomenis, rezultatas pateikiamas kitiems kraštų kompiuterijos įrenginiams arba centriniam serveriui. Aparatiniams klasifikatoriui priskiriamos kraštų kompiuterijos sistemos skirtos balso garsui signalu pagerinti naudojant aparatinius skaitmeninio signalo apdorojimo integruotus grandynus kartu su komunikacinio tinklo aparatiniiais sprendimais.

Mišrus. NEC integrinių grandynų gamintojas siūlo kraštų kompiuterijos galimybes realizuoti lauku programuojamos loginės matricos (LPLM) lustais (Ke, Zhang, Song ir Wan, 2018), kurių antrinė dalis sudaryta iš ARM mikrovaldiklio. Pateikiamas pavyzdys, atskleidžiantis vaizdų aptikimą klasifikavimo metodu, taip pat minimas saugumo lygio užtikrinimą įgyvendinus duomenų apdorojimą aparatinių lygmeniu (Jridi, Chapel, Dorez, Le Bougeant ir Le Botlan, 2018).

Klasifikatoriaus įgyvendinimas LPLM iš esmės sujungia programinio ir aparatinio įgyvendinimo aspektus, išskiriant LPLM galimybę vykdyti aparatines instrukcijas lygiagrečiai – atlikti klasifikavimą realiuoju laiku. Klasifikatorius integruojamas lusto pavidalu lygiagrečiai su esama sistema arba, jeigu sistema jau įgyvendinta LPLM terpėje, ją papildyti kraštų kompiuterijos klasifikatoriaus funkcionalumu.

Išvados

1. Kraštų kompiuterijoje klasifikavimo uždavinys yra svarbus, nes, atlikus duomenų klasifikavimą, interneto tinklas išnaudojamas efektyviai, tai lemia galimybę įgyvendinti realiojo laiko daiktų interneto įrenginių infrastruktūrą.
2. Saviorganizuojantys klasifikatoriai kraštų kompiuterijoje nėra pakankamai ištirti.

Literatūra

- Ai, Y., Peng, M., & Zhang, K. (2017). Edge cloud computing technologies for internet of things: a primer. *Digital Communications and Networks*, 4(2), 77-86. <https://doi.org/10.1016/j.dcan.2017.07.001>
- Al-Sayed, M. M., Khattab, S., & Omara, F. A. (2016). Prediction mechanisms for monitoring state of cloud resources using Markov chain model. *Journal of Parallel and Distributed Computing*, 96, 163-171. <https://doi.org/10.1016/j.jpdc.2016.04.012>
- Alippi, C., Fantacci, R., Marabissi, D., & Roveri M. (2016). A Cloud to the ground: the new frontier of intelligent and autonomous networks of things, *IEEE Communications Magazine*, 54(12), 140-20. <https://doi.org/10.1109/MCOM.2016.1600541CM>
- Alrawais, A., Alhothaily, A., Hu, C., & Cheng X. (2017). Fog computing for the internet of things: security and privacy issues. *IEEE Internet Computing*, 21(2), 34-42. <https://doi.org/10.1109/MIC.2017.37>
- Barcelo, M., Correa, A., Llorca, J., Tulino, A. M., Vicario, J. L., & Morell, A. (2016). IoT-Cloud service optimization in next generation smart environments. *IEEE Journal on Selected Areas in Communications*, 34(12), 4077-4090. <https://doi.org/10.1109/JSAC.2016.2621398>
- Chen, X., Shi, Q., Yang, L., & Xu, J. (2018). ThriftyEdge: resource-efficient edge computing for intelligent IoT applications. *IEEE Network*, 32(1). <https://doi.org/10.1109/MNET.2018.1700145>
- Danner, J., Wills, L., Ruiz, E. M., & Lerner, L. W. (2016). Rapid precedent-aware pedestrian and car classification on constrained IoT platforms. *Proceedings of the 14th ACM/IEEE Symposium on Embedded Systems for Real-Time Multimedia – ESTIMedia'16*. Pittsburgh, Pennsylvania, JAV. <https://doi.org/10.1145/2993452.2993562>
- Diallo, L., Hassan, A., Hashim, A., Eyiomika, M. J., Babiker, S., & Elagib, O. (2017). The rise of internet of things and Big Data on the Cloud: challenges and future trends. *International Journal of Future Generation Communication and Networking*, 10(3), 49-56. <https://doi.org/10.14257/ijfgcn.2017.10.3.06>
- Din, S., Paul, A., Ahmad, A., Gupta, B., & Rho, S. (2018). Service orchestration of optimizing continuous features in industrial surveillance using Big Data based fog-enabled internet of things. *IEEE Access*, 6. <https://doi.org/10.1109/ACCESS.2018.2800758>
- El-Sayed, H., Sankar, S., Prasad, M., Puthal, D., Gupta, A., Mohanty, M., & Lin, C. T. (2017). Edge of things: the big picture on the integration of edge, IoT and the Cloud in a distributed computing environment, *IEEE Access*, 6.
- Farris, I., Militano, L., Nitti, M., Atzori, L., & Iera, A. (2016). Federated edge-assisted mobile clouds for service provisioning in heterogeneous IoT environments. *IEEE World Forum on Internet of Things, WF-IoT 2015 – Proceedings* (pp. 591-596). <https://doi.org/10.1109/WF-IoT.2015.7389120>
- Georgakopoulos, D., Jayaraman, P. P., Fazia, M., Villari, M., & Ranjan, R. (2016). Internet of Things and Edge Cloud Computing Roadmap for Manufacturing. *IEEE Cloud Computing*, 3(4), 66-73. <https://doi.org/10.1109/MCC.2016.91>
- Hammadi-Mesmoudi, F., & Korczak, J. J. (1995). An unsupervised neural network classifier and its application in remote sensing, *Proceedings of the International Conference on Image Processing*, (410), 4-6. <https://doi.org/10.1049/cp:19950656>
- Hodo, E., Bellekens, X., Hamilton, A., Dubouilh, P.-L., Iorkyase, E., Tachtatzis, C., & Atkinson, R. (2016). Threat analysis of IoT networks using artificial neural network intrusion detection system. *2016 International Symposium on Networks*,

- Computers and Communications (ISNCC)* (pp. 1-6). Hammamet, Tunis. <https://doi.org/10.1109/ISNCC.2016.7746067>
- Holden, A. J., et al. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(July), 504-507.
- Intharawijitr, K., Member, S., Iida, K., & Member S. (2017). Simulation study of low latency network architecture using mobile edge computing. *IEICE TRANSACTIONS on Information and Systems*, (5), 963-972. <https://doi.org/10.1587/transinf.2016NTP0003>
- Jridi, M., Chapel, T., Dorez, V., Le Bougeant, G., & Le Botlan, A. (2018). SoC-based edge computing gateway in the context of the internet of multimedia things: experimental platform. *Journal of Low Power Electronics and Applications*, 8(1). <https://doi.org/10.3390/jlpea8010001>
- Ke, Q., Zhang, J., Song, H., & Wan, Y. (2018). Big data analytics enabled by feature extraction based on partial independence. *Neurocomputing*, 288, 3-10. <https://doi.org/10.1016/j.neucom.2017.07.072>
- Kubota, M., Fukuta, S., Yoshihide, N., & Kenichi, A. (2016). Dynamic resource controller technology to accelerate processing and utilization of IoT data. *FUJITSU Scientific & Technical Journal*, 52(4), 41-51.
- Li, G., Song, J., Wu, J., & Wang, J. (2018). Method of resource estimation based on QoS in edge computing. *Wireless Communications and Mobile Computing*, 2018, Article ID 73 089 13. <https://doi.org/10.1155/2018/7308913>
- Li, H., Ota, K., & Dong, M. (2018). Learning IoT in edge: deep learning for the internet of things with edge computing. *IEEE Network*, 32(1). <https://doi.org/10.1109/MNET.2018.1700202>
- Mao, Y., Zhang, J., Song, S. H., & Letaief, K. B. (2017). Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems. *IEEE Transactions on Wireless Communications*, 16(9). <https://doi.org/10.1109/TWC.2017.2717986>
- Pasca, T. V., Dama, S., Sathya, V., & Kuchi, K. (2017). A feasible cellular internet of things. *IEEE Consumer Electronics Magazine*, 6(1), 66-72. <https://doi.org/10.1109/MCE.2016.2614421>
- Sabella, D., Vaillant, A., Kuure, P., Rauschenbach, U., & Giust, F. (2016). Mobile-edge computing architecture: the role of MEC in the internet of things. *IEEE Consumer Electronics Magazine*, 5(4), 84-91. <https://doi.org/10.1109/MCE.2016.2590118>
- Satya, M. S. (2016). Edge computing: vision and challenges. *IEEE Internet of Things Journal*, 3(5), 637-646. <https://doi.org/10.1109/JIOT.2016.2579198>
- Sheng, Y., Wang, J., & Zhao, Z. (2016). A communication-efficient model of sparse neural network for distributed intelligence. *Proceedings – IEEE INFOCOM 2016, Septe(2)*, 515-520.
- Skirelis, J., & Navakauskas, D. (2017). Edge computing in IoT: preliminary results on modeling and performance analysis. *2017 5th IEEE Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)* (pp. 1-4). Ryga, Latvija. <https://doi.org/10.1109/AIEEE.2017.8270555>
- Taleb, T., Dutta, S., Ksentini, A., Iqbal, M., & Flinck, H. (2017). Mobile edge computing potential in making cities smarter. *IEEE Communications Magazine*, 55(3), 38-43. <https://doi.org/10.1109/MCOM.2017.1600249CM>
- Tran, T. X., Hajisami, A., Pandey, P., & Pompili, D. (2017). Collaborative mobile edge computing in 5G networks: new paradigms, scenarios, and challenges. *IEEE Communications Magazine*, 55(April), 54-61. <https://doi.org/10.1109/MCOM.2017.1600863>
- Vahid Dastjerdi, A., & Buyya, R. (2016). Fog computing: helping the internet of things realize. *IEEE Computer Society*, 49(8), 112-116. <https://doi.org/10.1109/MC.2016.245>
- Wadood, A., Ali, Z., Ghouzali, S., ALfawaz, B., Muhammad, G., Hossain, M. S. 2017. Biometric Security Through Visual Encryption for Fog Edge Computing. *IEEE Access*, 5, 5531-5538.
- Weisong, S., & Dustdar, S. (2016). The promise of edge computing. *IEEE Computer Society*, 59(5), 78-81.
- Zalieckaitė, L. ir Žilinskas, R. (2015). Daiktų interneto technologijos taikymo versle nauda ir rizika. *Informacijos Mokslai*, 72, 102-117. <https://doi.org/10.15388/Im.2015.72.9223>
- Zhang, J., Li, Q., Wang, X., Feng, B., & Guo, D. (2017). Towards fast and lightweight spam account detection in mobile social networks through fog computing. *Peer-to-Peer Networking and Applications*, 11(4), 778-792. <https://doi.org/10.1007/s12083-017-0559-3>

CLASSIFIER PLACE IN EDGE COMPUTING FOR INTERNET OF THINGS

J. Skirelis

Abstract

Internet of Things Cloud Computing is more and more substituted with Edge Computing. Such substitution solves problems of costly data, crowdedness and effectiveness of datacenters. This paper reviews and compares essential features of Cloud and Edge Computing technologies, revealing their structural relationship. Review of technologies applied in Edge computing in terms of technical equipment, methods and software used, revealed demand of classifier incorporation. To highlight classifiers advantages in Edge Computing, application fields were investigated, therefore currently existing solutions, with classifiers used were named. After determination of classification methods and most popular classifiers employed in Edge Computing it is observed that self-organized classifiers are insufficiently analyzed and requires additional research. Finally, based on existing solutions three categories – software, hardware and mixed type of possible classifier implementations in Edge Computing are presented.

Keywords: edge computing, internet of things, self-organized classifier.