2020 ‹ LETNIK XXVIII ‹ ŠTEVILKA 3 ‹ JUL./AVG./SEP. ‹ ISSN 1318-1882

03 UPORABNA INFORMATIKA

U P O R A B N A
I N F O R M A T I K A
2020 ŠTEVILKA 3 JUL/AVG/SEP LETNIK XXVIII ISSN 1318-1882

Znanstveni prispevki

Strokovni prispevki

Kratki znanstveni prispevki

Informacije

U P O R A B N A
I N F O R M A T I K A

2020 ŠTEVILKA 2 APR/MAJ/JUN LETNIK XXVIII ISSN 1318-1882

Ustanovitelj in izdajatelj
Slovensko društvo INFORMATIKA
Litostrojska cesta 54, 1000 Ljubljana

Predstavnik
Niko Schlamberger

Odgovorni urednik
Saša Divjak

Tehnični urednik
Slavko Žitnik

Lektoriranje angleških izvlečkov
Marvelingua (angl.)

Oblikovanje
KOFEIN DIZAJN, d. o. o.

Prelom in tisk
Boex DTP, d. o. o., Ljubljana

Naklada
200 izvodov

Naslov uredništva
Slovensko društvo INFORMATIKA
Uredništvo revije Uporabna informatika
Litostrojska cesta 54, 1000 Ljubljana
www.uporabna-informatika.si

Revija izhaja četrtletno. Cena posamezne številke je 20,00 EUR. Letna naročnina za podjetja 85,00 EUR, za vsak nadaljnji izvod 60,00 EUR, za posameznike 35,00 EUR, za študente in seniorje 15,00 EUR. V ceno je vključen DDV.

Revija Uporabna informatika je od številke 4/VII vključena v mednarodno bazo INSPEC.

Revija Uporabna informatika je pod zaporedno številko 666 vpisana v razvid medijev, ki ga vodi Ministrstvo za kulturo RS.

Revija Uporabna informatika je vključena v Digitalno knjižnico Slovenije (dLib.si).

Vabilo avtorjem

V reviji Uporabna informatika objavljamo kakovostne izvirne članke domačih in tujih avtorjev z najširšega področja informatike v poslovanju podjetij, javni upravi in zasebnem življenju na znanstveni, strokovni in informativni ravni; še posebno spodbujamo objavo interdisciplinarnih člankov. Zato vabimo avtorje, da prispevke, ki ustrezajo omenjenim usmeritvam, pošljejo uredništvu revije po elektronski pošti na naslov ui@drustvo-informatika.si.

Avtorje prosimo, da pri pripravi prispevka upoštevajo navodila, objavljena v nadaljevanju ter na naslovu http://www.uporabna-informatika.si.

Za kakovost prispevkov skrbi mednarodni uredniški odbor. Članki so anonimno recenzirani, o objavi pa na podlagi recenzij samostojno odloča uredniški odbor. Recenzenti lahko zahtevajo, da avtorji besedilo spremenijo v skladu s priporočili in da popravljeni članek ponovno prejmejo v pregled. Uredništvo pa lahko še pred recenzijo zavrne objavo prispevka, če njegova vsebina ne ustreza vsebinski usmeritvi revije ali če članek ne ustreza kriterijem za objavo v reviji.

Pred objavo članka mora avtor podpisati izjavo o avtorstvu, s katero potrjuje originalnost članka in dovoljuje prenos materialnih avtorskih pravic. Nenaročenih prispevkov ne vračamo in ne honoriramo. Avtorji prejmejo enoletno naročnino na revijo Uporabna informatika, ki vključuje avtorski izvod revije in še nadaljnje tri zaporedne številke.

S svojim prispevkom v reviji Uporabna informatika boste prispevali k širjenju znanja na področju informatike. Želimo si čim več prispevkov z raznoliko in zanimivo tematiko in se jih že vnaprej veselimo.

Uredništvo revije

Navodila avtorjem člankov

Članke objavljamo praviloma v slovenščini, članke tujih avtorjev pa v angleščini. Besedilo naj bo jezikovno skrbno pripravljeno. Priporočamo zmernost pri uporabi tujk in – kjer je mogoče – njihovo zamenjavo s slovenskimi izrazi. V pomoč pri iskanju slovenskih ustreznic priporočamo uporabo spletnega terminološkega slovarja Slovenskega društva Informatika Islovar (www.islovar.org).

Znanstveni članek naj obsega največ 40.000 znakov, strokovni članki do 30.000 znakov, obvestila in poročila pa do 8.000 znakov.

Članek naj bo praviloma predložen v urejevalniku besedil Word (*.doc ali *.docx) v enojnem razmaku, brez posebnih znakov ali poudarjenih črk. Za ločilom na koncu stavka napravite samo en prazen prostor, pri odstavkih ne uporabljajte zamika.

Naslovu članka naj sledi za vsakega avtorja polno ime, ustanova, v kateri je zaposlen, naslov in elektronski naslov. Sledi naj povzetek v slovenščini v obsegu 8 do 10 vrstic in seznam od 5 do 8 ključnih besed, ki najbolje opredeljujejo vsebinski okvir članka. Pred povzetkom v angleščini naj bo še angleški prevod naslova, prav tako pa naj bodo dodane ključne besede v angleščini. Obratno velja v primeru predložitve članka v angleščini.

Razdelki naj bodo naslovljeni in oštevilčeni z arabskimi številkami.

Slike in tabele vključite v besedilo. Opremite jih z naslovom in oštevilčite z arabskimi številkami. Vsako sliko in tabelo razložite tudi v besedilu članka. Če v članku uporabljate slike ali tabele drugih avtorjev, navedite vir pod sliko oz. tabelo. Revijo tiskamo v črno-beli tehniki, zato barvne slike ali fotografije kot original niso primerne. Slik zaslonov ne objavljamo, razen če so nujno potrebne za razumevanje besedila. Slike, grafikoni, organizacijske sheme ipd. naj imajo belo podlago. Enačbe oštevilčite v oklepajih desno od enačbe.

V besedilu se sklicujte na navedeno literaturo skladno s pravili sistema APA navajanja bibliografskih referenc, najpogosteje torej v obliki (Novak & Kovač, 2008, str. 235). Na koncu članka navedite samo v članku uporabljeno literaturo in vire v enotnem seznamu po abecednem redu avtorjev, prav tako v skladu s pravili APA. Več o sistemu APA, katerega uporabo omogoča tudi urejevalnik besedil Word 2007, najdete na strani http://owl.english.purdue.edu/owl/resource/560/01/.

Članku dodajte kratek življenjepis vsakega avtorja v obsegu do 8 vrstic, v katerem poudarite predvsem strokovne dosežke.

Primerjava hitrosti simetričnih bločnih šifer

Marko Kompara[1], Tomi Jerenko[1], Marko Hölbl[1]
[1] Univeza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, Maribor, Slovenija
marko.kompara@um.si, jerenkotomi@gmail.com, marko.holbl@um.si

Izvleček

Zaupnost podatkov je ena od osnovnih zahtev varovanja podatkov, ki je danes najpogosteje zagotovljena z uporabo simetričnih bločnih šifer. Najpogosteje uporabljena simetrična bločna šifra je AES. Njegova prilagodljivost z različnimi načini delovanja, možnostjo hitre programske in strojne implementacije, ter podpora s strani proizvajalcev opreme omogočajo, da je AES prisoten skoraj povsod, kjer je potrebno šifriranje podatkov. V prispevku je hitrost delovanja AES primerjana s šiframi Camellia, DES, 3DES, Serpent in Twofish. Poleg tega je v analizo vključena tudi strojna implementacija algoritma AES. Zanima nas, kakšen je dejanski učinek strojne implementacije algoritma na hitrost njegovega delovanja. Rezultati raziskave kažejo, da je AES najhitrejši algoritem med primerjanimi algoritmi. Razlika v hitrosti pa se v strojno pospešenem načinu izboljša za več kot cel red velikosti.

Ključne besede: AES, AES-NI, Camellia, DES, 3DES, Serpent, Twofish, primerjava hitrosti delovanja, simetrične bločne šifre.

Abstract

Confidentiality is one of the underlying data protection requirements. In modern cryptography, this is generally provided for by symmetric block ciphers. The most frequently used such cipher is AES. The cipher's modularity achieved with the help of the encryption modes, fast software and hardware implementation, and good support from many equipment manufacturers and software developers have allowed AES to be present wherever encryption is needed. In the paper, the performance of the AES algorithm is compared to Camellia, DES, 3DES, Serpent and Twofish. The hardware implementation of the AES is also included in the comparison. We are interested to see how much hardware implementation increases the algorithm's performance. Results show that AES is the fastest algorithm of those compared. Furthermore, the hardware-implemented version is more than an order of magnitude faster.

Keywords: AES, AES-NI, Camellia, DES, 3DES, Serpent, Twofish, speed comparison, symmetric block ciphers.

1 UVOD

Zagotavljanje varnosti in zasebnosti v digitalnem svetu je postalo zelo pomembno. Zgodovinsko je bila zaupnost podatkov zanimiva predvsem za državne organizacije. S širjenjem digitalizacije je ta postala pomembna za podjetja in kar nekaj časa je minilo, preden je zagotavljanje zaupnosti podatkov postalo pomembno tudi za navadnega uporabnika. Zasebnost uporabnikov na spletu je problematika, ki se je začela sistematično naslavljati še kasneje in je v veliki meri tudi še vedno aktualna. Šifrirni algoritmi so glavni način zagotavljanja zaupnosti in so pomembni tudi pri zagotavljanju zasebnosti. Poznamo več vrst šifrirnih algoritmov. Simetrični bločni algoritmi so najpogosteje uporabljena oblika šifrirnih algoritmov in med temi je šifra AES (Advanced Encryption Standard) najbolj razširjena. Natančnih vrednosti o deležih uporabe ni mogoče pridobiti, ampak konser-

vativna ocena je, da se AES uporablja v več kot 50% šifriranja vseh podatkov [1].

Algoritem AES je nastal leta 2001, kot rezultat NIST (National Institute of Standards and Technology) natečaja [2], v katerem so izbirali novo simetrično bločno šifro, ki bi postala standardizirana oblika šifriranja in tako nadomestila obstoječi algoritem DES (Data Encryption Standard) oziroma 3DES (Triple DES). AES je torej več kot dvajset let star algoritem. V tem prispevku želimo preveriti hitrost tega algoritma proti nekaterim drugim dobro poznanim algoritmom. V dvajsetih letih se je v veliki meri izboljšala tudi procesorska moč naprav, zato nas zanima, v kakšni meri strojna pospešitev še vpliva na hitrost delovanja algoritma AES.

V raziskavo smo vključili popularne knjižnice Crypto++, Libgcrypt, Nettle in OpenSSL. Izkazalo se je, da različne knjižnice razen osnovnega nabora tipično ne implementirajo velikega števila šifrirnih algoritmov. Zato smo se tudi omejili na šifrirne algoritme, ki so bili zastopani v vseh knjižnicah: AES, AES-NI, Camellia, DES, 3DES, Serpent in Twofish. Edina izjema v tem naboru je OpenSSL, ki ne implementira Serpent in Twofish, vendar ker je to zelo pogosto uporabljena knjižnica, smo se določili, da jo kljub temu vključimo. Čeprav je osnovni namen uporabe več različnih knjižnic preprečiti vpliv na rezultate hitrosti, ki bi lahko nastali kot posledica različnih kakovosti implementacij šifer, bodo rezultati te raziskave prikazali tudi razlike med delovanjem knjižnic in izpostavili, katere knjižnice vključujejo najbolj učinkovito implementacijo posamezne šifre.

V nadaljevanju tega prispevka bodo najprej na kratko predstavljeni vsi šifrirni algoritmi, ki so primerjani v tem delu. Zatem bomo kratko poglavje namenili predstavitvi okoliščin eksperimenta, katere omejitve smo pri raziskavi upoštevali in kako so meritve izvedene. V sledečem poglavju bomo predstavili in analizirali rezultate primerjave delovanja šifer in različnih knjižnic. V zadnjem poglavju bomo povzeli rezultate in zaključili prispevek.

2 SIMETRIČNE BLOČNE ŠIFRE

Simetrično bločno šifriranje [3] je postopek, v katerem se šifriranje in dešifriranje izvede na podlagi enakega ključa. Takšne šifre lahko istočasno obdelujejo samo podatke določene velikosti, ki jo imenujemo blok. Večje količine podatkov se razdelijo v več blokov, od katerih je vsak posamezno obdelan. Simetrične bločne šifre so najpogostejši način šifriranja podatkov (npr. na disku ali med prenosom), medtem ko se druge oblike šifriranja (asimetrične šifre in tokovne simetrične šifre) uporabljajo za bolj specializirane naloge. V nadaljevanju tega poglavja bomo na kratko predstavili simetrične bločne šifre, katerih hitrosti bomo merili v tem prispevku. Tabela 1 vključuje podatke o šifrah, ki smo jih uporabili v meritvah in knjižnicah, ki jih implementirajo.

2.1 Šifra AES

Standard AES (Advanced Encryption Standard) [4] je svojo pot začel kot algoritem Rijndael, ki je delo dveh belgijskih kriptografov (Vincent Rijmen in Joan Daemen) [5]. Rijndael je bil eden od 15 kandidatov in končni zmagovalec NIST (National Institute of Standards and Technology) natečaja za novo standardizirano simetrično bločno šifro, ki je potekal med 1997 in 2000. Rijndael je bil leta 2001 standardiziran kot AES v nekoliko omejenem delovanju, tako da deluje

Tabela 1: **Lastnosti izbranih šifer in njihova implementacija v knjižnicah.**

	Lastnost šifre		**Knjižnica**			
Šifre	**Blok (biti)**	**Testiran ključ (biti)**	**Crzpto++**	**Libgcrzpt**	**Nettle**	**OpenSSL**
AES	128	128	✓	✓	✓	✓
AES-NI	128	128	✓	✓	✓	✓
Camellia	128	128	✓	✓	✓	✓
DES	64	56	✓	✓	✓	✓
3DES	64	168	✓	✓	✓	✓
Serpent	128	128	✓	✓	✓	✓
Twofish	128	128	✓	✓	✓	✓

s 128 bitnimi bloki in s 128, 192 in 256 bitov dolgimi ključi. AES deluje po principu substitucijsko-permutacijskega omrežja (angl. substitution–permutation network).

2.2 Implementacija AES-NI

AES-NI [6, 7] je strojno pospešena implementacija algoritma AES, ki dobi svoje ime po Intel AES novih ukazih (angl. Intel Advanced Encryption Standard New Instructions). To je bila prva množična implementacija takšnega delovanja šifre. Danes s tem imenom poimenujemo tudi stojno pospešene implementacije drugih proizvajalcev (npr. na AMD in ARM procesorjih). AES-NI vsebuje šest ukazov, ki omogočajo strojno izvajanje celotnega algoritma AES. Štirje ukazi so namenjeni šifriranju in dešifriranju, dva ukaza pa sta namenjena generiranju vmesnih ključev algoritma. Ti ukazi so fizično vgrajeni v centralno procesno enoto, zaradi česar je njihovo izvajanje tudi toliko hitrejše. Ti ukazi so neposredno dostopni iz uporabniškega prostora, kar omogoča tudi enostavnejšo implementacijo AES šifre. Neposredno izvajanje šifriranja in dešifriranja na procesorju tudi izniči potrebo po hranjenju iskalnih tabel v predpomnilniku in zagotavlja časovno zakasnitev, ki ni odvisna od obdelovanih podatkov. Takšno delovanje preprečuje nekatere najbolj razširjene in najbolj nevarne napade po stranskem kanalu (angl. side channel attack).

2.3 Šifra Camellia

Camellia je najnovejša med primerjanimi šiframi. Razvili so jo v Mitsubishi Electric in Nippon Telegraph and Telephone leta 2000 na Japonskem [8]. Camellia deluje po načinu Feistelove mreže in uporablja 128 bitne bloke z možnostjo 128, 192 in 256 bitnih ključev. Te postavke so bile namenoma izbrane, da je šifra primerljiva s šiframi, ki so sodelovale na natečaju AES. Za šifro so pokazali, da je primerljiva v hitrosti in nivoju varnosti s kandidati natečaja, njeno prednost pa predstavlja kompaktna strojna implementacija [9]. Deloma se Camellia lahko izvede tudi z ukazi AES-NI (substitucijski del v S-škatlah), kar omogoča njeno pospešeno delovanje na obstoječi strojni opremi [8]. V tem prispevku so merjene zgolj programske implementacije te šifre (ne izkorišča se pospešitev, ki bi jo prinesla uporaba AES-NI). Camellia je vključena tudi v ISO/IEC standard in je bila potrjena za uporabo v projektu Evropske Unije NESSIE (New European Schemes for Signatures, Integrity and Encryption) in japonskem projektu CRYPTOREC (Cryptography Research and Evaluation Committees).

2.4 Šifri DES in 3DES

DES (Data Encryption Standard) je standardiziran algoritem DEA (Data Encryption Algorithm), ki ga je razvil IBM [10]. Algoritem je bil leta 1977 standardiziran s strani NBS (National Bureau of Standards), ki je s časom postal današnji NIST. To je vodilo k hitremu sprejemu šifre v praksi, vendar so se zelo hitro pojavili tudi pomisleki glede njene varnosti. Največja pomanjkljivost je njena dolžina ključa, ki je znašala samo 56 bitov. DES je bil dokončno opuščen kot standard leta 2005 in ni več primeren za varovanje podatkov, ker ne zadošča modernim minimalnim standardom dolžine ključa [11]. DES je zasnovan na Feistelovi mreži s 64 bitnim ključem, pri čemer se 8 bitov uporablja za zaznavanje napak, in 64 bitnim blokom.

TDES (Triple Data Encryption Standard) oz. 3DES je standard algoritma TDEA, ki je bil ustvarjen leta 1995 in je v svojem delovanju enak standardu DES, le da se izvede trikrat ob uporabi treh različnih ključev (možne so tudi kombinacije ponovne uporabe enakega ključa, vendar se ti načini ne smatrajo več kot varni) [11]. Posledično je ta sestavljen iz 168 naključnih bitov in 24 paritetnih bitov. Čeprav izgleda, kot da 3DES zagotavlja najvišji nivo varnosti s 168 bitnim ključe, je standard ranljiv na napad s srečanjem v sredini (angl. meet-in-the-middle attack), ki zniža varnost na 112 bitov [12]. Posledično je 3DES poleg DES najmanj varna šifra od primerjanega nabora algoritmov.

2.5 Šifra Serpent

Serpent izhaja iz leta 1998 in je druga šifra v našem naboru, ki je bila med kandidati NIST natečaja in glede na rezultate natečaja tudi druga najboljša [13]. Čeprav je Serpent veljal za bolj varen algoritem kot Rijndael, je bila njegova pomanjkljivost predvsem počasnejše delovanje v programskih implementacijah. Tako kot AES deluje s 128 bitnimi bloki in omogoča uporabo 128, 192 ali 256 bitnega ključa, čeprav sam algoritem vedno uporabi 256 bitno vrednost, ki je v primeru manjšega posredovanega ključa razširjena na to velikost. Kot je že bilo omenjeno smo se v tej raziskavi omejili na 128 bitne ključe (ki bodo kot del algoritma nato avtomatsko razširjeni na 256 bitov). Tako kot AES je tudi Serpent zgrajen iz substitucijsko-permutacijskih gradnikov.

2.6 Šifra Twofish

Twofish [14] je še ena šifra iz leta 1998 in je tretja šifra (poleg Rijndael in Serpent) v tem seznamu, ki je bila udeležena na natečaju, na katerem je bil izbran AES. Za svoje delovanje uporablja Feistelovo mrežo. Podatki se obdelujejo v 128 bitov velikih blokih in možna je uporaba ključev velikosti 128, 192 ali 256 bitov, čeprav algoritem omogoča tudi vnos drugih vrednosti, ki pa so nato dopolnjene z ničlami do prvega od teh mejnikov.

3 ŠIFRA AES

Standard AES (Advanced Encryption Standard) [4] je svojo pot začel kot algoritem Rijndael, ki je delo dveh belgijskih kriptografov (Vincent Rijmen in Joan Daemen) [5]. Rijndael je bil eden od 15 kandidatov in končni zmagovalec NIST (National Institute of Standards and Technology) natečaja za novo standardizirano simetrično bločno šifro, ki je potekal med 1997 in 2000. Rijndael je bil leta 2001 standardiziran kot AES v nekoliko omejenem delovanju, tako da deluje

Tabela 2: **Povprečne meritve hitrosti v ciklih/zlog, za vse kombinacije šifer, knjižnic, velikosti podatkov in operacijo.**

		16B		32B		512B		10MB	
		Šif	**Deš**	**Šif**	**Deš**	**Šif**	**Deš**	**Šif**	**Deš**
Crypto++	AES	17,051	19,856	14,326	19,047	11,488	18,556	11,358	18,512
	AES-NI	6,250	6	3,653	3,967	0,819	0,814	0,732	0,732
	Camellia	23,32	23,215	22,675	22,861	22,17	22,232	22,183	22,26
	DES	43,484	43,388	42,526	42,584	41,652	41,672	41,874	41,872
	3DES	117,094	117,143	116,539	116,56	115,059	115,091	115,275	115,274
	Serpent	37,788	33,413	37,019	33,238	36,228	32,704	36,302	32,782
	Twofish	19,6	19,468	18,699	18,899	18,31	18,207	18,265	18,215
Libgcrypt	AES	13,75	16,247	11,878	14,158	9,952	12,321	9,692	12,17
	AES-NI	8,553	8,371	4,75	4,684	1,672	1,773	1,534	1,577
	Camellia	26,16	25,423	22,795	22,71	19,99	20,088	19,885	19,881
	DES	45,386	44,93	42,513	42,361	40,067	40,104	40,188	40,177
	3DES	96,858	96,646	93,799	93,395	91,323	90,794	91,318	91,008
	Serpent	41,467	38,019	38,609	34,855	36,217	32,875	36,213	32,549
	Twofish	21,873	21,638	19,312	19,137	16,799	16,87	16,733	16,827
Nettle	AES	12	12,117	11,689	11,791	11,49	11,45	11,545	11,529
	AES-NI	4,554	4,5	2,875	2,779	1,352	1,288	1,286	1,238
	Camellia	19,988	19,986	19,851	19,837	19,503	19,475	19,608	19,611
	DES	41,856	42,197	40,803	41,201	39,983	40,58	40,339	40,823
	3DES	124,36	124,666	122,789	123,516	120,573	121,525	121,462	121,917
	Serpent	38,325	32,396	38,169	32,174	14,115	13,107	14,21	13,186
	Twofish	20,685	18,642	20,416	18,505	20,226	18,33	19,818	18,323
OpenSSL	AES	11,873	15,067	10,555	13,434	9,492	11,836	9,268	11,666
	AES-NI	6,681	7,642	3,456	3,937	0,753	0,799	0,677	0,677
	Camellia	21,027	21,964	20,384	20,766	18,622	18,855	18,689	18,9
	DES	47,983	47,263	46,493	45,326	45,132	43,516	45,207	43,827
	3DES	127,068	125,512	124,848	123,495	123,225	121,698	124,715	123,989
	Serpent	Knjižnica ne implementira šifre Serpent.							
	Twofish	Knjižnica ne implementira šifre Twofish							

s 128 bitnimi bloki in s 128, 192 in 256 bitov dolgimi ključi. AES deluje po principu substitucijsko-permutacijskega omrežja (angl. substitution–permutation network).

4 IMPLEMENTACIJA AES-NI

AES-NI [6, 7] je strojno pospešena implementacija algoritma AES, ki dobi svoje ime po Intel AES novih ukazih (angl. Intel Advanced Encryption Standard New Instructions). To je bila prva množična implementacija takšnega delovanja šifre. Danes s tem imenom poimenujemo tudi stojno pospešene implementacije drugih proizvajalcev (npr. na AMD in ARM procesorjih). AES-NI vsebuje šest ukazov, ki omogočajo strojno izvajanje celotnega algoritma AES. Štirje ukazi so namenjeni šifriranju in dešifriranju, dva ukaza pa sta namenjena generiranju vmesnih ključev algoritma. Ti ukazi so fizično vgrajeni v centralno procesno enoto, zaradi česar je njihovo izvajanje tudi toliko hitrejše. Ti ukazi so neposredno dostopni iz uporabniškega prostora, kar omogoča tudi enostavnejšo implementacijo AES šifre. Neposredno izvajanje šifriranja in dešifriranja na procesorju tudi izniči potrebo po hranjenju iskalnih tabel v predpomnilniku in zagotavlja časovno zakasnitev, ki ni odvisna od obdelovanih podatkov. Takšno delovanje preprečuje nekatere najbolj razširjene in najbolj nevarne napade po stranskem kanalu (angl. side channel attack).

4.1 Analiza rezultatov

Analizo hitrosti različnih simetričnih bločnih šifrirnih algoritmov smo začeli s statistično analizo. Podatki pridobljeni v eksperimentu so bili pretežno nenormalno porazdeljeni, zato smo se odločili za uporabo neparametričnih testov. Za vsako knjižnico smo posebej izvedli *Kruskal-Wallis test* [18], ki pokaže, ali med algoritmi (znotraj posamezne knjižnice) obstajajo signifikantne razlike. Rezultat je bil v vseh primerih pozitiven, zato smo nadaljevali z *Mann-Whitney U testi* [19], s katerimi med vsakim parom algoritmov preizkusimo signifikantnost razlik (v tem primeru hitrosti) med njima. Kljub uporabi *Bonferronijevega popravka* [20] so statistični testi za vse parne primerjave pokazali, da med algoritmi obstajajo signifikantne razlike. P-vrednosti so bile tudi po zaokroženju na več deset decimalnih mest še vedno 0. Zato smo se odločili, da teh rezultatov ne vključimo, temveč raje nadaljujemo z analizo velikosti vpliva (angl. effect size). Velikost vpliva je metrika, ki nam pove, kako velika oz. pomembna je razlika, ki so jo statistični testi pokazali. V tej nalogi smo uporabili *Cohenov d* za izračun velikosti učinka, ki je definiran v enačbi (1), kjer sta $\bar{A}$ in $\bar{B}$ povprečni vrednosti meritev algoritmov A in B, in σ je standardni odklon.

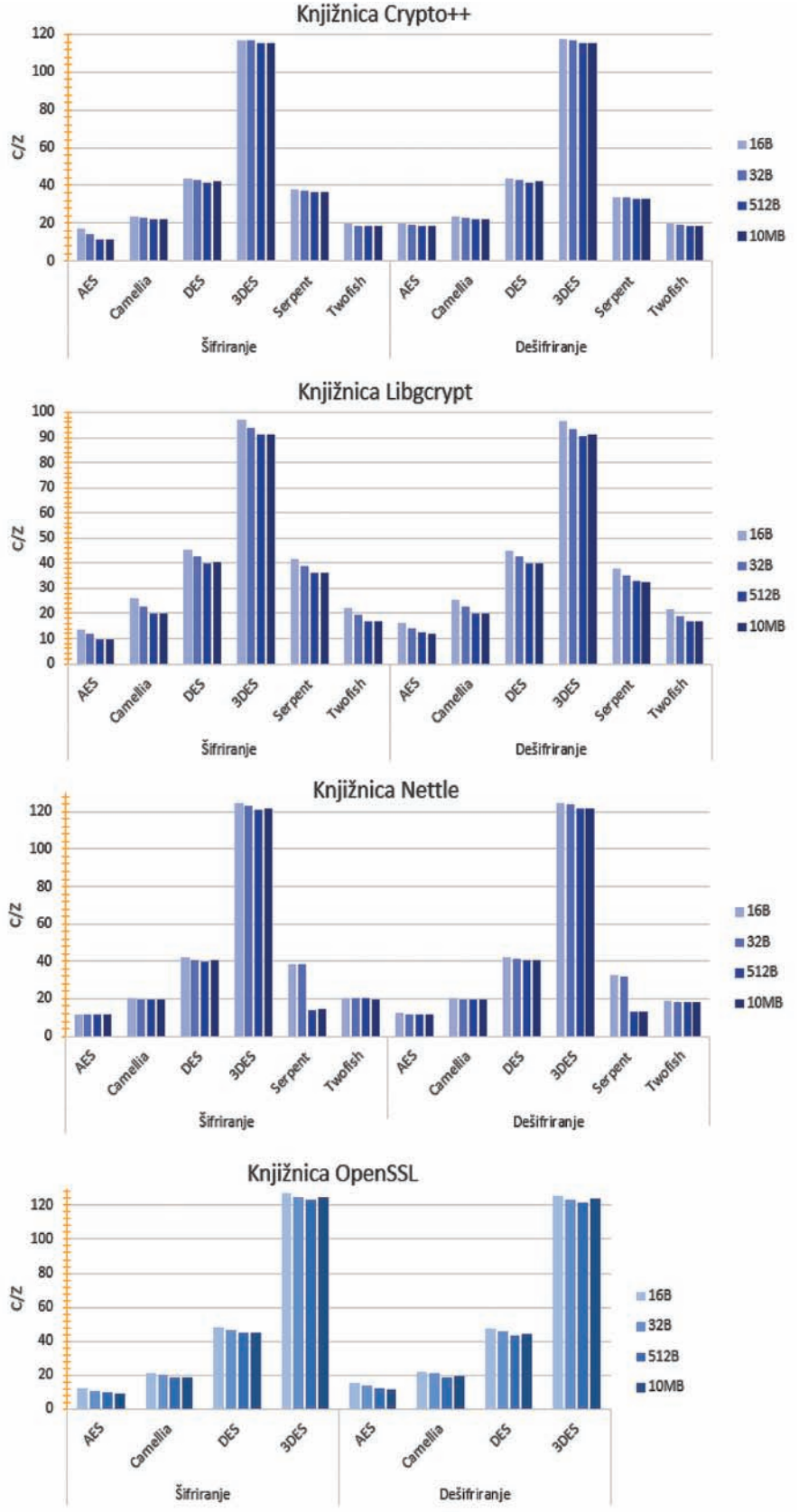
$$d = \frac{\bar{A} - \bar{B}}{\sqrt{\frac{\sigma_A^2 + \sigma_B^2}{2}}} \qquad (1)$$

Rezultati velikosti učinka so zaradi zelo majhnih standardnih odklonov računalniško generiranih podatkov zelo veliki v primerjavi z lestvico za interpretacijo rezultatov, ki jo je predlagal Cohen [21]. Ne glede na to so med velikostmi učinka za vsak par šifer (znotraj iste knjižnice) opazne razlike. Rezultati kažejo, da velikost učinka raste z večanje velikost vhodnih podatkov. Iz tega lahko razberemo, da z večanjem količine šifriranih podatkov, raste tudi pomembnost razlik med algoritmi. Rezultati nakazujejo, da so si med seboj po hitrosti delovanja (vsaj pri manjši količini podatkov) najbolj podobne šifre AES, Camellia in Twofish ter DES in Serpent. Iz rezultatov je tudi hitro razvidno, da se algoritma AES-NI in 3DES v hitrosti izvajanja najbolj razlikujeta od ostalih.

Rezultati statistične analize nakazujejo, da so med algoritmi pomembne razlike v hitrosti delovanja, vendar na podlagi teh testov nismo dobili praktične predstave, kako velike so te razlike. Zato bomo v nadaljevanju razlike v hitrosti delovanja različnih simetričnih bločnih šifer opisali in analizirali še s pomočjo opisne statistike.

4.2 Primerjava hitrosti delovanja simetričnih bločnih šifer

Na sliki 1 so prikazani grafikoni z rezultati hitrosti v ciklih na zlog (C/Z) delovanja programsko implementiranih šifer v različnih knjižnicah. Najbolj očiten rezultat je zagotovo počasno delovanje algoritma 3DES ne glede na uporabljeno knjižnico. Druga najpočasnejša šifra je DES. Ti dve sta tudi najmanj varni od vseh primerjanih šifer, zato je uporaba teh šifer v kakršenkoli drug namen, kot zaradi združljivosti za nazaj (angl. backward compatible) nerazumljiva. Za slabe rezultate lahko v majhni meri vpliva tudi izbira velikosti šifriranih in dešifriranih podatkov, ki so večkratniki 128 bitov, kar pomeni, da morata algoritma DES in 3DES procesirati večje število blokov, ker



Slika 1: **Grafikoni primerjave šifrirnih algoritmov po knjižnicah.**

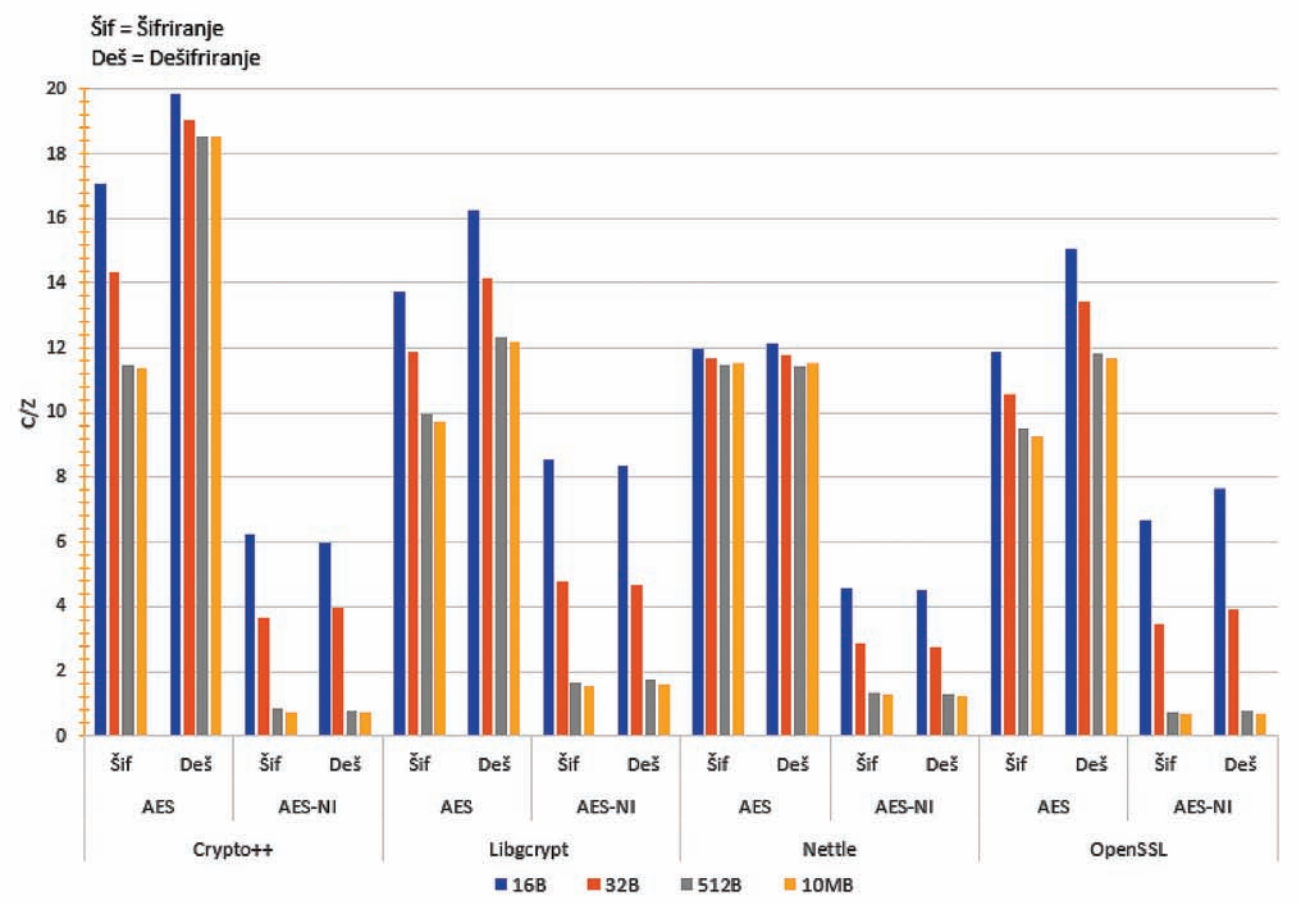
procesirata vhodne podatke v blokih velikosti 64 bitov. Tretja najpočasnejša šifra v večini knjižnic je Serpent. Glede na to, da smo že v opisu šifre omenili, da velja za bolj kompleksen algoritem, ki zagotavlja višji nivo varnosti, je takšen rezultat pričakovan. Camellia in Twofish sta bila primerljivo hitra z zelo majhnimi razlikami med vsemi knjižnicami. Ne glede na to je bila šifra Twofish v povprečju malenkostno hitrejša. Pomanjkljivost te šifre je predvsem to, da ni implementirana v knjižnici OpenSSL, ki je zelo pogosto uporabljena knjižnica [22].

Med vsemi programsko implementiranimi algoritmi je bil brez dvoma najhitrejši algoritem AES. Na podlagi tega lahko sklepamo, da je najpogosteje uporabljen šifrirni algoritem tudi najhitrejši. Med knjižnicami se je za izvajanje AES in Camellia najboljše odrezala knjižnica OpenSSL, Serpent je bil najhitrejši v knjižnici Nettle, medtem ko je za preostale algoritme bila Libgcrypt najboljša izbira. Predvsem v knjižnici Nettle, v določeni meri pa tudi pri Crypto++, je mogoče opaziti, da z večanjem velikosti šifriranih oz. dešifriranih podatkov hitrost ne narašča, kot bi pričakovali zaradi vedno manjšega deleža režijskih stroškov. Če torej pogosto šifriramo zelo majhne količine podatkov, bi ti dve knjižnici mogoče (odvisno od dane šifre) bili boljša izbira kot knjižnici OpenSSL in Libgcrypt.

4.3 Primerjava delovanja AES in AES-NI

V prejšnjem poglavju smo pokazali, da je AES najhitrejša programsko implementirana šifra v naboru algoritmov, ki smo jih primerjali. Velik delež modernih procesorskih enot dodatno podpira tudi strojno pospešitev tega algoritma. Na sliki 2 je prikazana primerjava hitrosti delovanja šifre AES v programski in strojno pospešeni implementaciji (AES-NI) v različnih knjižnicah.

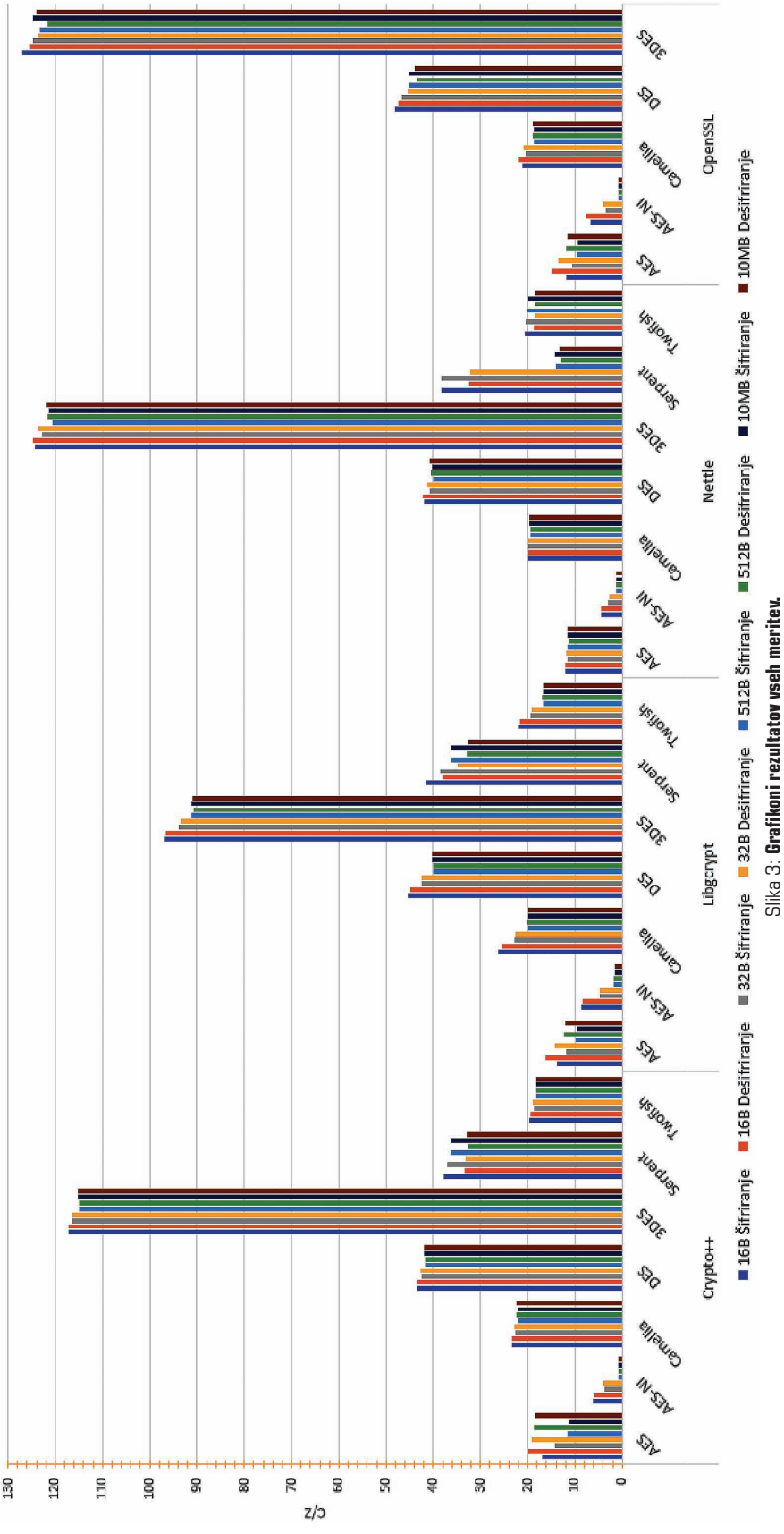
Razlike v hitrosti delovanja med AES in AES-NI so takoj očitne, ne glede na uporabljeno knjižnico. AES-NI je vedno signifikantno hitrejši. Razlike v primeru manjših velikosti podatkov so nekoliko manjše zaradi režijskega dela, ki je potreben ne glede na



Slika 2: **Grafikoni primerjave AES in AES-NI hitrosti delovanja.**

tip implementacije. Za podatke velikosti 16B je AES-NI med 1,6 (Libgcrypt) in 2,7 (Crypto++) krat hitrejši pri šifriranju in med 1,9 (Libgcrypt) ter 3,3 (Crypto++) krat hitrejši pri dešifriranju podatkov. Z večanjem količine podatkov hitrost izvajanja proporcionalno na količino podatkov narašča, predvsem v AES-NI. Tu vidimo, da je signifikantna razlika med meritvami do velikosti podatkov 512B. Med 512B in 10MB je izboljšanje komaj opazno, kar pomeni, da smo dosegli zgornji prag hitrosti izvajanja. Za podatke velikosti 10MB je AES-NI med 6,3 (Libgcrypt) in 15,5 (Crypto++) krat hitrejši pri šifriranju in med 7,7 (Libgcrypt) ter 25,3 (Crypto++) krat hitrejši pri dešifriranju podatkov. Opazimo tudi, da je med programsko in strojno pospešeno implementacijo vedno do najmanjšega izboljšanja prišlo v knjižnici Libgcrypt in do največjega v Crypto++. Absolutno najhitrejše delovanje za majhne količine podatkov pa dosegla knjižnica Nettle (4,5 C/Z za 16B podatkov pri šifriranju in dešifriranju), medtem, ko je najhitrejše, za vse razen najmanjših podatkov, delovala implementacija knjižnice OpenSSL (0,7 C/Z za 10MB podatkov pri šifriranju in dešifriranju). Knjižnica OpenSSL se je v splošnem najbolje odrezala tudi pri izvajanju programske implementacije šifre AES.

Iz slike 2 je mogoče razbrati še eno lastnost programske implementacije AES. V vseh knjižnicah je v programskih implementacijah dešifriranje počasnejše od šifriranja. Ta lastnost je značilna za AES in v neki meri tudi za Serpent, ker imata podobno struk-



Slika 3: **Grafikoni rezultatov vseh meritev.**

turo delovanja (substitucijsko-permutacijsko omrežje). Kot bomo videli v nadaljevanju to ni značilno za druge primerjane šifre. Te razlike nastanejo zaradi zelo majhnih razlik med operacijami, ki se uporabljajo za šifriranje in dešifriranje in so opazne samo v programski implementaciji. Kot lahko vidimo AES-NI izvaja šifriranje in dešifriranje skoraj brez razlike v hitrosti med obema operacijama. To je tudi razlog, da so izboljšave med hitrostjo delovanja programske implementacije in strojne implementacije toliko večje pri dešifriranju.

4.4 Razprava

Na sliki 3 so povzeti vsi rezultati te raziskave. Iz grafikona lahko razberemo razlike med šifriranjem in dešifriranjem, ki smo jih omenili na koncu prejšnjega poglavja. Programsko implementiran AES in Serpent imata zobčaste vrhove stolpičnih grafikonov. Ta oblika nakazuje razlike med hitrostjo šifriranja in dešifriranja s temi šiframi, medtem ko so sosedni stolpiči pri ostalih šifrah, ki so osnovane na Feistelovi mreži, na isti višini. To je posledica delovanja Feistelove mreže, v kateri sta operaciji šifriranja in dešifriranja identični in posledično tudi enako hitri pri enaki velikosti vhodnih podatkov.

Med vsemi šiframi je AES-NI ne glede na izbrano knjižnico vedno najhitrejša izbira algoritma. Prikaz hitrosti takšnega delovanja v primerjavi z ostalimi šiframi (Slika 3), pokaže, kako velika je dejansko ta razlika. Algoritmi AES, AES-NI in Camellia se najhitreje izvajajo s knjižnico OpenSSL. Šifra Serpent je najhitrejša v knjižnici Nettle, medtem ko knjižnica Libgcrypt v primerjavi z drugimi knjižnicami, najhitreje izvaja šifre DES, 3DES in Twofish. Čeprav je knjižnica Crypto++ ena najbolj znanih knjižnic, namenjenih kriptografiji, izvajanje s to knjižnico ni za noben algoritem najhitrejše, vendar pa ne glede na okoliščine zagotavlja solidno hitrost delovanja vseh primerjanih šifer.

5 ZAKLJUČEK

V prispevku smo primerjali hitrosti šifriranja in dešifriranja različnih šifrirnih algoritmov v več boljše poznanih in sprejetih knjižnicah. Za daleč najhitrejše se je izkazalo strojno pospešeno izvajanje algoritma AES, ki je signifikantno hitrejše kot programska implementacija - v povprečju je AES-NI pri 16B podatkov 2,5 krat in pri 10MB podatkov 14,9 krat hitrejši kot osnoven AES. Tudi med izključno programskimi šiframi se je AES izkazal za najhitrejšega med vsemi primerjanimi algoritmi. Za drugi in tretji najhitrejši algoritem sta se izkazala Twofish in Camellia z zelo primerljivimi rezultati. Tem so sledili še Serpent, DES in 3DES. 3DES je bil občutno najpočasnejši algoritem v naboru primerjanih šifer. Rezultati so pokazali, da se v praksi dejansko uporablja najbolj učinkovit algoritem in čeprav je mogoče intuitivno mišljenje, da so starejši algoritmi, ki niso več varni bolj preprosti za izvajanje, rezultati jasno pokažejo, da temu ni tako in so dejansko najslabši v naboru primerjanih algoritmov.

Zaključimo lahko torej, da je AES še vedno najhitrejši algoritem in da strojna implementacija v veliki meri izboljša učinkovitost tega algoritma. Raziskava tudi nakazuje, da je za najhitrejšo možno implementacijo tega algoritma potrebno uporabiti knjižnico OpenSSL.

Afiliacije

Ta raziskava je nastala ob podpori raziskovalnega programa št. P2-0057, katerega je sofinancirala Javna agencija za raziskovalno dejavnost Republike Slovenije iz državnega proračuna ter projekta Cyber Security Network of Competence Centres for Europe (CyberSec4Europe), katerega je financirala EU iz okvirnega programa EU za raziskave in inovacije – Obzorje H2020.

LITERATURA

[1] C. Paar, Introduction to Cryptography by Christoff Paar: Lecture 8: Advanced Encryption Standard (AES), (2014). https://www.youtube.com/watch?v=NHuibtoL_qk.

[2] Crypto competitions - AES: the Advanced Encryption Standard, (n.d.). https://competitions.cr.yp.to/aes.html (dostopano 21 april 2020).

[3] D. Altermatt, Symmetric Encryption – An Introduction, (2019). https://www.scip.ch/en/?labs.20190815 (dostopano 17 junij 2020).

[4] M.J. Dworkin, E.B. Barker, J. R. Nechvatal, J. Foti, L. E. Bassham, E. Roback, J. F. Dray Jr., Advanced Encryption Standard (AES), (2001). doi:10.6028/NIST.FIPS.197.

[5] J. Daemen, V. Rijmen, The Block Cipher Rijndael, in: Int. Conf. Smart Card Res. Adv. Appl., Springer Verlag, 2000: pp. 277–284. doi:10.1007/10721064_26.

[6] J. Rott, Intel® Advanced Encryption Standard Instructions (AES-NI), (2012). https://software.intel.com/en-us/articles/intel-advanced-encryption-standard-instructions-aes-ni (dostopano 6 februar 2020).

[7] S. Gueron, Intel® Advanced Encryption Standard (Intel® AES) Instructions Set - Rev 3.01, 2012. https://software.intel.com/en-us/articles/intel-advanced-encryption-standard-aes-instructions-set (dostopano 21 april 2020).

[8] J. Kivilinna, Block Ciphers: Fast Implementations on x86-64 Architecture, 2013.

[9] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima, T. Tokita, Camellia: A 128-Bit block cipher suitable for multiple platforms – Design and analysis, in: Lect. Notes Comput. Sci., Springer Verlag, 2001: pp. 39–56. doi:10.1007/3-540-44983-3_4.

[10] W. Stallings, Cryptography and Network Security, 4th ed., Prentice Hall, 2005. http://www.inf.ufsc.br/~bosco.sobral/ensino/ine5680/material-cripto-seg/2014-1/Stallings/Stallings_Cryptography_and_Network_Security.pdf.

[11] E. Barker, Recommendation for Key Management Part 1: General, NIST Spec. Publ. 800-57 Part 1 Revis. 4. (2016). https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf.

[12] W. Diffie, M.E. Hellman, Exhaustive Cryptanalysis of the NBS Data Encryption Standard, Computer (Long. Beach. Calif). 10 (1977) 74–84. doi:10.1109/C-M.1977.217750.

[13] J. Sugier, Implementing AES and Serpent Ciphers in New Generation of Low-Cost FPGA Devices, in: Complex Syst. Dependability, 2013: pp. 273–287.

[14] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson, Twofish: A 128Bit Block Cipher, 1998. https://www.researchgate.net/publication/245272403_Twofish_A_128Bit_Block_Cipher (dostopano 5 februar 2020).

[15] Intel® CoreTM i5-4200M Processor Product Specifications, (2013). https://ark.intel.com/content/www/us/en/ark/products/76348/intel-core-i5-4200m-processor-3m-cache-up-to-3--10-ghz.html (dostopano 21 april 2020).

[16] G. Paoloni, How to Benchmark Code Execution Times on Intel IA-32 and IA-64 Instruction Set Architectures, (2010). https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/ia-32-ia-64-benchmark-code-execution-paper.pdf.

[17] M. Dworkin, Recommendation for block cipher modes of operation: Methods and Techniques, 2001. doi:10.6028/NIST.SP.800-38a.

[18] A.C. Leon, Kruskal–Wallis test, in: Compr. Clin. Psychol., Elsevier, 1998: pp. 243–285. doi:10.1016/b0080-4270(73)00264-9.

[19] N.R. Smalheiser, The Mann-Whitney U Test, in: Data Lit. How to Make Your Exp. Robust Reprod., 2017. https://www.sciencedirect.com/book/9780128113066/data-literacy.

[20] E.W. Weisstein, Bonferroni Correction, (n.d.). https://mathworld.wolfram.com/BonferroniCorrection.html (dostopano 21 april 2020).

[21] J. Cohen, Statistical power analysis for the behavioral sciences, 1977.

[22] Cryptography and Encryption Libraries, LibHunt. (2020). https://cpp.libhunt.com/categories/661-cryptography (dostopano 22 april 2020).

▪

Dr. Marko Kompara je raziskovalec in asistent na Fakulteti za elektrotehniko, računalništvo in informatiko, Univerze v Mariboru. Doktorat iz računalništva in informatike je pridobil leta 2019. Tematika doktorske disertacije se je nanašala na protokole vzpostavitve ključa v strojno omejenih okoljih. Trenutno deluje kot raziskovalec na Horizon 2020 projektu CyberSec4Europe. Njegova raziskovalna področja vključujejo: zasebnost, kriptografija, zaščita brezžične komunikacije in zaščita informacijskih sistemov.

▪

Tomi Jerenko, magister inženir informatike in tehnologij komuniciranja, je pridobil svoj naziv na Fakulteti za elektrotehniko, računalništvo in informatiko v Mariboru leta 2019. Trenutno je zaposlen v Berlinu pri največjem evropskem ponudniku oblačne infrastrukture, imenovanem 1&1 IONOS, kjer sodeluje v razvoju z ekipama za aplikacijsko varnost in zagotavljanje kakovosti. Hkrati je tudi magistrski kandidat za naziv magister znanosti na nemški fakulteti Brandenburgische Technische Universität Cottbus-Senftenberg iz področja kibernetske varnosti.

▪

Dr. Marko Hölbl je docent na Fakulteti za elektrotehniko, računalništvo in informatiko, Univerze v Mariboru. V preteklosti je uspešno sodeloval v več nacionalnih in mednarodnih projektih trenutno pa je med drugim tudi lokalni koordinator Horizon 2020 projekta CyberSec4Europe. Deluje tudi kot generalni tajnik CEPIS LSI in EAEEIE, član ECSO WG 6 in član izvršnega odbora SDI. Njegova raziskovalna področja vključujejo: kriptografija, zaščita informacijskih sistemov, varnost na spletu, zaznavanje varnosti in zasebnosti na spletu, podatkovne baze in analiza podatkov.

.

Sentimeter: Interdisciplinarni pristop k izdelavi medijskega portala

Ajda Pretnar[1], Dan Podjed[2], Marko Bajec[1], Slavko Žitnik[1]
[1] Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, Ljubljana
[2] Znanstvenoraziskovalni center Slovenske akademije znanosti in umetnosti, Novi trg 2, Ljubljana
ajda.pretnar@fri.uni-lj.si, dan.podjed@zrc-sazu.si, marko.bajec@fri.uni-lj.si, slavko.zitnik@fri.uni-lj.si

Izvleček

Spletni portal Sentimenter je namenjen prikazovanju sentimenta v medijskih objavah za različne družbene skupine. Te smo identificirali z interdisciplinarnim pristopom, pri čemer smo izhodiščne skupine določili z vprašalnikom ter metodo hierarhičnega razvrščanja v skupine, dokončno pa smo jih potrdili in izoblikovali z intervjuji in fokusnimi skupinami. Na podlagi družbenih skupin smo oblikovali zaslonske maske medijskega portala, ki so bile oblikovane iterativno v sodelovanju s končnimi uporabniki. V fokusni skupini smo s predstavniki petih družbenih skupin, ki smo jih definirali, portal pregledali in preoblikovali po meri ljudi. Celotna raziskava je trajala pol leta. Z interdisciplinarnimi pristopi, ki združujejo antropologijo in podatkovno rudarjenje, smo tako v kratkem času oblikovali izdelek, ki je uporabnikom prijazen, razumljiv in zanimiv.

Ključne besede: sentiment, antropologija, interdisciplinarni pristopi, mešane metode, podatkovno rudarjenje

Abstract

The web portal Sentimenter monitors the sentiment in media clips for different social groups. We identified the social groups with an interdisciplinary approach. We used a questionnaire and hierarchical clustering to form preliminary groups and confirmed them through interviews and focus groups. The final social groups were used as a basis for the design of wireframes for the portal. The wireframes were created iteratively in collaboration with the end users. In the final focus group which consisted of the representatives of all social groups, we reviewed and redesigned the portal to fit the needs of the users. The entire research lasted six months. With interdisciplinary approaches that combine anthropology and data mining, a user-friendly, understandable and interesting product can be produced in a short period.

Keywords: Sentiment, anthropology, interdisciplinary approaches, mixed methods, data mining.

1 UVOD

Interdisciplinarni pristopi k snovanju digitalnih rešitev niso novost (Suchman, 1985; Podjed in dr., 2016), a se jih kljub temu še vedno razmeroma redko uporablja v praksi. Vsaj za antropologijo velja, da so raziskave dolgotrajne in potekajo na majhnem vzorcu ljudi, kar običajno ne prepriča vodij raziskovalno-razvojnih skupin, da se splača te pristope uporabiti za snovanje digitalnih rešitev. Takšne predpostavke moramo ovreči. Prvič, antropološke metode niso nujno dolgotrajne. Fokusne skupine so ena od metod, ki je hitra in učinkovita pri snovanju izdelkov in vzpostavljanju povratne zanke med načrtovalci in razvijalci ter uporabniki oziroma prejemniki. Hkrati se tudi v antropološkem raziskovanju vedno bolj uveljavljajo kvantitativni pristopi (Pretnar in Podjed, 2019), ki omogočajo hitre in učinkovite analize vprašalnikov in tekstovnih gradiv. Drugič, t. i. »bogati podatki« (angl. *thick data*), ki jih pridobimo z antropološkim raziskovanjem, lahko uspešno dvignejo tudi kazalce uspešnosti (West, 2014; Madsbjerg in Rasmussen, 2014; Pearson, 2015), kar se odraža v novih strankah in njihovem zadovoljstvu, večji prodaji in s tem povečanemu dobičku.

V pričujočem prispevku predstavimo interdisciplinirani pristop k oblikovanju medijskega portala, pri katerem povežemo antropologijo in računalništvo. Želeli smo oblikovati spletno mesto, kjer bi širša javnost lahko dostopala do medijskih objav, jih pregledovala in analizirala. Pri tem nismo želeli izdelati

še enega portala z zbirko novic, temveč smo hoteli celotno izkušnjo približati uporabniku in jo obogatiti s podatki, relevantnimi za skupine ljudi, ki bodo dostopale do informacij.

Ideja o izdelavi t.i. »družbenega barometra« je nastala v okviru sodelovanja med Fakulteto za računalništvom in informatiko Univerze v Ljubljani in podjetjem PressClipping. Podjetje je vodilno v Sloveniji na področju priprave medijskih pregledov, zato ima vpogled v dnevno dogajanje in stanje v družbi, ki se odraža v slovenskih elektronskih in tiskanih medijih. V okviru interne prenove sistemov in dodane analitike smo skupaj izoblikovali idejo o »družbenem barometru,« ki bi v obliki nadzorne plošče javnosti prikazoval aktualno stanje v družbi. Analize bi bile prikazane po posameznih skupnostih, ki pa jih je bilo predhodno treba jasno identificirati.

Prvi korak k osebni noti portala je bil prilagajanje družbenim skupinam, ki bodo uporabljale rešitev. Vsebine nismo nameravali prilagoditi vsakemu posamezniku, saj bi to zahtevalo vpis obiskovalca z uporabniškim računom, s čimer bi vzpostavili dodatno oviro pri uporabi. Hkrati nismo želeli, da je portal preveč splošen, zato smo si zamislili, da je vsebina prilagojena različnim družbenim skupinam. Pri tem smo seveda morali najprej ugotoviti, kaj in katere te skupine sploh so. Za njihovo identifikacijo smo uporabili tako antropološke kot računske pristope.

Drugi korak je bilo oblikovanje zaslonskih mask, ki bi uporabnika intuitivno vodile po vsebinah. Zaslonske maske smo oblikovali iterativno, kar pomeni, da smo funkcionalnosti izboljševali v več fazah, pri čemer je bila ključna faza fokusna skupina, kjer so potencialni uporabniki izpostavili, kateri elementi so dobro zasnovani in kaj bi bilo potrebno spremeniti. Končni izdelek je torej rezultat interdisciplinarnega sodelovanja med računalničarji, antropologi in podjetjem, ki ni zasnovan zgolj za uporabnike, temveč predvsem v sodelovanju z njimi.

2 METODOLOGIJA

Pri razvoju rešitev, ki so v izhodišču interdisciplinarne narave, je ključno razumevanje vseh vidikov produkta. V primeru »družbenega barometra« je tako pomembno, da najprej razumemo potrebe in želje uporabnikov, na podlagi katerih lahko začnemo s tehnično izvedbo. Zato smo skupino razširili s strokovnjaki s področja antropologije, s katerimi smo skupaj dosegli končni cilj. Pri zasnovi je bilo treba pregledati vse medijske vire, na podlagi katerih se bo izdelala nadzorna plošča. Nadalje je bilo treba identificirati družbene skupine, skušati razumeti njihova ključna zanimanja ter ugotoviti, na kakšen način čimbolj jasno in preprosto prikazati rezultate analiz. Med raziskavo smo delovni naslov »družbeni barometer« zamenjali z imenom Sentimeter, ki z eno besedo jasno poudari namen nadzorne plošče. Poleg tega je ime tudi skladno s klasifikacijo čustvovanj, ki smo jo izbrali za prikaz kategorij novic.

Portal smo zasnovali z raznovrstnimi metodološkimi pristopi. V začetni fazi razvoja smo najprej oblikovali vprašalnik o samoidentifikaciji, ki je anketirance spaševal po oznakah, s katerimi se najbolj poistovetijo. Vprašalnik smo izbrali, ker je stroškovno in časovno primeren pristop za zbiranje izhodiščnih podatkov, hkrati pa samoidentifikacijski vprašalniki uspešno razložijo preference in vedenje ljudi (Kuo in Margalit, 2012). Vprašalnik smo izdelali z orodjem EnKlik Anketa in ga posredovali po različnih spletnih kanalih (forumi, družbena omrežja in obvestila na Fakulteti za računalništvo in informatiko Univerze v Ljubljani). Dostopen je bil avgusta in septembra 2019, skupno 5 tednov. Čeprav vprašalnik, ki je dostopen izključno na spletu, ne zajame celotne populacije (Pay Ton, 2020), je za našo raziskavo pomemben predvsem tisti del populacije, ki dostopa do spleta. Glavni del vprašalnika – in poleg demografskih podatkov edini obvezni – je zajemal izbiro treh najpomembnejših identifikacijskih oznak (Priloga 1), pri čemer so posamezniki lahko dopisali svojo oznako, če na izbiro ni bilo tiste, ki je zanje najprimernejša.

Rezultate vprašalnika[1] smo analizirali z orodjem za strojno učenje Orange (Demšar in dr., 2013). Za identifikacijo skupin smo uporabili spremenljivke s samoidentifikacijami, ki smo jih s hierarhičnim razvrščanjem na podlagi Jaccardove razdalje in Wardove mere podobnosti razvrstili v pet skupin. Značilne lastnosti posameznih skupin smo določili s testom enake verjetnosti.

Splošno analizo dostopa do medijskih vsebin smo avgusta 2019 opravili s fokusno skupino, ki je predstavljala eno od odkritih identifikacij. S prebivalci Centra starejših Trnovo smo se pogovarjali, kako starejši ljudje dostopajo do novic, katere medije

[1] Dostopno na http://zitnik.si/research/files/anketa-druzbeni-barometer.csv.

uporabljajo ter katerim medijem zaupajo (Priloga 2). Fokusno skupino smo izvedli, ker temelji na odprtem pogovoru o vnaprej določeni temi, pri čemer je bila v našem primeru osrednja tema uporaba medijev in način dostopa do novic. Raziskovalna tehnika, ki smo jo uporabili, omogoča sočasno obravnavo različnih tematik, pri čemer nove tematike izhajajo iz pogovora – to pomeni, da udeleženci fokusne skupine narekujejo smer pogovora na podlagi osebnih izkušenj ali članstva v posamezni družbeni skupini (Morgan, 1997; Zavella, 2014; Stewart, 2017). Srečanja so se prebivalci doma starejših občanov udeležili na pisno vabilo, med pogovorom pa smo kot moderatorji zagotovili glas vsakemu posamezniku, ki je med skupinsko debato želel izraziti mnenje. Fokusna skupina je služila osmišljanju in podrobnejši identifikaciji ene od predvidenih družbenih skupin, torej upokojencev.

V naslednji projektni fazi, ko smo že identificirali družbene skupine, smo oktobra 2019 opravili še drugo fokusno skupino s predstavniki vseh petih družbenih skupin, za katere smo pripravili medijski portal. Predstavnike smo na srečanje povabili pisno, izbrani pa so bili po načelu »snežne kepe« – to pomeni, da smo skušali pritegniti znance, ki so ustrezali kriterijem posamezne družbene skupine, ter jih prosili, naj povabijo še svoje znance. To srečanje je bilo namenjeno natančni analizi prototipa medijskega portala, pri čemer smo z udeleženci po načelu sodelovalnega oblikovanja (Podjed, 2019) identificirali glavne prednosti in slabosti novega izdelka ter zbrali predloge za izboljšavo. Ponovno je bil format pogovora odprt, pri čemer smo z udeleženci pregledali predloge zaslonske maske nastajajočega portala ter za posamezno funkcijo vprašali, ali je razumljiva oziroma kaj pričakujejo, da se zgodi ob kliku na posamezen del portala (Priloga 3).

3 DRUŽBENE SKUPINE

Družbene skupine so, če sledimo Greenwoodovi definiciji, ki jo je prilagodil po Émilu Durkheimu, načini vedenja, razmišljanja in čustvovanja posameznikov, ki so pogojeni s tem, da se drugi člani skupine predstavljajo in poistovetijo s temi istimi načini (Greenwood, 2003). Tako posameznik še vedno ostane samosvoj v razmerju do sveta in družbe, hkrati pa je del skupnosti, ki povzema določen del oziroma vidik tega odnosa. Skozi prizmo konceptualnega obrata iz objektivizma s preddefinirami skupinami v bolj kontekstno odvisno osredotočanje na posameznika kot nosilca družbenih interakcij (Ville in Guérin-Pace, 2005) so ljudje tisti, ki definirajo družbene skupine in ne obratno, torej da bi bili ljudje definirani z družbenimi skupinami.

Stroga zamejitev družbenih skupin je nemogoča, saj lahko posameznik ali posameznica sočasno pripada več družbenim skupinam, hkrati pa je pripadnost odvisna od tega, ali se oseba poistoveti s skupino ali pa jo vanjo umestijo drugi. Kljub visoki stopnji prekrivanja med skupinami in velikemu številu skupin, ki jih tvorijo človeške družbe, je smiselno identificirati nekaj glavnih. Ljudje smo vajeni razmišljati v kategorijah, saj prevelika osredotočenost na razlike pomeni zahtevo po ogromni količini spomina za shranjevanje informacij in predstavlja preveliko kognitivno obremenitev. Zato je človek razvil sposobnost prepoznavanja višjenivojskih struktur in podobnosti med njimi, kar mu omogoča razlikovanje med ključnimi kategorijami in koncepti (Seger in Miller, 2010).

Definicija in analize družbenih skupin sicer sodijo na širše področje družboslovja in humanistike, mi pa smo se tega lotili z bolj specifičnimi antropološkimi ter interdisciplinarnimi pristopi, saj smo želeli, da predlagane kategorije odražajo družbeno krajino, kot jo ljudje sami razumejo in občutijo. Pri snovanju družbenih kategorij smo se sicer delno oprli tudi na razmejitev, ki jo ponujajo državne statistične službe (Razpotnik, 2018; glej tudi SURS, 2020) in uradi, glavni del pa smo osnovali na podlagi fokusnih skupin in intervjujev. Dodatno smo kategorije preverili v kontekstu medijskega poročanja ter s pregledom družbenih medijev (Facebook, Twitter, Instagram, LinkedIn, YouTube itd.). Na tej podlagi smo želeli opredeliti čim širše skupine, torej take, ki zajemajo velik spekter ljudi, hkrati pa smo želeli poiskati in opredeliti tiste skupine, ki so najbolj aktivne in lahko prispevajo največ povratnih informacij. Pri tem smo se oprli na državno statistiko in pregledali obstoječe demografske skupine. Sočasno smo želeli, da bi bile skupine čim bolj disjunktne, s čimer bi dosegli, da se posameznik lahko prepozna vsaj v eni od skupin, četudi le delno.

Najbolj očitna in pogosto uporabljana kategorizacija je po spolu, ki je običajno binarna. Predlagali smo, da te kategorizacije v projektu ne uporabimo, in sicer zato, ker z njo vnaprej stereotipiziramo spremljanje medijskih objav po spolih in hkrati utrjujemo stereotipe. Poleg tega s tovrstno binarno delitvijo iz-

pustimo skupino ljudi, ki se ne identificira ne z moškim ne z ženskim spolom ali pa se ne želi opredeliti.

Končne družbene skupine smo opredelili z vprašalnikom. Ta je bil zasnovan tako, da so se ljudje samoidentificirali, torej izbrali kategorije, ki so zanje smiselne, nato pa smo z gručenjem profilov samoidentifikacij oblikovali pet družbenih skupin.

3.1 Tipi skupin

Vnaprej smo predvideli oblikovanje okvirno petih družbenih skupin, ki so dovolj splošne in zastopane, in sicer so to upokojenci, študenti in dijaki, zaposleni, starši ter brezposelni. Te kategorije naj bi predstavljale glavni del uporabniškega vmesnika. Uporabnik lahko tako izbere kategorijo, ki ga zanima, in pregleda novice, ki se navezujejo na izbrano kategorijo. Pet identificiranih skupin se uporabi tudi za računanje t. i. *sentimenta novic*, s katerim prikazujemo prepoznan sentiment, povezan s posamezno skupino. Tipi so bili izhodiščno določeni glede na podatke SURS, druge novičarske portale in družbena omrežja.

3.1.1 Upokojenci

Upokojenci so osebe, ki prejemajo pokojnino in nimajo statusa zaposlenega, brezposelnega ali študenta. So široka demografska skupina, ki obsega 20 % prebivalstva, do leta 2030 pa naj bi narasla na 25 %. Zaznamuje jih nizka stopnja zanimanja za zunanje okoliščine, nizka kupna moč ter visoka stopnja ranljivosti (Hohnerlein, 2019; SURS, 2020).

3.1.2 Študenti in dijaki

Študenti so osebe, ki niso zaposlene, brezposelne ali upokojene, in so vpisane v programe izobraževanja. V Sloveniji zajemajo 9 % odstotkov prebivalstva.[2] So del populacije, ki ima volilno pravico, vendar pa jih večina še ne prispeva v državni proračun. Običajno so proaktivni, z izrazitimi interesi. Zaznamuje jih visoka stopnja potrošništva (SURS, 2020) ter dobra socialna povezanost. Med študente lahko dodamo še dijake, ki sicer pretežno še nimajo volilne pravice, vendar je to demografski segment, ki se postopno vključuje v družbo kot aktivni del prebivalstva, hkrati pa ga močno zaznamujejo politične in druge odločitve, sprejete v tem času. Dijaki za razliko od študentov večinoma nimajo volilne pravice, imajo pa podobno izrazito oblikovane interese.

3.1.3 Starši

Leta 2018 je bilo v Sloveniji 577.544 uradno zabeleženih družin, od tega 150.004 brez otrok. Pojem družine je izjemno heterogen (enostarševske družine, istospolna partnerstva, družine brez otrok). Po definiciji SURS je družina življenjska skupnost oseb v zasebnem gospodinjstvu, kar vključuje tako poročene pare kot zunajzakonske skupnosti ter pare z otroki. Pri tej družbeni skupini se omejimo zgolj na starše. Starši so skupnost dveh oseb, ki imata potomce oziroma sta skrbnika mladoletnih oseb. Zaznamuje jih visoka stopnja zanimanja za zunanje okoliščine, zlasti s področja zdravstva, šolstva in ekonomije.

3.1.4 Zaposleni

Zaposleni so po definiciji SURS osebe, starejše od 15 let, ki imajo pogodbo o zaposlitvi ali opravljajo kakšno drugo priznano obliko dela. Mednje se štejejo delavci, prekarno zaposleni, samozaposleni in vodstveni kader. Gre za široko heterogeno skupino, ki je le delno vezana na demografijo. Skupina izkazuje močnejši interes za gospodarske razmere (ekonomija), infrastrukturo ter zdravstvo. V ta segment je težje zajeti kmetovalce, saj predstavljajo specifično podpopulacijo. So samozaposleni ter imajo rahlo drugačne interese, zlasti jih seveda zanimajo kmetijska politika, subvencije, podnebne razmere in vreme.

3.1.5 Brezposelni

To so ljudje, ki nimajo redne zaposlitve ali ni samozaposlen, so pa v aktivnem iskanju zaposlitve. Kategorija je pretežno prehodna, saj naj bi po definiciji brezposelni po nekem času našli zaposlitev in tako prešli med zaposleno prebivalstvo.

3.2 Rezultati

Vprašalnik, ki smo ga razposlali, je pravilno in v celoti izpolnilo 230 ljudi. V tem delu raziskave je sodelovalo 51 moških in 161 žensk (24 % proti 76 %) ter največ mladih odraslih (21–40 let, 53 %). Najizrazitejše samoidentifikacije so bile prijatelj (29 %), ljubitelj živali (27 %), partner (26 %) in starš (25 %), s čimer smo pridobili izhodiščne družbene skupine za oblikovanje portala. V konjičke usmerjene samoidentifikacije so močnejše kot generacijske ali ekonomske, vendar jih je težje strniti v enotno skupino ter identificirati novice, ki so zanje specifične. Poleg samoidentifikacij smo

[2] Delež se nanaša na vse učence, dijake in študente, starejše od 15 let

anketirance vprašali tudi, kaj jim predstavlja stanje v družbi. Večina le-tega povezuje z gospodarskimi (77 %) in političnimi razmerami (60 %), manj z osebnim zadovoljstvom. Ker se gospodarstvo in politika močno odražata v medijskih objavah, portal predstavlja ustrezen kanal za prikazovanje stanja v družbi.

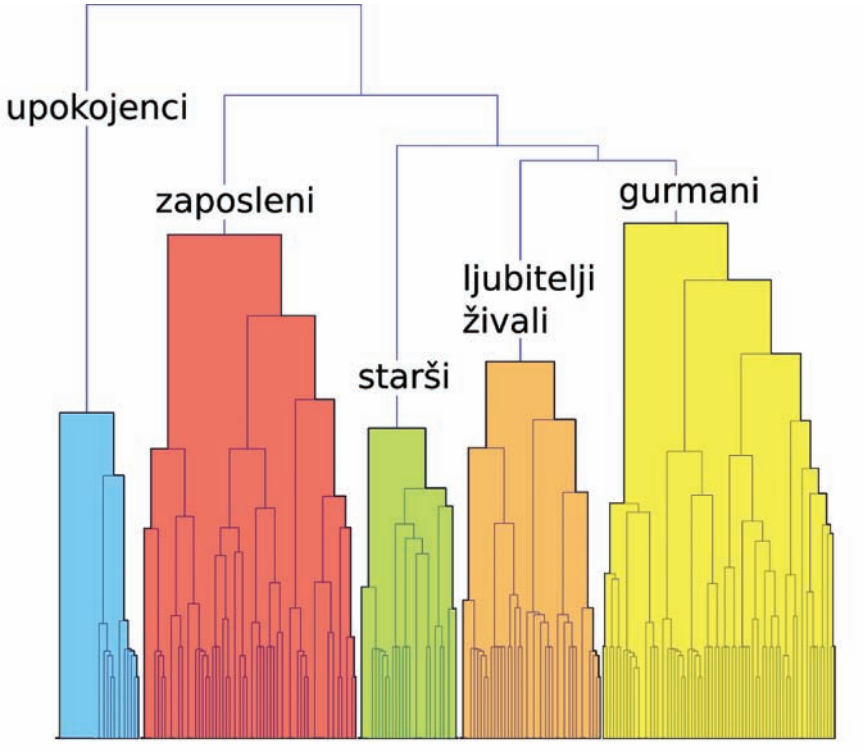
3.3 Gručenje

Rezultate, pridobljene z vprašalnikom, smo analizirali s pomočjo orodja za strojno učenje in podatkovno rudarjenje Orange. Skupine smo določili s tehniko odkrivanja skupin in meritev podobnosti. Odločili smo se, da želimo oblikovati pet skupin, saj je tako število gruč še mogoče razložiti tudi kvalitativno, hkrati pa je dovolj verjetno, da se bo vsak posameznik lahko identificiral z vsaj eno izmed njih. Pet skupin je pomenilo mejo tudi iz praktičnih razlogov, saj bi uporabniški vmesnik portala lahko postal prekompleksen, če bi določili več skupin.

Profile samoidentifikacij iz vprašalnika, ki so bili v tabeli zapisani z binarnimi vrednostmi, smo uporabili za računanje podobnosti med odgovarjajočimi. Podobnost smo merili z Jaccardovo razdaljo, saj deluje na principu razlike med preseki značilk, kar je koristno, če želimo oblikovati homogene in disjunktne skupine. Nato smo za določanje skupin uporabili hierarhično razvrščanje z Wardovo mero razdalje.

Izrisani dendrogram (Slika 1) smo prerezali tako, da smo oblikovali pet skupin. Značilnosti posameznih skupin smo določili z računanjem vrednosti χ^2, pri čemer smo iskali, po čem se posamezna skupina pomembno razlikuje od drugih. χ^2 v tem primeru torej računa razliko med izbrano skupino in ostalimi štirimi skupinami.

Prva skupina so upokojenci (86 % upokojencev je v tej skupini, $\chi^2 = 195{,}18$). V drugi skupini prevladujeta identifikaciji »prijatelj« (67 %, $\chi^2 = 89{,}90$) in »delavec« (76 %, $\chi^2 = 48{,}22$). V precejšnjem deležu so v tej skupini tudi študenti (62 %, $\chi^2 = 24{,}28$), čeprav manj izrazito kot druge identifikacije. V tretji skupini so starši oziroma skrbniki (52 %, $\chi^2 = 24{,}28$), pri čemer so – kljub razmeroma majhnemu deležu vseh staršev – v tej skupini vsi anketirani starši (z drugimi besedami, ni »ne-staršev«). Četrto skupino močno zaznamujejo ljubitelji živali (62 %, $\chi^2 = 116{,}52$), peto pa, presenetljivo, gurmani (88 %, $\chi^2 = 45{,}34$) in umetniki (75 %, $\chi^2 = 26{,}33$). Pri tem velja poudariti, da je peta skupina največja in tudi najbolj heterogena, v njej pa prevladujejo t. i. prostočasne identifikacije.



Slika 1: **Prepoznanih 5 skupin, pri čemer sta si skupini ljubiteljev živali in gurmanov blizu.**

Spol in starost ne razlikujeta bistveno med samoidentifikacijami, s pričakovanima izjemama za upokojence in študente, ki sta najbolj vezani na starost. Zanimivo (a ne povsem nepričakovano) je, da so močne samoidentifikacije vezane na medčloveške odnose ter na konjičke. Tri izrazite identifikacije so »starš«, »upokojenec« ter »prijatelj«, kar pomeni, da smo do neke mere uspešno prepoznali vsaj dve kategoriji.

Glede na predhodno definirane skupine nam je dovolj dobro uspelo zajeti 3 od 5 skupin, in sicer upokojence, zaposlene in starše. Skoraj nihče se v vprašalniku ni identificiral kot brezposeln, najverjetneje zato, ker gre za prehodno stanje in ker ima ta oznaka negativno konotacijo. Študenti niso oblikovali lastne skupine, so pa močno zaznamovali precej številčno skupino zaposlenih.

3.4 Dostop do medijev

Drugi sklop vprašanj, ki smo jih zastavili, je obsegal način dostopa do novic. Rezultati vprašalnika kažejo, da večina ljudi dostopa do novic po družbenih omrežjih ter spletnih portalih, pri čemer je seveda treba upoštevati, da je bila anketa, ki smo jo izvedli, na voljo izključno na spletu. Skupina ljudi nad 60 let novice pridobi tudi po televiziji, medtem ko pri skupini 21-40 let prevladujejo spletni portali. Družbena omrežja dokaj enakomerno uporabljajo vse starostne skupine.

3.5 Stanje v družbi

Zadnji sklop vprašanj je bil povezan z interpretacijo besedne zveze »stanje v družbi«. Anketirance smo vprašali, kaj jim ta besedna zveza pomeni, pri čemer so lahko izbirali med naslednjimi možnostmi: gospodarske razmere, osebno zadovoljstvo, politične razmere, kakovost infrastrukture, dostop do zdravstva, dostop do šolstva, medosebni odnosi in stanje okolja. Večina anketirancev je izbrala bodisi gospodarske bodisi politične razmere, močno prisotni pa so bili tudi medosebni odnosi. Kljub temu je zamisel stanja v družbi še vedno povezana z gospodarstvom in politiko, kar se pomembno povezuje z medijskimi objavami. Če namreč merimo sentiment medijskih objav za kategoriji gospodarstvo in politika, potem lahko predvidoma razmeroma dobro zajamemo in predstavimo tudi splošno stanje v družbi.

3.6. Končne skupine

Za oblikovanje portala smo obdržali uspešno prepoznane tri skupine in sicer upokojence, starše in zaposlene. Kot četrto skupino smo dodali mlade, ker so študentske in dijaške teme vseeno izrazite, študenti pa so močno zaznamovali drugo skupino (zaposleni). Skupino brezposelnih smo popolnoma opustili, saj gre za prehodno identifikacijo, ki je po svoji konotaciji razmeroma negativna in posameznikov dolgoročno ne zaznamuje. Zadnjo skupino smo poimenovali ekologi, čeprav to ni najbolj ustrezno. V to skupino načeloma sodijo ljubitelji živali in hobijisti. Za ekologijo (pravzaprav za okoljevarstvo) smo se odločili, ker smo k oblikovanju portala želeli pristopiti tudi nekoliko aktivistično in poudariti ključne teme sedanjosti, med katere nedvomno sodi varovanje okolja. Prostočasne identifikacije smo bili primorani izpustiti, saj so preveč heterogene in jih je težko opredeliti v kontektsu medijskih objav.

Prepoznane skupine smo torej oblikovali zlasti na podlagi rezultatov vprašalnika, pri čemer so anketiranci s pomočjo samoidentifikacije določili tiste, ki so najpogostejše oziroma najbolj zaznamujejo posamezne skupine. Končni predlog smo oblikovali s pomočjo dveh fokusnih skupin, kjer so sogovorniki potrdili ali zavrnili jasnost skupin, razložili, kako jih sami interpretirajo, ter oblikovali svoje predloge. Pet skupin je tako rezultat sodelovalnega pristopa z uporabniki ter tehnike mešanih metod z uporabo kvantitativnih analiz vprašalnikov ter kvalitativnih interpretacij drugih izsledkov.

4 SENTIMETER

Tehnična izvedba Sentimetra temelji na semantičnih analizah, ki jih izvaja podjetje PressClipping in so na voljo zgolj naročnikom v okviru njihovih naročniških portalov. V skladu z avtorskimi pravicami in dostopnostjo vsebin bo podjetje javnosti ponudilo omejen nabor rezultatov analiz za namen javne nadzorne plošče Sentimeter. V nadaljevanju opišemo uporabniški vidik nadzorne plošče in predstavimo, kako je zasnovana.

4.1 Uporabniški vmesnik

Uporabniški vmesnik portala smo zasnovali v več fazah. V prvi fazi smo pripravili izhodiščne zaslonske maske, ki so bile opora za razvoj idej, testiranje povezav med elementi in grafične predstavitve informacij. Kot najpomembnejše vprašanje smo v tem delu identificirali, kako prikazati sentiment v novicah. Možnosti za prikaz sentimenta so bile številčna ocena med 0 in 1 (oziroma odstotki), z barvanjem

deležev pozitivnega ali negativnega sentimenta ter stiliziran prikaz nasmejanih obrazov oziroma t. i. »smeškov« (Slika 2).

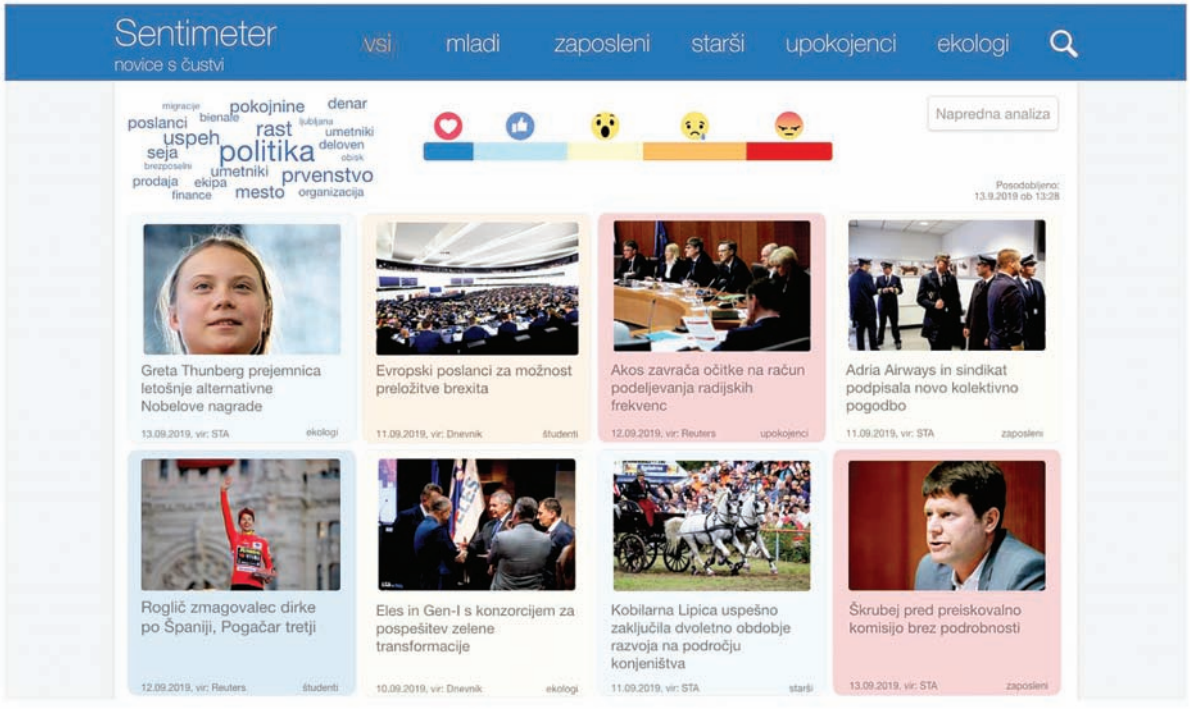
Ker so se pri pogovoru z deležniki te opcije izkazale za nezadostne, smo razširili idejo smeškov z emotikoni. Ti so v sodobnem času splošno prisotni kot povratna informacija na družabnih omrežjih, s čimer so ljudem blizu in poznani. Poleg tega lahko z emotikoni prikažemo širšo paleto čustev kot zgolj s prikazom deleža pozitivnega sentimenta. Tako smo se odločili, da za prikaz čustev uporabimo podobne emotikone, kot jih uporablja omrežje Facebook.

Slika 2: **Naložen palični grafikon prikazuje razmerje sentimenta v novicah. Čustvo je označeno z emotikonom nad deležem, ki ga to predstavlja v novici.**

Izbor smeškov je vezan na čustva, ki jih posameznik doživlja ob branju novic. Glavne čustvene kategorije, in sicer jezo, veselje, strah, gnus ter presenečenje, smo povzeli po Ekmanu (1992), saj so izmed splošnih psiholoških čustvenih kategorizacij (Ekman, 1992; Plutchik, 1980; McNair in dr., 1971) najbolj interpretabilne, poleg tega pa jih je razmeroma veliko. Dodali smo še kategorijo navdušenja, da so sentimenti polarno uravnoteženi. V Ekmanovi klasifikaciji so namreč vsaj tri čustva negativna (jeza, gnus, strah), eno je neopredeljivo (presnečenje), zgolj eno pa pozitivno (veselje).

Sentiment – torej klasifikacija čustev v besedilih na podlagi čustveno zaznamovanih besed – je predstavljen z vrstico, ki prikazuje delež barv, nad vsako barvo pa je prikazan tudi emotikon, ki označuje pripadajoče čustvo. Čustva so urejena po valenci; na levi je najbolj pozitivno čustvo, na desni najbolj negativno, pri čemer smo presenečenje označili kot nevtralno, ker menimo, da ima tako pozitivno kot negativno konotacijo. Čustva se odražajo tudi v barvi okna posamezne novice, pri čemer je novica obarvana s prevladujočim čustvom. Ob kliku na posamezen emotikon se uporabniku prikažejo novice, kjer prevladuje izbrano čustvo.

Prva stran prikazuje vse novice (Slika 3), uporabnik pa lahko prehaja med družbenimi skupinami, ki smo jih identificirali v predhodni raziskavi. Za vsako družbeno skupino portal prikaže novice, za katere prepozna, da so povezane s to skupino. Sočasno se posodobita tudi prikaz čustev in oblak besed v zgornjem levem kotu. Tudi oblak besed je interaktiven in



Slika 3: **Vstopna stran portala Sentimeter prikaže najnovejše objave, obarvane s sentimentom, deleže sentimenta ter oblak besed, ki prikazuje najpogostejše besede novic za izbrano podskupino**.

Slika 4: **Prikaz funkcionalnosti napredne analize portala Sentimeter za skupino mladih.**

omogoča iskanje novic po ključnih besedah. Čeprav oblaki besed niso najbolj primerne vizualizacije za prikaz pogostosti besed, pa so ljudem blizu in so jih tudi v fokusni skupini, ki smo jo izvedli, omenili kot preferenčne.

V desnem zgornjem kotu portala je dostop do naprednih analiz (Slika 4). V tem delu uporabniku ponudimo dodatne možnosti raziskovanja vsebin, in sicer po posameznih čustvih skozi čas. Palični diagram prikaže podrobno razdelitev sentimenta za izbrano družbeno skupino, stolpčni grafikon pa pogostost izbranega sentimenta v času. Na desni strani se prikažejo novice za izbran sentiment (na Sliki 4 je, na primer, prikazano navdušenje).

5 SKLEP

Vsak medijski portal je namenjen predvsem prejemnikom novic. Zato je bil primarni cilj projekta ugotoviti, kaj ljudje od takega portala pričakujejo, kaj jih zanima, katere vsebine so zanje relevantne in kaj bi jih pritegnilo k dolgotrajni uporabi. Nadalje smo želeli oblikovati rešitev, ki temelji na željah in potrebah ljudi, ki bodo rešitev uporabljali. V sodelovanju z njimi smo odkrivali in določali različne možnosti uporabe portala. Izločili smo, denimo, funkcije, ki so se nam zdele očitne, njim pa ne. Poenostavili smo tudi prekompleksne rešitve ter upoštevali predloge za izboljšanje uporabniške izkušnje. Kombinacija tehnik z različnih znanstvenih področij – predvsem pa antropologije in računalništva – nam je omogočila hitro, učinkovito in preprosto oblikovanje tehnološke rešitve po meri ljudi, kar bi težko dosegli z monodisciplinarnimi pristopi.

Interdisciplinarni pristopi so, kot smo pokazali s tem prispevkom, uspešni tako za akademske raziskave kot tudi za aplikativne in razvojne projekte. Kombinacija antropologije in računalništva je posebej uporabna, saj lahko tako oblikujemo digitalne rešitve, ki so relevantne za ljudi in preproste za uporabo. Kot smo prikazali v prispevku, lahko antropologi (in drugi raziskovalci s področja humanistike in družboslovja) uspešno identificirajo potrebe posameznikov in družbenih skupin ter prepoznajo navade in prakse ter tudi ovire pri doseganju ciljev, medtem ko računalničarji z znanji programiranja in poznavanjem umetne inteligence, oblikujejo tehnološko dovršene izdelke z dodano vrednostjo. Ključ do razvoja ljudem prijaznih in uporabnih rešitev, ki upoštevajo družbeno raznolikost, je po našem mnenju v interdisciplinarnem sodelovanju.

Zahvala

Avtorji se zahvaljujejo podjetju Press Clipping d.o.o. in njihovim sodelavcem (Boštjan Vilčnik, Uroš Topolovec), ki so omogočili izdelavo raziskave.

LITERATURA

[1] Suchman, L. (1985). *Plans and Situated Actions: The Problem of Human-Machine Communication*. New York: Cambridge University Press.

[2] Podjed, D., Group. M. in Bezjak Mlakar, A. (2016). Applied Anthropology in Europe: Historical Obstacles, Current Situation, Future Challenges. *Anthropology in Action*, 23(2), 53-63. https://doi.org/10.3167/aia.2016.230208

[3] Pretnar, A. in Podjed, D. (2019). Data Mining Workspace Sensors: A New Approach to Anthropology. *Prispevki za novejšo zgodovino*, 59(1), 179-197. https://ojs.inz.si/pnz/article/view/318

[4] Madsbjerg, C. in Rasmussen, M.B. (2014). An Anthropologist Walks into a Bar. *Harvard Business Review*, 92(3), 80-88. https://hbr.org/2014/03/an-anthropologist-walks-into-a-bar

[5] Pearson, C. (14 julij 2015). My Two Years as an Anthropologist on the Photoshop Team. https://medium.com/startup-frontier/my-two-years-as-an-anthropologist-on-the-photoshop-team-e700acb7d3d5

[6] Kuo, A. in Margalit, Y. (2012). Measuring Individual Identity: Experimental Evidence. *Comparative Politics*, 44(4), 459-479. https://www.jstor.org/stable/23211822

[7] Pay Ton, J. (2020). Quantitative Analysis. V Honor and the Political Economy of Marriage: *Violence Against Women in the Kurdistan Region of Iraq* (str. 104-128). Rutgers University Press.

[8] Demšar, J., Curk, T., Erjavec, A., Gorup. Č., Hočevar, T., Milutinović, M., Možina, M., Polajnar, M., Toplak, M., Starič, A., Štajdohar, M., Umek, L., Žagar, L., Žbontar, J., Žitnik, M. in Zupan, B. (2013). Orange: Data Mining Toolbox in Python. *Journal of Machine Learning Researc*h, 14(Aug), 2349-2353. http://jmlr.org/papers/volume14/demsar13a/demsar13a.pdf

[9] Morgan, D. L. (1997). *Focus Groups as Qualitative Research*. 2. izdaja. Thousand Oaks, London in New Delhi: Sage Publications.

[10] Zavella, P. (2014). Focus Groups/Group Qualitative Interviews. V M. B. Schenker, X. Castaneda, A. Rodriguez-Lainz (ur.), *Migration and Health: A Research Methods Handbook* (str. 293-305). University of California Press.

[11] Stewart, D. W., Shamdasani, P. N. in Rook, D. W. (2007). *Focus Groups: Theory and Practice*. Thousand Oaks, London in New Delhi: Sage Publications.

[12] Podjed, D. (2019). Razvoj etnografsko utemeljene tehnološke rešitve. *Glasnik Slovenskega etnološkega društv*a, 59(1), 39-48.

[13] Greenwood, J. D. (2003). Social Groups and Social Explanation. *Noûs*, 31(1), 93-112. https://www.jstor.org/stable/3506206

[14] Ville, I. in Guérin-Pace, F. (2005). Identity in Question: The Development of a Survey in France. Population, 60(3), 231-258. https://www.cairn.info/revue-population-2005-3-page-277.htm

[15] Seger, C. A. in Miller, E. K. (2010). Category Learning in the Brain. *Annual Review of Neuroscience*, 33, 203-219. https://doi.org/10.1146/annurev.neuro.051508.135546

[16] Razpotnik, B. (2018). Metodološko pojasnilo: Socioekonomske značilnosti prebivalstva in selivcev. Ljubljana: Statistični urad Republike Slovenije. Dostopno na: https://www.stat.si/StatWeb/File/DocSysFile/7785/05-245-MP.pdf.

[17] SURS. (18 maj 2020). Dostopno na: https://www.stat.si/statweb.

[18] Hohnerlein, E. M. (2019). Pension indexation for retirees revisited – Normative patterns and legal standards. *Global Social Policy*, 19(3), 246-265. https://doi.org/10.1177/1468018119842028

[19] Ekman, P. (1992). An Argument for Basic Emotions. *Cognition & Emotion*, 6(3-4), 169-200. https://doi.org/10.1080/02699939208411068

[20] Plutchik, R. (1980). A General Psychoevolutionary Theory of Emotion. V *Theories of Emotion* (str. 3-33). Academic Press. https://doi.org/10.1016/B978-0-12-558701-3.50007-7

[21] McNair, D. M., Lorr, M., in Droppleman, L. F. (1971). Manual for the Profile of Mood States (POMS). San Diego: *Educational and Industrial Testing Service.*

▪

Ajda Pretnar je doktorska študentka na Oddelku za etnologijo in kulturno antropologijo Filozofske fakultete ter raziskovalka v Laboratoriju za bioinformatiko na Fakulteti za računalništvo in informatiko (oboje Univerza v Ljubljani). Ukvarja se z metodologijo interdisciplinarnega raziskovanja in uporabo pristopov strojnega učenja v humanistiki in družboslovju.

▪

Dan Podjed je docent in raziskovalec v Inštitutu za slovensko narodopisje Znanstvenoraziskovalnega centra SAZU in Inovacijsko-razvojnem inštitutu Univerze v Ljubljani ter predavatelj na Filozofski fakulteti Univerze v Ljubljani. Raziskovalno se posveča razmerju med ljudmi in sodobnimi tehnologijami ter sodeluje pri razvoju ljudem in okolju prijaznih rešitev. Med letoma 2010 in 2018 je vodil Mrežo za aplikativno antropologijo Evropskega združenja socialnih antropologov (EASA) in z njo ustanovil mednarodni simpozij Zakaj svet potrebuje antropologe (Why the World Needs Anthropologists), ki ga prirejajo od leta 2013.

▪

Marko Bajec je redni profesor na Fakulteti za računalništvo in informatiko Univerze v Ljubljani, kjer poučuje dodiplomske in podiplomske predmete s področja razvoja informacijskih sistemov in podatkovnih baz. Raziskovalno se ukvarja z metodami in pristopi k snovanju in razvoju informacijskih sistemov in obvladovanjem informatike ter v zadnjih letih predvsem s podatkovnimi tehnologijami za predstavitev, analizo in vizualizacijo podatkov. Leta 2009 je ustanovil Laboratorij za podatkovne tehnologije ter prevzel njegovo vodenje. Je član številnih domačih in tujih združenj, komisij in odborov. V okviru fakultete je vodil več aplikativnih in raziskovalnih projektov.

▪

Slavko Žitnik je docent na Fakulteti za računalništvo in informatiko Univerze v Ljubljani, kjer poučuje predmete s področja podatkovnih baz in obdelave podatkov. Raziskovalno se ukvarja z obdelavo naravnega jezika, predvsem na semantični ravni. Je predsednik Sveta za elektronske komunikacije RS, sodeluje pri organizaciji konferenc s področja informatike in pri projektih, povezanih z obdelavo podatkov na področju interneta stvari.

Priloga 1

Vprašanja v samo-identifikacijskem vprašalniku.

1. Katerim skupinam pripadate? Izberite tri kategorije, s katerimi se najmočneje identificirate. Če ustrezne kategorije ni med naštetimi, jo dopišite. (obvezno vprašanje, razvrsti naključno)
2. Preko katerih medijev dostopate do novic o dogajanju po Sloveniji in po svetu? (obvezno vprašanje)
   2.1 Katere časopise berete? (pogojen z 2)
   2.2 Katere radijske postaje spremljate? (pogojen z 2)
   2.3 Katere TV postaje spremljate? (pogojen z 2)
   2.4 Katera družbena omrežja uporabljate za dostop do novic? (pogojen z 2)
   2.5 Katere spletne portale spremljate? (pogojen z 2)
3. Pod besedno zvezo »stanje v družbi« si predstavljam:
4. Kako bi ocenili trenutno stanje v družbi?
5. Spol
6. V katero starostno skupino spadate?

Predlagane oznake za samoidentifikacijo:
starš/skrbnik, delavec/-ka, kmetovalec/-ka, umetnik/-ica, glasbenik/-ica, knjigoljub/-ka, upokojenec/-ka, ljubitelj/--ica živali, športnik/-ca, znanstvenik/-ca, poslovnež/-inja, aktivist/-ka, prijatelj/-ica, domoljub/-ka, inženir/-ka, modni navdušenec/-ka, gurman/-ka, popotnik/-ica, partner/--ka, ekolog/-inja, vernik/-ica, študent/-ka, brezposeln/-a, nič od naštetega

Priloga 2

Vprašanja za fokusno skupino Centra starejših Trnovo.

1. Koliko izmed vas ima tablico ali pametni telefon?
2. Koliko vas ima računalnik?
3. Kako dostopate do informacij? Preko česa?
4. Kje gledate televizijo? Ali jo skupaj?
5. Zakaj ne marate Skypa?
6. Berete časopise in revije?
7. Ali poznate, uporabljate družbena omrežja?
8. Zakaj niste na družbenih omrežjih?
9. Kako komunicirate z otroki, vnuki, vrstniki?
10. Katerim medijem zaupate? Katerim verjamete?
11. Katere medije spremljate?
12. Kje dobite informacije o volitvah? O kulturnih dogodkih?
13. Kaj vam je všeč in kaj vam ni všeč pri načinu podajanja informacij v izbranem mediju?
14. Ali kdaj pišete medijem oz. kako drugače izrazite svoje mnenje?
15. Ali nam tehnologije lahko olajšajo možnost odločanja?

Priloga 3

Vprašanja za fokusno skupino o zaslonskih maskah za medijski portal.

1. Ali je jasno, kaj prva stran sporoča? Vam je stran všeč? Kaj vam je na njej všeč? Kaj vam na njej ni všeč?
2. Ali je prikaz sentimenta jasen? Kaj pomenijo barve? Bi znali opisati lestvico čustev?
3. Kaj pričakujete, da se zgodi če kliknete na:
   3.1. graf sentimenta
   3.2. novico
   3.3. besedo v oblaku besed
   3.4. orodno vrstico na vrhu, na primer na besedo mladi
   3.5. gumb napredna analiza
4. Kaj pomeni barva novice?
5. Kakšen tip novic želite videti, ko prvič pridete na to stran? Kaj pa ko pridete na to stran ponovno?
6. Katere skupine predstavljajo poimenovanja v orodni vrstici na vrhu? Opišite skupine. So take skupine smiselne?
7. Kaj bi vas najbolj zanimalo pri obisku takega portala?

Merjenje nadmorske višine gladine jezer iz optičnih satelitskih slik

Domen Mongus[1], Matej Brumen[1], Borut Kozan[2]
[1]UM FERI, Koroška cesta 46, 2000 Maribor
[2]PETROL d.d., Dunajska 50, 1000 Ljubljana
domen.mongus@um.si, matej.brumen@um.si, borut.kozan@petrol.si

Izvleček

V tem članku se osredotočamo na implementacijo naprednega geografskega informacijskega sistema, ki omogoča samodejno ocenjevanje večletnih profilov gibanja nadmorskih višin gladine jezer. Implementacija predlaganega sistema temelji na integraciji platforme Open Access Hub, ki omogoča dostop do odprtih podatkov Copernicus. Pri tem se osredotočamo na multispektralne optične slike para satelitov Sentinel 2, ki omogočajo izvedbo meritev s petdnevno časovno in desetmetrsko prostorsko ločljivostjo. Ključni gradniki predlaganega sistema so podsistem za samodejen zajem podatkov ter njihovo predobdelavo, vključno s tehnikami podatkovnega razšumljanja, obrezovanje slik in njihovo prostorsko poravnavo, digitalni model reliefa oziroma točkovna definicija nadmorskih višin okoliškega terena, algoritem za razpoznavo in filtriranje oblačnih slik, komponenta za razpoznavo pokrivnosti površja, izračun vodnega indeksa in segmentacijo gladine jezer ter sistem za preslikavo razpoznanih mej gladine jezer v nadmorske višine. Z rezultati demonstriramo, da predlagan pristop omogoča dovolj visoko natančnost za praktično uporabo.

Ključne besede: Razpoznava vodnih območij, satelitske slike, program Copernicus, ansambelske metode za razpoznavo objektov

Abstract

*I*n this paper, we focus on the implementation of an advanced geographic information system that enables the automatic extraction of annual levels of lake surface elevations. The implementation of the proposed system is based on the integration of the Open Access Hub platform which allows access to Copernicus open data, specifically the multispectral optical images of a pair of Sentinel 2 satellites. These make possible measurements with a five-day temporal and ten-meter spatial resolutions. The key building blocks of the proposed system are the subsystem for automatic data harvesting and preprocessing comprised of data decomposition techniques, image cropping and spatial alignment, digital terrain model or point definition of the altitudes of the terrain surrounding the monitored lake, cloud image recognition and a filtering algorithm, a land cover recognition algorithm, water index calculations and lake level segmentation, together with a system for mapping the recognized lake surface boundaries to altitudes. The results demonstrate that the proposed approach achieves sufficient precision for practical use.

Keywords: Water recognition, satellite imaging, Copernicus programme, ensemble method for object recognition.

1 UVOD

V zadnjem desetletju smo priča silovitemu napredku optičnih tehnologij daljinskega zaznavanja. Te omogočajo podrobno opazovanje površja Zemlje ter odkrivanje vzorcev in karakteristik na velikih geografskih področjih. Napredni satelitski sistemi danes omogočajo sistematično spremljanje večjih prostorskih entitet in oceno njihovih karakteristik za namene podatkovno podprtega odločanja. V tem kontekstu so še posebej pomembni optični sateliti, ki so prilagojeni za sistematično spremljanje pokrovnosti tal [1]. Pomemben korak v tej smeri je nedavno naredila Evropska vesoljska agencija (ESA) z misijo Sentinel 2 v okviru programa Copernicus. Sentinel 2 je multispektrali optični satelit, ki omogoča spremljanje zemlje s srednjo prostorsko (10 – 16m) in visoko časovno (5 dni) ločljivostjo [2], kar je zagotovilo potrebno kontinuiteto podatkov za skoraj realno-časovno podporo odločanju [3]. Vseeno pa tudi podatki Sentinel 2 zahtevajo implementacijo naprednih tehnik umetne inteligence in strojnega učenja za izdelavo uporabnih podatkovnih produktov v vsakodnevni praksi.

V kontekstu multi- (ali hiper-) spektralnih podatkov se ključne informacije o opazovanih objektih skrivajo v tako imenovanih spektralnih podpisih. Slednji predstavljajo intenziteto odbite svetlobe na nivoju posameznega piksla glede na specifično valovno dolžnino elektromagnetnega sevanja. Zaradi specifičnih absorbcijskih lastnosti merjenih kemijskih elementov takšen pristop omogoča izjemno natančno razpoznavo materialov in oceno pokrovnosti zemeljskega površja. Vseeno pa so optični merilniki na satelitu pasivni in so tako zmožni meriti zgolj energijo elektromagnetnega sevanja, ki je večinoma posledica odboja sončne svetlobe. Posledično pa so tovrstne meritve neizogibno podvržene nenehno spreminjajočim se optičnim pogojem, ki so odvisni od atmosferskih razmer, vlažnosti tal, višine sonca, kota pogleda in fenoloških učinkov na površju Zemlje [4]. Tovrstna časovnoprostorska spremenjivost povzroča prekrivanje sketralnih odzivov, kar bistveno poslabša našo zmožnost razpoznave pokrovnosti. Posebno zahteven primer tega so sence, ki zaradi pomanjkanja direktne sončne svetlobe povzročijo pomembno izgubo spektralnih lastnosti merjenih objektov. Ker je v sencah, zaradi odbojnosti neba, pogosto prisotna zgolj šibka modra svetloba, spektralni podpis senčenih objetov spominja na spektralne podpise vode. Razločevanje vodnih območij od ostalih pokrovnih tipov pa zato zahteva predobdelavo podatko (glej sliko 1).

Ker je količina skladiščene vodne v jezereh pomemben faktor, tako za številne gospodarske panoge (na primer načrtovanje namakalnih programov v kmetijstvu, načrtovanje proizvodnje hidroenergije ter načrtovanje vodnega prometa [5,6]), kakor tudi okoljske študije [7,8], je bilo na to temo v preteklosti izvedenih več raziskav. Običajen pristop k izboljševanju razpoznave vodnih območij, ki ga uporablja večina sorodnega dela, naslavlja problematiko z uporabo tako imenovanih radimetričnih popravkov [9,10]. Te običajno izvedejo predhodno zaznavo senčnih območij, znotraj katerih izvedejo rekonstrukcijo spektralnih lastnosti. Takšen pristop pa zgolj omili izpostavljeno težavo, saj še vedno ne omogoča natančnega razločevanja vode znotraj senc. Nedavno smo predstavili izboljšano splošno metodo razpoznave pokrovnosti tal, ki namesto radimetričnih popravkov izvede segmentacijo učnih vzorcev glede na pravilnost razpoznave in tako izvede razpoznavo senc implicitno [11]. V tem članku predstavljamo nadgradnjo metode za dejansko merjenje nadmorske višine gladine jezer. V poglavju 2 podrobneje predstavimo postopek implicitne zaznave senčnih področji in razpoznavo vodnih območij. V poglavju 3 predstavimo visokonivojsko arhitekturo sistema in njegove ključne komponente. V poglavju 4 predstavimo rezultate svojega dela, medtem ko v poglavju 5 povzamemo ključne ugotovitve študije.

Slika 1: **Primer nerazločne meje med vodnimi in senčnimi področji na primeru Blejskega jezera.**

2. ANSAMBELSKI PRISTOP K RAZPOZNAVI POKROVNOSTI POVRŠJA

V tem poglavju podrobneje opišemo ansambelski pristop k razpoznavi pokrovnosti tal iz slik Sentinel 2, najprej predstavljen v [11], katerega prilagoditev predstavlja jedro uporabljene metode razpoznave vodnih površin. Za razliko od tradicionalnih pristopov k strojnemu učenju, predlagana metoda izvede segmentacijo učnih vzorcev glede na napake v razpoznavi osnovnega (šibkega) klasifikatorja. Konkretneje, metoda sestoji iz naslednjih korakov:

1. Izgradnja prostora značilnic, kjer izvedemo oceno značilnic, pri čemer se zanašamo na teksturno analizo in izračun spektralnih indeksov, ki jih strukturirano predstavimo.
2. Učenje klasifikatorja, pri čemer so avtorji metode pokazali, da najboljše rezultate dajeta naivni Bayesov klasifikator in naključni gozd,
3. Oceno natančnosti klasifikatorja, pri čemer izračunamo funkcijo gostote verjetnosti napak glede na vsako značilko,
4. Izvedemo segmentacijo učnih vzorcev z izbrano pragovno vrednostjo glede na minimizacijo entropije ter

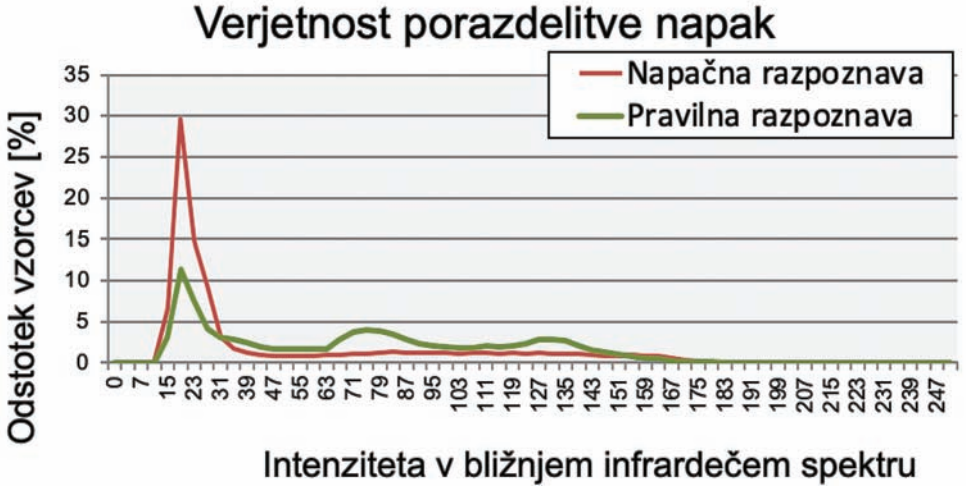
5. Ponovimo učenje z dvema klasifikatorjema, pri čemer se vsak uči nad svojim segmentom učne množice.

Zadnje tri korake ponavljamo, dokler ne dosežemo zaključnega kriterija, ki ga običajno določa število iteracij. Z drugimi besedami, predstavljeni postopek ustvari odločitveno drevo, ki omogoča kalibracijo klasifikatorjev na specifična prekrivanja spektralnih podpisov. Zaključni kriterij določa največjo dovoljeno globino drevesa in, posledično, število uporabljenih šibkih klasifikatorjev. Ker je uporaba osnovnih značilnic odvisna od domene uporabe, učenje klasifikatorjev pa dobro poznan postopek, v nadaljevanju tega poglavja podrobneje predstavimo zgolj koraka ocene natančnosti klasifikatorja in segmentacije učnih vzorcev.

2.1 Ocena natančnosti klasifikatorja

V tem poglavju predstavimo postopek ocenjevanja natančnosti osnovnega klasifikatorja, ki predstavlja izhodišče za segmentacijo učnih vzorcev in predstavitev novega šibkega klasifikatorja. Zaradi prekrivanja spektralnih podpisov šibki klasifikatorji namreč pogosto dajejo prevelik poudarek na specifične značilnosti, ki v splošnem omogočajo dobro razločevanje pomembnega deleža vseh vzorcev. Vseeno pa je očitno, da pri tem naredijo napake ravno v delu, kjer se spektralni podpisi prekrivajo. Slika 2 ilustrira to dejstvo na primeru razpoznave pokrovnosti Blejskega jezera in okolice iz slike 1, pri čemer je kot šibki klasifikator bil izbran naivni Bayesov klasifikator, učenje in analiza napak pa sta bila izvedena nad istim naborom osnovnih spektralnih podpisov multispektralnih slik Sentinel 2. Konkretneje, funkciji porazdelitve verjetnosti pravilno in napačno razpoznanih vzorcev, prikazani na sliki 2, sta izrisani glede na vrednosti vzorcev v bližnjem infrardečem spekteru (angl. near infrared, NIR), ki je posebej pomemben za razpoznavo vodnih območij (natančnejši opis uporabe spektra NIR podamo v poglavju 3). Pri tem pa je očitno, da je bila intenziteta skoraj tretjina vseh napačno razpoznanih vzorcev iz razpona [20, 24], znotraj razpona [20, 32] pa se nahaja več kot 50 % intenzitet vseh napačno razpoznanih vzorcev, ki so predvsem posledica spektralnega prekrivanje med vodnimi in senčnimi območji.

Ker lahko podobno obnašanje opazimo tudi pri uporabi drugih šibkih klasifikatorjev (na primer odločitvena drevesa ali naključni gozd), je v splošnem smotrno v osnovni klasifikacijski model vključiti nov šibki klasifikator, ki je posebej prilagojen na razpoznavo vzorcev iz območja največjega prekrivanja spektralnih podpisov (v tem primeru na razponu [0, 36]). Pri tem je pomembno tudi dejstvo, da se obnašanje šibkih klasifikatorjev ne razlikujejo bistveno, kadar takšno oceno napak izvedemo nad različnimi testnimi množicami. Slednje pa nam omogoča, da segmentacijo v fazi učenja izvedemo neposredno nad učnimi vzorci.



Slika 2: **Verjetnost porazdelitve napak šibkega klasifikatorja pri razpoznavi vodnih območij glede na intenzitete slike v spektru NIR.**

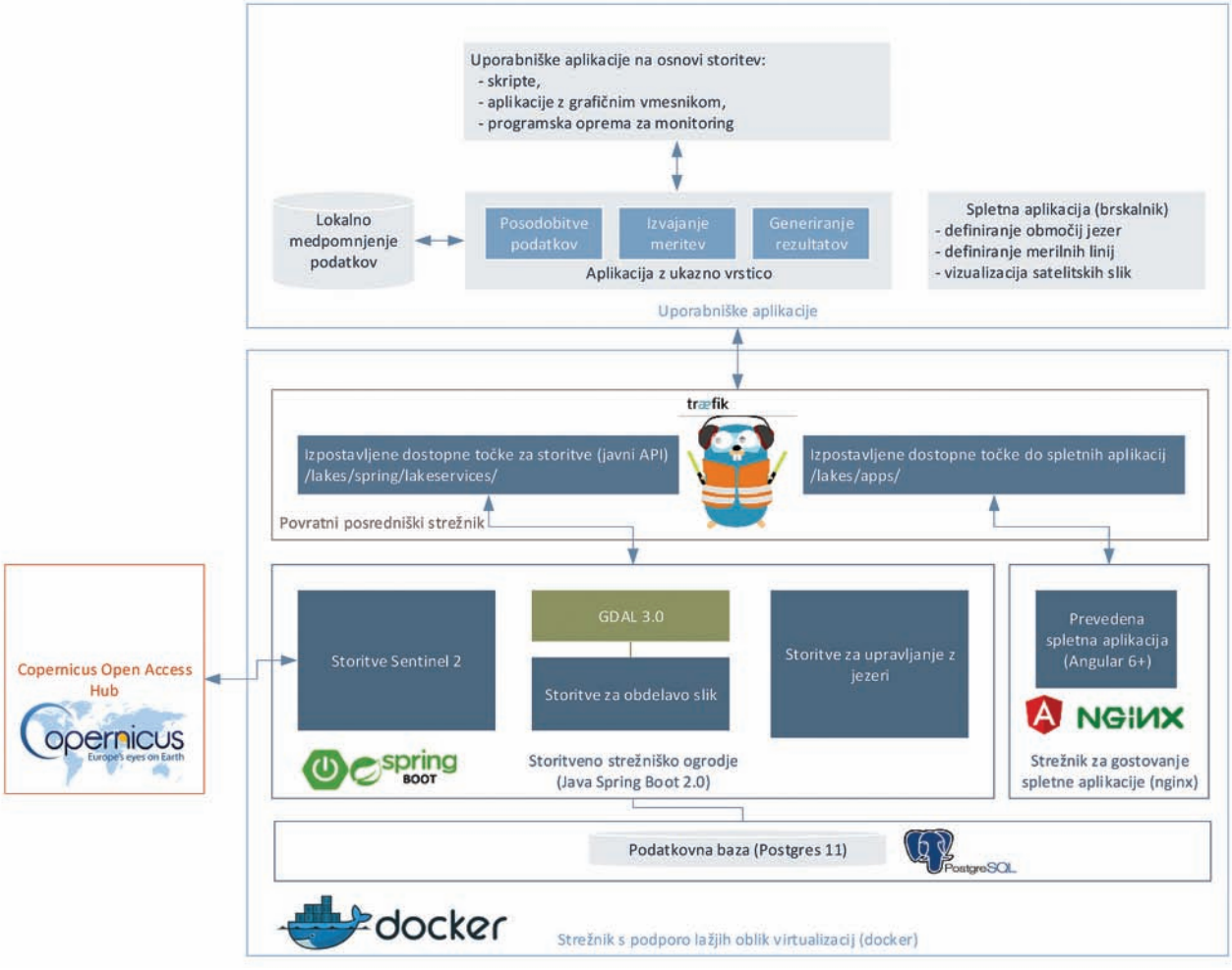
2.2 Segmentacija učnih vzorcev

Ocena natančnosti šibkih klasifikatorjev, predstavljena v prejšnjem poglavju, omogoča preprosto predstavitev novega klasifikatorja, ki ga prilagodimo za obravnavo vzorcev s prekrivajočimi spektralnimi podpisi. Ker učenje šibkega klasifikatorja vodi v nenatančnosti tako znotraj območja prekrivajočih spektralnih podpisov, kakor tudi zunaj njega, je smiselno prvotni šibki klasifikator zavreči in namesto njega predstaviti dva nova. Iz ocene natančnosti izhaja, da je razdelitev vzorcev najbolj naravno izvesti po principu minimizacije entropije med pravilno in nepravilno razpoznanimi vzorci, podobno kot to izvedemo v primeru izgradnje odločitvenega drevesa. Z drugimi besedami, segmentacijo učnega vzorca izvedemo z iskanjem optimalne pragovne vrednosti v naboru danih značilk, s katero dosežemo največji relativni informacijski prispevek (angl. information gain).

Medtem ko je matematična podlaga za izračun podana v [11], segmentacijo učne množice izvedemo v naslednjih korakih:

1. **Inicializacija klasifikacijskega modela**, pri čemer izvedemo učenje osnovnega šibkega klasifikatorja, ki predstavlja edini list drevesa.
2. **Ocena natančnosti klasifikacijskega modela**, kjer izračunamo verjetnosti porazdelitve napak glede na vsako izmed vhodnih značilk, kot je to predstavljeno v prejšnjem poglavju.
3. **Iterativna delitev učnih vzorcev,** kjer izvedemo:

a. iskanje optimalnega delitvenega kriterija in pripadajočo značilko
b. delitev učnih vzorcev glede na delitveni kriterij
c. učenje dveh novih šibkih klasifikatorjev ter
d. zamenjava lista drevesa z odločitvenim kriterijem in dvema pripadajočima šibkima klasifikatorjema v novih listih (poddrevesa).



Slika 3: **Visokonivojska arhitektura sistema za samodejno merjenje nadmorske višine gladine jezer.**

4. **Izhod metode** je tako klasifikacijski model, ki hierarhično organizira odločitvene kriterije za izbor pripadajočega šibkega klasifikatorja, s katerim se bo izvedla razpoznava testnega vzorca.

3. SISTEM ZA SAMODEJNO MERJENJE NADMORSKE VIŠINE GLADINE JEZER

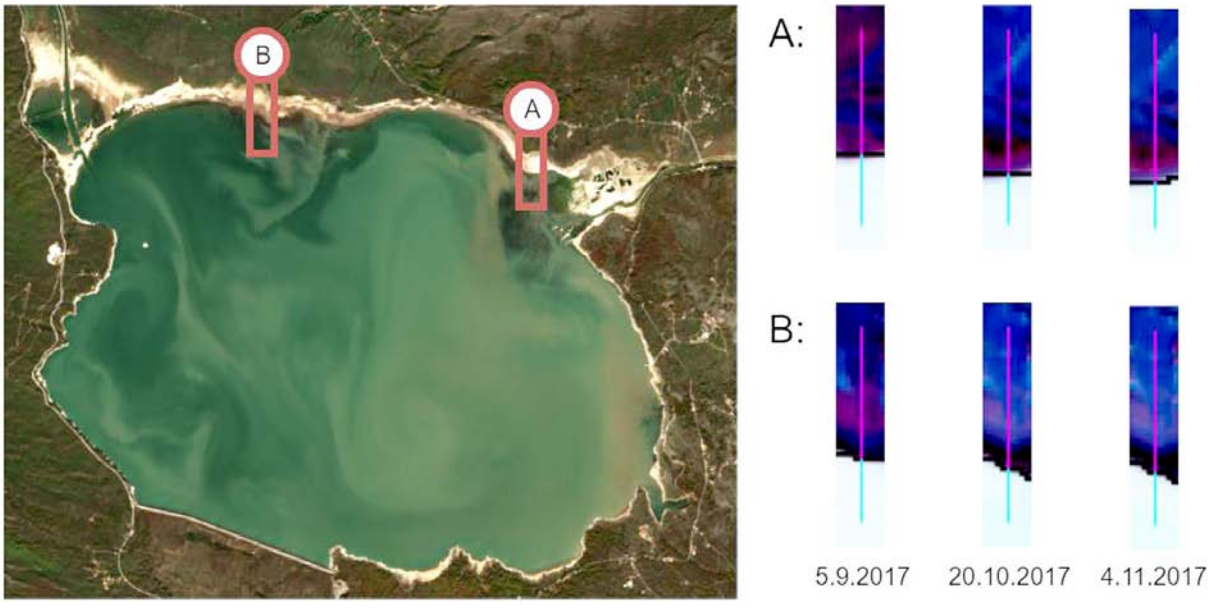
Implementacija predlaganega sistema temelji na integraciji platforme Open Access Hub, ki omogoča dostop do odprtih podatkov Copernicus. Njeni ključni gradniki so:

- kontrolne uporabniške aplikacije, ki omogočajo posodobitev podatkov, izvajanje meritev in generiranje rezultatov ter definicije merjenih jezer, parametrov meritev in njihovo vizualno preverjanje skozi namenski GIS,
- spletne storitve, ki omogočajo razpoznavo obale jezer ter preslikavo razpoznanih mej gladine jezer v nadmorske višine in
- sistem za zbiranje podatkov s podporno podatkovno bazo, ki vključuje vse potrebne parametre za samodejno delovanje.

Visokonivojsko arhitekturo sistema prikazuje slika 3, v nadaljevanju tega poglavja pa podrobneje opišemo storitveni del sistema, ki predstavlja jedro sistema.

Ključna značilka, na osnovi katere izvajamo razpoznavo vode, je tako imenovani normalizirani indeks vodne razlike. Tega izračunamo kot razmerje med razliko v intenziteti spektra NIR in kratkovalovnega infrardečega spektra ter seštevkom njunih intenzitet. Na sliki 4 so prikazana merilna območja nadmorske gladine jezera in izračunani normalizirani indeks vodne razlike v treh primerih iz obdobja med 5. 9. 2017 in 4. 11. 2017. Pri tem je pomembno poudariti dejstvo, da razpoznave meje gladine jezera ne izvajamo nad njegovo celotno površino, saj je slednje lahko preveč podvrženo šumu. Namesto tega raje določimo specifična merilna območja, kjer je obala dovolj plitva, da majhne spremembe v zaznavi roba ne povzročijo prevelike napake v izmeri nadmorske višine. Vseeno pa imamo teh merilnih mest več, saj tako zvišamo odpornost na prisotnost oblakov (glej sliko 4) in drug morebitni šum v podatkih.

Tudi razpoznave vodnih območij pri tem ne izvajamo nad celotnim merilnim območje, pač pa zgolj nad premico, ki ji pripišemo višinske vrednosti iz digitalnega modela reliefa. V primeru, da za merjeno območje takšnega modela nimamo, lahko tako preprosto določimo zgolj dve kontrolni višinski vrednosti in med njima izvedemo linearno interpolacijo višin. Slika 4b prikazuje tako definirane meritve in razpoznano vodo območje (svetlomoder del črte) ter teren (rdeč del črte).



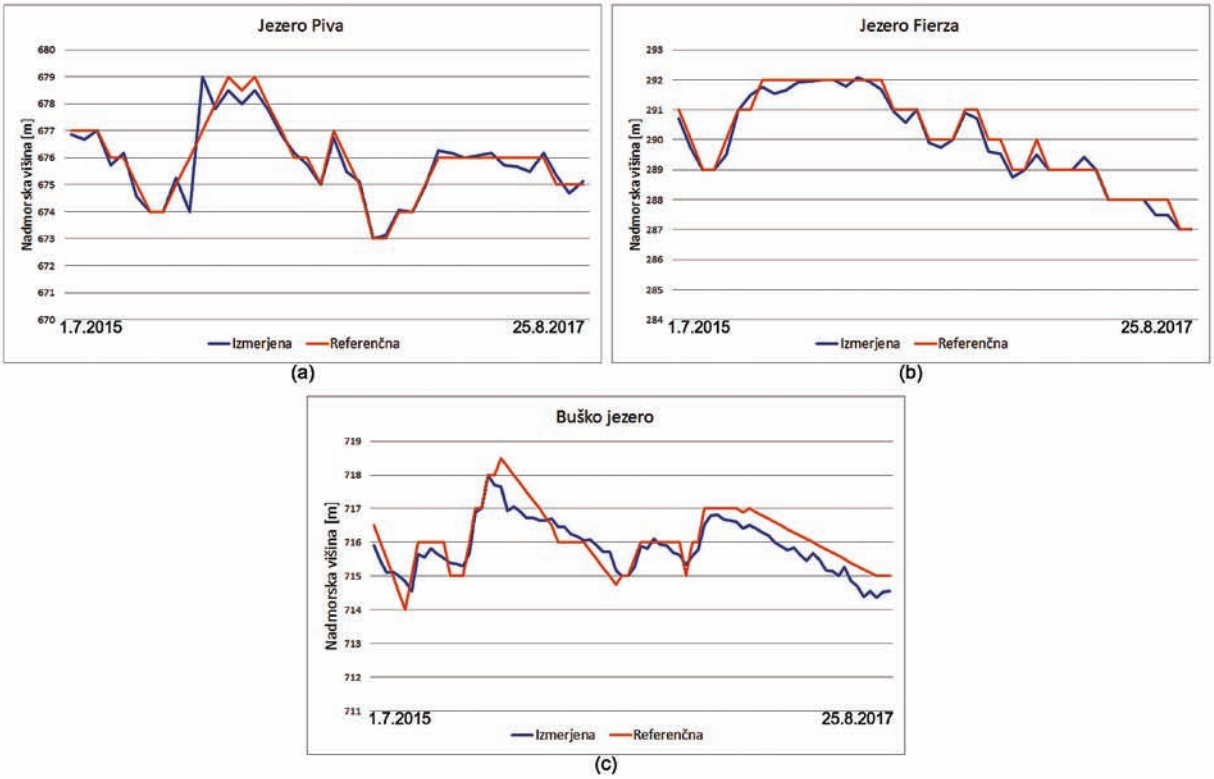
Slika 4: **Definicija meritve nadmorske višine gladine jezer, kjer (a) prikazuje merilna območja in (b) izračunane normalizirane indekse vodnih razlik z razpoznano mejo gladine (moder del črte).**

4. REZULTATI

Validacijo predlagane infrastrukture smo izvedli nad tremi jezeri, in sicer jezero Piva v Makedoniji, jezero Fierza v Albaniji ter Buško jezero v Bosni in Hercegovini. Stanje slednjega ob visokem in nizkem vodostaju je prikazano na sliki 5.

Slika 5: **Buško jezero ob (a) visokem in (b) nizkem vodostaju.**

Slika 6 prikazuje letne profile vseh merjenih jezer v času od 1.7.2015 do 25.8.2017. Zaradi neobstoječih zgodovinskih podatkov pravilnosti izmerjenih rezultatov ni mogoče preveriti absolutno, saj zgodovinske meritve za merjena jezera ne obstajajo. Kontrolo so zato izvedli strokovnjaki v podjetju Petrol d.d., kjer je model tudi v uporabi. Kontrolne meritve nadmorskih višin so bile pridobljene z ročnim pregledom slik, pri določitvi nivojev vode pa so bili uporabljeni tudi kontekstualnimi podatki, kot so to na primer količine padavin, odtek vode in dnevne temperature. Rezultati so prikazani na sliki 6.



Slika 6: **Dvoletni profili gibanja nadmorskih višin gladine jezer (a) Piva, (b) Fierza in (c) Buškega jezera.**

Izvedena ekspertna analiza je pokazala, da so izmerjeni podatki dovolj natančni in zanesljivi za praktično uporabo. Največja napaka je bila izmerjena na primeru Buškega jezera in je znašala 1.5m, medtem ko se je koren povprečnega kvadrata napake gibal med 0.26m (v primeru jezera Frieza) in 0.51m (v primeru jezera Piva). Podroben vizualni pregled rezultatov je pokazal, da se ključni razlogi za napake skrivajo v naklonu terena, saj zaradi razmeroma nizke ločljivosti slik Sentinel 2 (10 m in več) to hitro vodi v razmeroma velike višinske razlike. Tako so bile največje napake zaznane, ko so severni (položni) del Buškega jezera prekrivali oblaki in meritve tam niso bile mogoče. Podobno velja tudi za primere večjih napak v oceni nadmorske višine gladine jezera Pliva, ko so bile meritve omejene na njegovo vzhodno stran. Zaradi oblačnega vremena pa je sicer v splošnem bilo neuporabnih tudi do 25 % vseh posnetkov posameznega jezera.

Ker je za številne aplikacije v podporo načrtovanja (na primer namakanja ali vodnega prometa) pogosto pomembneje spremljanje sprememb nivojev vode, smo nad pridobljenimi podatki ocenili tudi Pearsonov korelacijski koeficient. Slednji je bil zaradi že omenjene strmine naklona vzhodne, južne in zahodne obale najnižji v primeru Buškega jezera (0.89), medtem ko je v primeru jezer Piva in Frieza znašal 0.93 in 0.98.

5. ZAKLJUČEK

V članku smo predstavili sistem za samodejno merjenje nadmorske višine gladine jezer za podporo pri odločanju. Predlagan sistem temelji na segmentaciji ključnih učnih vzorcev za izboljšano klasifikacijo v primerih prekrivanja spektralnih podpisov, kot je to običajen primer med vodo in senčnimi območji. Merjenje meje vodne gladine omogoča preračun nadmorske višine gladine jezera glede na digitalni model reliefa, pri čemer odpornost metode na oblake bistveno izboljšamo z večkratnim vzorčenjem vzdolž plitvih obal. Z rezultati smo pokazali tudi, da je predlagana metoda primerna za praktično uporabo, še zlasti v kontekstu ocenjevanja večletnih profilov gibanja gladine jezer.

LITERATURA

[1] CHENG, Gong, HAN, Junwei: A survey on object detection in optical remote sensing images. ISPRS Journal of Photogrammetry and Remote Sensing, 2016, št. 117, str. 11-28.

[2] DRUSCH, Matthias in sodelavci: Sentinel-2: ESA's optical high-resolution mission for GMES operational services, Remote sensing of Environment, 2012, št. 120, str. 25-36.

[3] VAN DER MEER, Freek in sodelavci: Potential of ESA's Sentinel-2 for geological applications, Remote sensing of environment, 2014, št. 148, str. 124-133.

[4] GOMEZ, Cristina, WITHE, Joanne, WULDER, Michael: Optical remotely sensed time series data for land cover classification: A review, ISPRS Journal of Photogrammetry and Remote Sensing, 2016, št. 116, str. 55-72.

[5] COOPS, Hugo, BEKLIOGLU, Meryem, CRISMAN, Thomas L: The role of water-level fluctuations in shallow lake ecosystems–workshop conclusions, Hydrobiologia, 2003, št 506 (1-3), str. 23-27.

[6] ZHANG, Jiqun, XU, Kaiqin, YANG, Yonghui, QI, Lianhui, HAYASHI, Seiji, WATANABE, Masataka: Measuring water storage fluctuations in Lake Dongting, China, by Topex/Poseidon satellite altimetry, Environmental Monitoring and Assessment, 2006, št. 115 (1-3), str. 23-37.

[7] JAWAK, Shridhar, KULKARNI, Kamana, LUIS Alvarinho: A review on extraction of lakes from remotely sensed optical satellite data with a special focus on cryospheric lakes, Advances in Remote Sensing, 2015, št. 4(3), str. 196.

[8] SONG, Chunqiao, HUANG, Bo, KE, Linghong: Inter- annual changes of alpine inland lake water storage on the Tibetan Plateau: Detection and analysis by integrating satellite altimetry and optical imagery, Hydrological Processes, 2014, št. 28(4), str. 2411-2418.

[9] KASETKASEM, Teerasit, VARSHEY, Pramod: An optimum land cover mapping algorithm in the presence of shadows, IEEE Journal of Selected Topics in Signal Processing, 2011, št. 5(3), str. 592-605.

[10] MOVIA, Alessia, BEINAT Alberto Beinat, CROSILLA Fabio: Shadow detection and removal in RGB VHR images for land use unsupervised classification, ISPRS Journal of Photogrammetry and Remote Sensing, 2016, št. 119, str. 485-495.

[11] MONGUS, Domen, ŽALIK, Borut: Segmentation schema for enhancing land cover identification: A case study using Sentinel 2 data, International journal of applied earth observation and geoinformation, 2018, št. 66, str. 56-68.

▪

Domen Mongus je izredni profesor na Univerzi v Mariboru in član izvršnega odbora mednarodneha združenja GISIG (anlg. Geographical Information System International Group). V preteklosti je bil podpredsednik programskega odbora nacionalnega Strateškega razvojno inovacijskega partnerstva v okviru slovenske strategije pametne specializacije (S4) na področju Pametnih mest in skupnosti in član izvršnega odbora krovne evropske organizacije za geografske informacije EUROGI. Njegovi raziskovalni interesi vključujejo obdelavo podatkov daljinskega zaznavanja, prostorsko-časovno analitiko in geoprostorsko inteligenco. Za svoje dosežke je prejel več nacionalnih in mednarodnih nagrad, med drugim je bil leta 2015 imenovan za mladega znanstvenika podonavske regije, leta 2018 pa je prejel najprestižnejšo institucionalno akademsko nagrado za izjemen prispevek k znanstvenemu in pedagoškemu ugledu ter odličnosti Univerze v Mariboru.

▪

Matej Brumen je vodja razvojne ekipe v laboratoriju za Geometrijsko Modeliranje in Algoritme Multimedije na Fakulteti za Elektrotehniko Računalništvo in Informatiko. Njegovo področje obsega vse od obdelave prostorskih podatkov in 3D vizualizacij do geografsko informacijskih sistemov in mikrostoritev ter razvoj končnih rešitev v sodelovanju številnimi slovenskimi in mednarodnimi industrijskimi partnerji.

▪

Borut Kozan je vodja oddelka razvoja energentov v podjetju Petrol, kjer se ukvarja z razvojem področja napredne analitike, poslovnih modelov in procesov v segmentu prodaje in trgovanja z električno energijo in zemeljskim plinom. V času študija na Fakulteti za elektrotehniko (UL) je njegovo raziskovalno delo obsegalo modeliranje trga električne energije s poudarkom na agentnem učenju, modeliranje Evropske sheme trgovanja z emisijami (EU ETS), ekonomike investicij v energetskih sistemih ter modeliranje naključnih procesov. Del doktorskega raziskovanja je opravil na Universidad Pontificia Comillas v okviru Institute for Research in Technology.

Izzivi integracije zdravstvenih aplikacij: suporaba standardov OpenEHR in FHIR

*Marina Trkman[a], Mitja Lapajne[b], Božidarka Radović[b]
[a]Institut Jožef Stefan, Jamova cesta 39, 1000 Ljubljana
[b]Better, Koprska ulica 100, 1000 Ljubljana
*marina.trkman@e5.ijs.si

Izvleček

Veliko zdravstvenih aplikacij mora komunicirati med seboj. Prepisovanje podatkov iz ene aplikacije v drugo je nedopustno, saj se pri tem dogajajo napake, ki ogrožajo paciente. Ker želimo omogočiti avtomatski prenos podatkov iz ene v drugo aplikacijo, se je potrebno osredotočiti na zagotavljanje sintaktične in semantične interoperabilnosti. V članku prestavljamo primer integracije dveh zdravstvenih aplikacij v Angliji. Angleški NHS pri prenosu podatkov narekuje uporabo podatkovnih elementov standarda FHIR, s čimer dosežemo sintaktično interoperabilnost. Da bi dosegli semantično interoperabilnost, je potrebno zagotoviti, da se pomen konteksta pri prenosu ne spremeni. Cilja našega članka sta dva. Prvi je predstaviti predlog preslikave podatkovnih elementov iz standarda openEHR v integracijski standard imenovan FHIR. Drugi je ugotoviti, s katerimi izzivi se soočajo programerji ob uporabi standarda FHIR. V članku smo predstavili, kako angleška skupnost INTEROPen promovira uporabo razširjenega standarda CareConnect FHIR v angleškem zdravstvenem sistemu. Ugotovili smo, da temeljni izziv predstavlja prepisovanje podatkov iz enega standarda v drugega, saj ogroža pacientovo varnost, ker se lahko zgodijo napake pri prepisovanju. Tudi raznolika interpretacija pomena podatkovnih elementov objektov FHIR predstavlja izziv. In na zadnje tudi izdaje novih različic standarda FHIR, CareConnect FHIR in eksternih šifrantov. Ponudniki aplikacij, ki so povezane med seboj, morajo sodelovati, da sočasno umestijo nove spremembe v integriranih aplikacijah oziroma predvidijo morebitne napake in strategije za reševanje težav.

Ključne besede: odprti standard, Fast Healthcare Interoperability Resources, openEHR, interoperabilnost, zdravstvo

Abstract

Many healthcare applications need to communicate patient data. Transcribing data from one application to another is not acceptable since it may result in medication errors which can endanger the lives of patients. The data needs to be transferred from one application to another automatically. In order to integrate applications, syntactic and semantic interoperability is crucial. We present an integration of two healthcare applications in an English hospital. The NHS in England has a requirement that healthcare applications must be integrated with a standard called FHIR. The use of the standard ensures syntactic interoperability. In order to achieve semantic interoperability, the translation from openEHR to FHIR must be performed in such a way that the context of data is preserved. Our paper has two objectives. The first is the transformation of data elements (fields) from the openEHR to the FHIR standard. The second is the identification of challenges that healthcare providers need to be prepared for when using FHIR. We presented an English community called INTEROPen that manages the use of the CareConnect FHIR extended standard for integrating healthcare applications in England. Furthermore, we identified practical challenges of using CareConnect FHIR. The crucial challenge is related to transcribing patient data from the openEHR standard to the CareConnect FHIR standard. Such transcriptions can compromise a patient's safety due to potential transcription errors. Another challenge is the interpretation of FHIR fields, which is possible in several ways. Also, new releases of FHIR, CareConnect FHIR and external dictionaries pose a challenge since they require the prompt and careful overhaul of applications. Vendors of healthcare application need to collaborate in order to implement changes in integrated applications in a synchronized manner, to foresee the potential errors and, consequently, prepare strategies to mitigate them.

Keywords: Open standard, Fast Healthcare Interoperability Resources, openEHR, interoperability, healthcare..

1 UVOD

Zdravstveni informacijski sistem zajema številne sisteme, kot so administrativne aplikacije, medicinske naprave, nadzorne aplikacije, radiološke elektronske slike ter aplikacije za upravljanje in predpisovanje zdravil (Board on health care services, 2004; Schleyer, Rahurkar, & Schaffer, 2019). Ti sistemi pogosto niso povezani med seboj (Shahmoradi, Habibi-Koolaee, Ebrahimi, Khoy, & Soltani, 2017). Posledično so lahko pacientovi podatki na neki zdravstveni ustanovi (na primer bolnišnici) zastareli in nepopolni.

Integracija različnih zdravstvenih informacijskih sistemov izboljša dostop do kontekstno občutljivih informacij. Prav tako omogoča izboljšave toka dela ter večjo varnost in vizualizacijo podatkov (Meyer et al., 2005). Prednosti integracije občutijo tako delavci v zdravstvu kot tudi pacienti. Povezani informacijski sistemi omogočajo takojšnji dostop do celostnih podatkov o zdravju in zdravljenju pacienta zdravstvenim delavcem iz različnih zdravstvenih ustanov. Zdravstvenim delavcem ni več treba prepisovati podatkov, kar znatno zmanjšuje verjetnost napake pri prepisovanju. Pacientom pa ni treba skrbeti, da bi natisnjene izvide izgubili.

Ko želimo povezati dva informacijska sistema, je treba poskrbeti za interoperabilnost. Aplikacija je interoperabilna z drugo aplikacijo, če lahko uporablja storitve in/ali podatke te druge aplikacije (Parv, Kruus, Motte, & Ross, 2016). Tipično komunikacija med aplikacijami poteka s pomočjo vmesne programske opreme (angl. middleware). Poznamo več tipov vmesnikov (Shahmoradi et al., 2017). Sporočilno-orientiran vmesnik (angl. message-oriented middleware) se uporablja za integriranje zdravstvenih aplikacij na regionalni ravni (Radović, 2019), saj omogoča avtomatizacijo procesov, kot so predpisovanje in naročanje zdravil ter naročanje in sprejem pacientov. Pri tovrstnem vmesniku se podatki prenašajo s pomočjo podatkovnih baz in programskih vmesnikov (angl. Application Programming Interfaces, API). API-ji morajo imeti strukturirano urejene pacientove podatke (Mandel, Kreda, Mandl, Kohane, & Ramoni, 2016).

Namen članka je predstaviti sintaktično in semantično interoperabilnost dveh aplikacij različnih ponudnikov z uporabo standarda FHIR (angl. Fast Healthcare Interoperability Resources). Nivo sintaktične interoperabilnosti je dosežen z uporabo odprtokodnih standardov, ki definirajo strukturo meta podatkov za izmenjevanje pacientovih podatkov. Uporaba standarda še ne zagotavlja semantične interoperabilnosti. Semantična interoperabilnost poudarja pomembnost ohranitve pomena podatkovpri prenosi. V praksi to pomeni, da imata integrirani aplikaciji enako opredeljene podatkovne elemente standarda FHIR. To pa ni lahko dosegljivo. Cilja našega članka sta tako dva. Prvi je predstaviti predlog preslikave podatkovnih elementov iz standarda openEHR v integracijski standard imenovan FHIR. Drugi pa ugotoviti, s katerimi izzivi se soočajo programerji ob uporabi standarda FHIR.

V nadaljevanju v drugem poglavju predstavljamo standard za shranjevanje strukturiranih podatkov o pacientih openEHR. Nato v tretjem poglavju predstavimo integracijski standard FHIR, ki se uporablja za prenos podatkov iz ene aplikacije v drugo. V četrtem poglavju predstavimo naš predlog zagotavljanja semantične interoperabilnosti: preslikave podatkovnih elementov iz openEHR v FHIR (in obratno). Študijo primera uporabe predloga preslikave predstavimo v petem poglavju. V šestem poglavju predstavimo izzive, ki jih nosi uporaba standarda FHIR. V sedmem poglavju smo podali zaključne misli.

2 STANDARD OPENEHR

Glavni namen openEHR standarda je shranjevanje strukturiranih podatkov o pacientu v obliki elektronskega zdravstvenega zapisa (angl. Electronic Health Record, EHR). Sestavljen je iz odprte specifikacije, domenskih modelov, programske opreme za oblikovanje standardov in programske opreme za zagotavljanje interoperabilnosti (openEHR Foundation, 2020).

OpenEHR Foundation združuje tako klinične kot tehnične strokovnjake, ki iščejo načine, kako učinkoviteje upravljati s podatki o pacientih. Prvi problem se nanaša na občutljivo naravo podatkov o pacientih, katerih osveževanje mora biti učinkovito in nadzorovano. Drugi problem predstavlja neenotna semantika zdravstvenih terminov. Nepravilno implementirane spremembe lahko vodijo do tega, da neka informacija o pacientu izgine ali postane nepravilna in s tem pacientu nevarna. Tretji problem je, da podatki nastajajo v mnogih zdravstvenih organizacijah in v različnih aplikacijah. Pacientovi podatki se morajo sproti osveževati ter biti ob pravem času na pravem mestu na voljo vsem upravičenim zdravstvenim strokovnjakom. Zato je potrebna primerna tehnologija za shranjevanje podatkov, ki nudi dostop

do njih mnogim aplikacijam, ki uporabljajo različno programsko in strojno opremo.

OpenEHR Foundation predlaga openEHR standard kot rešitev teh problemov. OpenEHR specifikacija vključuje informacijske modele zdravstvenih podatkov, ki odgovarjajo na naslednje vprašanja (openEHR Foundation, 2020):

- Kako shraniti klinične in demografske podatke pacientov?
- Kako poizvedovati po bazi, ne da bi poznali strukturo same baze?
- Kako poizvedovati po domenskih modelih?
- Kako uporabiti ISO standard za komunikacijo o vsebini domene in podatkih?
- Kakšna je struktura API-ja?

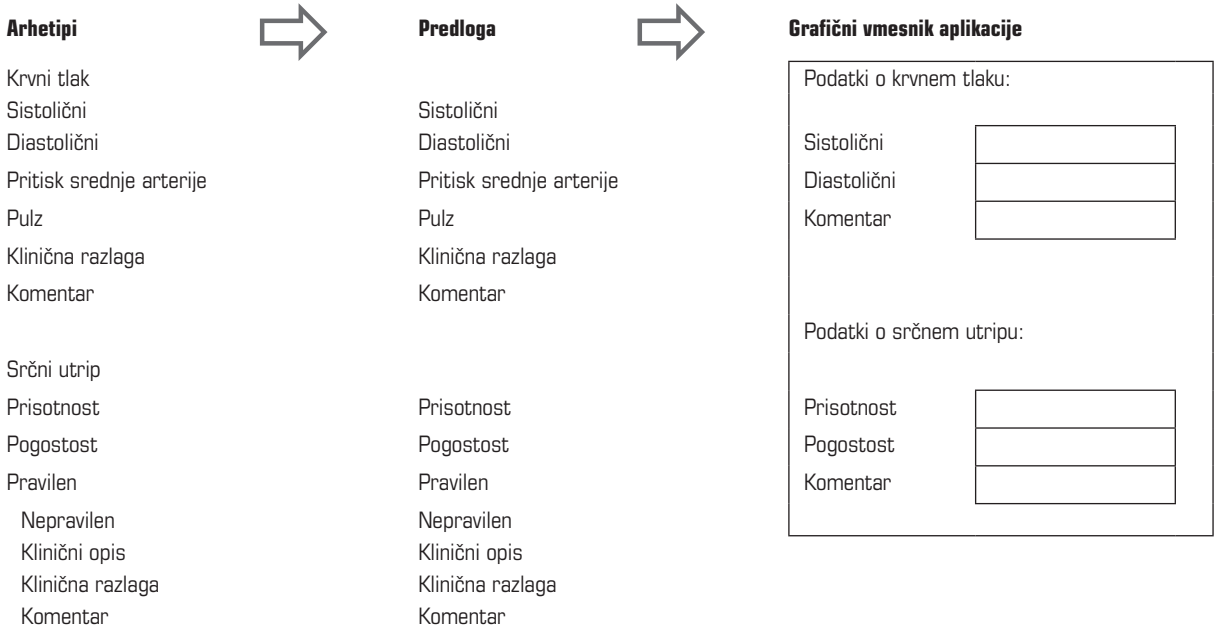
OpenEHR obsega uporabo konceptov in pravil. Najpomembnejše pravilo je, da so podatkovni (razvojni) modeli ločeni od domenskih (vsebinskih, kliničnih) modelov. Domenske modele izdela in potrdi skupnost kliničnega osebja, medtem ko podatkovne modele upravlja ponudnik programske opreme. Domenski modeli so referenčni modeli (angl. reference model), arhetipi (angl. archetypes) in predloge (ang. template). Referenčni model predstavlja sestavne dele podatkov, ki jih uporablja arhetip. Arhetip vsebuje maksimalen skupek podatkov o nekem konceptu, na primer krvnem tlaku. Predloga pa je izbrana skupina podatkov o tem konceptu za neki tip primera uporabe. Na Sliki 1 je primer uporabe modelov arhetipov in predlog za potrebe neke aplikacije. Drugo pravilo je, da je predstavitev podatkov (v neki aplikaciji) ločena od podatkovnega vira, ki je neodvisen od katerekoli aplikacije.

OpenEHR je odprti standard, ki se uporablja pri izgradnji odprte platforme (angl. open platform) za zdravstveni ekosistem (Apperta fundation, 2018). OpenEHR se uporablja za shranjevanje strukturiranih podatkov. Za shranjevanje nestrukturiranih podatkov, kot so slike, pa se uporabljajo standardi, kot so IHE-XDS (Apperta fundation, 2018), DICOM in PACS (Li, 2014).

Integracije med aplikacijami znotraj istega ekosistema so relativno enostavne, saj vse shranjujejo strukturirane podatke po istem standardu - openEHR. Integracija dveh aplikacij, kjer je ena nastala po standardu openEHR, druga pa ne, je zahtevnejša. Potreben je »prevajalec« in to vlogo prevzame standard FHIR (Apperta fundation, 2018).

3 STANDARD FHIR

FHIR specifikacija predstavlja standard za elektronsko izmenjavo zdravstvenih informacij (HL7, 2020), (Wagholikar et al., 2016). Mnogi verjamejo, da bo FHIR postal uveljavljen zdravstveni standard na področju prenosa pacientovih podatkov (Sarita, Dave, & Yunfeng, 2017), saj v praksi pridobiva vse večjo veljavo (Apperta fundation, 2018). Deluje tako, da konkretne pacientove podatke opremi s svojimi podat-



Slika 1: **Primer uporabe domenskih modelov openEHR za potrebe aplikacije (Radović, 2019)**

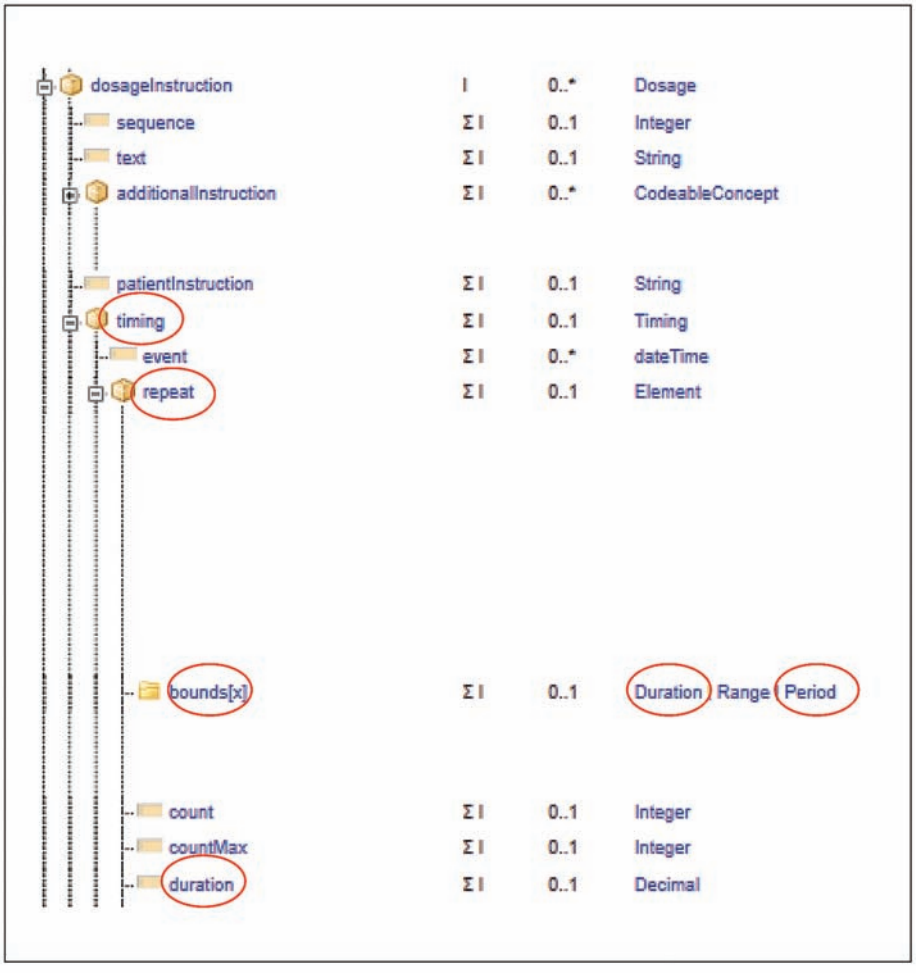
kovnimi elementi za prenos v drug sistem s pomočjo dokumenta XML ali JSON.

Osnova za delo s FHIR specifikacijo so objekti (angl. resources), ki so navedeni v FHIR-ovi knjižnici različice STU3. Objekt je lahko katera koli vsebina, ki se prenaša in ima naslednje značilnosti: meta podatke, svojo sestavo, ki temelji na podatkovnih tipih, in opis berljiv človeku. Če želimo poslati pacientove podatke, uporabimo FHIR-ov objekt imenovan Patient. Nabor podatkovnih elementov znotraj tega objekta določa, katere podatke o pacientu lahko pošljemo, na primer ime, priimek, številko zavarovanja in naslov. Če želimo poslati podatke o zdravilu, uporabimo FHIR-ov objekt, imenovan Medication. Za pošiljanje podatkov o dozi, načinu aplikacije, in frekvence zdravila uporabimo MedicationRequest.

Problem standarda FHIR je, da specifikacija premalo omejuje podatkovne elemente objektov (Mandel et al., 2016). Iz tega izhajajo pomisleki glede varnosti pacientovih podatkov, ki jih pošiljamo iz enega sistema v drugega (Franz, Schuler, & Krauss, 2015). V našem članku prispevamo k boljšemu razumevanju problematike varnosti pacientovih podatkov pri prenosu s standardom FHIR.

4 PRIMER: INTEGRIRANJE APLIKACIJ S FHIR

V angleški bolnišnici so uporabljali dva nepovezana informacijska sistema: aplikacijo za elektronsko predpisovanje zdravil OPENeP in lekarniško aplikacijo za izdajo zdravil Lekarna. Študijo primera integracije smo omejili na dogodek prenosa podatkov v lekarno o enem zdravilu, ki ga ima pacient ob odpustu. Ker obe aplikaciji delujeta v angleški bolnišnici, morata biti po navodilih NHS integrirana s standardom FHIR, ki ga podpira HL7 (Radović, 2019). Angleški NHS je mnenja, da je potrebno FHIR dopolniti tako, da bolje definira semantično in sintaktično integracijo zdravstvenih aplikacij.



Slika 2: **Izsek iz specifikacije za dosageInstruction v CareConnect FHIR (INTEROPen, 15.6.2020)**

4.1 Semantična in sintaktična integracija

V Angliji deluje skupnost INTEROPen, ki združuje ponudnike aplikacij in NHS Digital , da bi se dogovorili o podrobnostih integriranja. Ker so podatkovni elementi nekaterih FHIR-ovi objektov preširoko zastavljeni, je skupnost definirala njihove razširitve. Razširjeni objekti temeljijo na uporabi objektov iz FHIRove knjižnice različice STU3, ki je bila objavljena marca leta 2017. Razširjeni objekti skupnosti INTEROPen so zbrani v knjižnici HL7 UK INTEROPen CareConnect[1].

CareConnect je med drugim objavil razširjen objekt, ki omogoča integracijo aplikacije za elektronsko predpisovanje in lekarniške aplikacije. CareConnect-MedicationRequest-1 objekt se uporablja za prenos podatkov o naročilu enega zdravila za enega pacienta. Objekt predstavlja razširitev FHIRovega objekta MedicationRequest. Razširitev prinaša novosti pri strukturiranju navodil za doziranje in posledično uvaja nekatere nove (pod-)podatkovne elemente. Informativni izsek iz specifikacije o doziranju je na Sliki 2.

Razširjen objekt ima veliko podatkovnih elementov[2], vendar je samo manjši nabor redno v uporabi in še manjši nabor je obvezen. Obvezni podatkovni elementi za prenos so odvisni od dogovora med deležniki integracije. S podatkovnimi elementi zagotavljamo sintaktično interoperabilnost.

Aplikacija za predpisovanje zdravil OPENeP za shranjevanje kliničnih podatkov o pacientih uporablja standard openEHR. Ker ga lekarniški sistem ne uporablja, je potreben prepis podatkov v arhitekturo podatkovnih elementov, ki jih določa CareConnect FHIR. Standarda openEHR in FHIR imata različne podatkovne elemente. Pri prenosu podatkov pacienta iz enega sistema v drug je treba poskrbeti, da se informacija (torej kontekst, vsebina) ne popači. Za namene semantične interoperabilnosti smo v Tabeli 1 pripravili predlog preslikav podatkovnih elementov iz openEHR v FHIR na primeru naročila zdravila. Iz tabele je razvidno, da je ime zdravila Aspirin po standardu openEHR shranjeno v podatkovni element *medicationItem*, po standardu FHIR pa v *Medication*.

V XML dokumentu lahko uporabimo več objektov skupaj. Na primer, z uporabo objekta MedicationRequest in njegovega podatkovnega elementa *contained* se lahko navezujemo na objekt Medication. Na Sliki 3 predstavljamo XML dokument na primeru naročila 500 mg aspirina, ki naj ga pacient vzame oralno naslednjih 7 dni na 8 ur. Dokument je osredotočen na podatkovne elemente za doziranje zdravila.

Tabela 1: **Preslikava podatkov iz openEHR v CareConnect FHIR – zagotavljanje semantične interoperabilnosti (Radović, 2019)**

Element	Primer	OpenEHR	FHIR
Ime zdravila	Aspirin	MedicationItem	ResourceMedication
Doza	100	structuredDoseAndTimingDirections → dosage → doseAmount	dosageInstruction → dose: quantity → value
	Mg	structuredDoseAndTimingDirections → dosage → doseAmount	dosageInstruction → dose: quantity → value
Način aplikacije	Oralno	route	dosageInstruction → route
Pogostost doziranja	3 x na dan	structuredDoseAndTimingDirections → dosage → timing → frequency	dosageInstruction → timing →repeat → period dosageInstruction → timing →repeat → frequency dosageInstruction → timing →repeat → periodUnit
	Vsakih 8 ur	structuredDoseAndTimingDirections → dosage → timing → frequency	dosageInstruction → timing →repeat → frequency dosageInstruction → timing →repeat → periodUnit
Komentar	Bolečina	comment	Note → text
Indikacije	Vnetje sečil	clinicalindication	reasonCode → text

[1] Seznam CareConnect objketov je na voljo na povezavi https://fhir.hl7.org.uk/.
[2] Specifikacija objekta CareConnect MedicationRequest je na voljo na spletni povezavi https://fhir.hl7.org.uk/STU3/StructureDefinition/CareConnect-MedicationRequest-1.

```
<!--objekt MEDICATION -->
<Medication>
        <id value="med1" />
        <code>
                <coding>
                        <system value=http://snomed.info.sct />
                        <code value="7947003" />
                        <display value="product containing aspirin (medicinal product)" />
                </coding>
        </code>
</Medication>


<!-- objekt MEDICATION REQUEST -->
<medicatioRefernce>
        <reference value="#med1" />
</medicatioRefernce>

<!-- navodila za doziranje -->
<dosageInstruction>
<!-- 500 mg zdravila -->
        <doseQuantity>
                <value value="500.0" />
                <unit value="mg" />
                <system value="http://unitsofmeasure.org" />
                <code value="mg" />
        </doseQuantity>

<!—zdravilo jemlje 7 dni na vsakih 8 ur-->
        <timing>
                <repeat>
                        <boundsDuration>
                                <value value="7" />
                                <unit value="day" />
                                <system value="http://unitsofmeasure.org" />
                                <code value="d" />
                        </boundsDuration>

                        <frequency value="1" />
                        <period value="8" />
                        <periodUnit value="h" />
                </repeat>

        </timing>

</dosageInstruction>
```

Slika 3: **Primer uporabe podatkovnih elementov CareConnect FHIR**

5 IZZIVI INTEGRACIJE ZDRAVSTVENIH APLIKACIJ

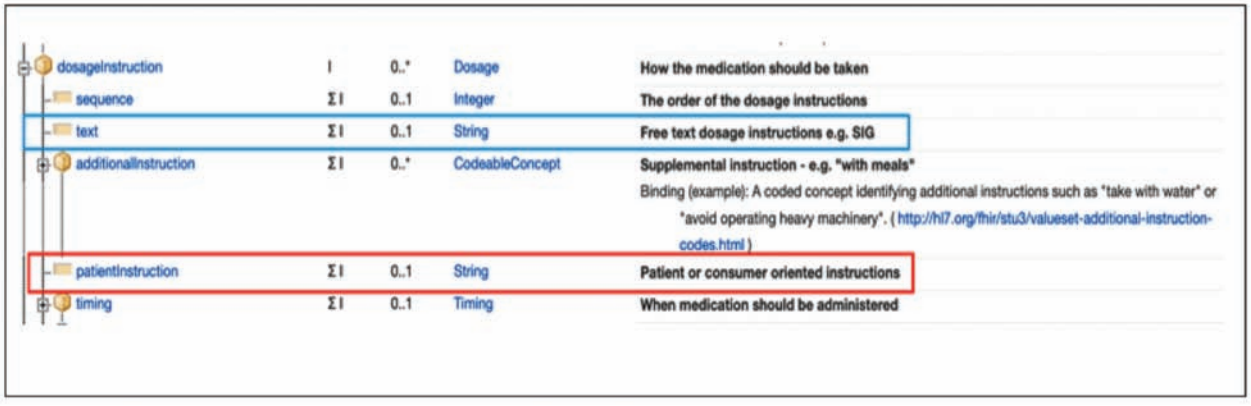
Integracija po standardu, kot je FHIR, reši problem skupnega jezika za izmenjavo podatkov, odpira pa nove izzive. V naši študiji primera smo prikazali primer semantične in sintaktične integracije med dvema zdravstvenima aplikacijama. Pri tem smo prišli do naslednjih izzivov integracije s FHIR-jem, ki jih morajo reševati razvojna podjetja.

Izziv 1: Prepisovanje pacientovih podatkov iz standarda za shranjevanje podatkov openEHR v standard za prenos podatkov FHIR ogroža pacientovo varnost.

Zdravstveni sistem je sestavljen iz mnogih specializiranih zdravstvenih aplikacij. Nekatere aplikacije se uporabljajo zaporedno. V našem primeru aplikacija za predpisovanje zdravil posreduje podatke v lekarniško aplikacijo. Smiselno je, da sta aplikaciji podatkovno povezani, saj je v nasprotnem primeru potrebno ročno prepisovanje podatkov. Aplikacija za predpisovanje zdravil je zgrajena po openEHR standardu, lekarniška aplikacija pa ne. Ker si želita izmenjevati podatke, potrebujeta integracijski standard CareConnect FHIR (NHS Digital, 2020). Praksa je pokazala, da ni dovolj, da vsak posamezen ponudnik aplikacije naredi preslikavo podatkovnih elementov iz svojega standarda v FHIR (in obratno). Za boljšo semantično integracijo je potrebno, da se ponudniki integriranih aplikacij dogovorijo na praktičnih primerih, kateri podatek je shranjen v katerem podatkovnem elementu standarda FHIR. Delovanje skupnosti INTEROPen v Angliji je primer tovrstnega dogovarjanja na državni ravni. INTEROPen podpira državna organizacija NHS Digital. V njej sodelujejo javne organizacije in razvojna podjetja. Skupaj stremijo k temu, da se vse aplikacije v njihov zdravstveni ekosistem integrirajo na enak način.

Izziv 2: Raznolika interpretacija pomena podatkovnih elementov standarda FHIR.

FHIR specifikacija določa veliko podatkovnih elementov, ki jih lahko uporabimo za prenos podatkov. Niso pa vsi elementi enolično razumljivi. Na primer, če želimo zapisati navodilo pacientu, naj vzame tableto na tešče, bi se lahko en razvijalec programske opreme odločil za uporabo podatkovnega elementa dosageInstruction -> text, drugi pa za dosageInstruction -> patientInstruction - glej Sliko 5. Problematično je, če razvijalci ene aplikacije shranijo podatek v en podatkovni element, razvijalci druge aplikacije pa podatek pričakujejo v drugem podatkovnem elementu.



Slika 5 : **Izsek iz specifikacije za dosageInstruction v FHIR STU3 (INTEROPen, 15.6.2020).**

Tabela 2: **Različni tipi navodil za doziranje zdravil ter njihovi podatkovni elementi v CareConnect FHIR (Radović, 2019).**

Vprašanja o doziranju	V kateri podatkovni element shranimo informacijo
Kako dolgo naj traja terapija?	Dosage instruction → timing → repeat → bounds → duration
Kako dolgo naj traja individualna doza	Dosage instruction → timing → repeat → duration
Koliko časa naj traja dobava zdravila?	Dispense request → expectedSupplyDuration
Od kdaj do kdaj naj traja terapija?	Dosage instruction → timing → repeat → bounds → period

V Tabeli 2 so primeri različnih navodil za doziranje zdravil. Struktura uporabljenih podatkovnih elementov je razvidna na Sliki 2. INTEROPen je na podlagi konkretnih primerov navodil osnoval dogovor, ki enolično določa, v katere podatkovne elemente naj se shranijo določeni tipi časovnih informacij o doziranju zdravila. Na primer, INTEROPen je podrobneje definiral strukturo podatkovnih elementov razširjenega objekta CareConnect Medication Resource in sicer s primeri, kdaj uporabiti *additionalInformation*, kdaj *patientInstruction* in kdaj *text*.

Izziv 3: Izdaja novih različic standarda FHIR zahteva premišljene vsebinske in časovne popravke v aplikacijah.

Kot omenjeno, se v trenutni knjižnici skupnosti INTEROPen uporablja FHIR-ova knjižnica različice STU3. Različica STU3 je izšla marca leta 2017. Z decembrom 2018 je na voljo nova različica imenovana Release #4. V pripravi je že Release #5, ki bo izdan v drugi polovici 2020. Knjižnica CareConnect FHIR se bo v prihodnosti zaradi napredka tehnologije morala prilagoditi in zamenjati bazno knjižnico FHIR STU3 s takrat najustreznejšo različico. Sprememba bazne knjižnice bo zahtevala spremembe v razširjenih objektih CareConnect FHIR in posledično prilagoditve vseh aplikacij, ki bodo do takrat integrirane na star način.

Izziv 4: Izdaja novih različic CareConnect FHIR zahteva premišljene vsebinske in časovne popravke v aplikacijah.

Objekti CareConnect FHIR se še vedno dopolnjujejo in spreminjajo. Trenutno je aktualna različica 2.5.0.-alpha.0. Pojavljajo se prve integracije po CareConnect FHIR objektih ter prvi seznami CareConnect RESTful API-jev[3], ki temeljijo na omenjenih objektih. Če se podatkovni element v nekem objektu spremeni, morajo razvojne organizacije spremembe implementirati.

Izziv 5: Izdaja novih različic zunanjih šifrantov zahteva premišljene vsebinske in časovne popravke v aplikacijah.

Aplikacije uporabljajo eksterne šifrante (kot je na primer dm+d), da bi lahko enolično komunicirale. Dm+d je šifrant vseh zdravil, ki se lahko predpisujejo pacientom v Angliji. Aplikacija za elektronsko predpisovanje OPENeP enkrat mesečno implementira spremembe na dm+d, ki so objavljene s strani NHS Digital. Spremembe lahko vključujejo dodelitev novih kod zdravilom. Če je bilo neko zdravilo prej pod kodo 1, je lahko v novi različici pod kodo 1001. Če bi OPENeP opravil to osvežitev podatkov enkrat na mesec, lekarniška aplikacija pa na tri mesece, bi lekarniški sistem javil napako in ne bi mogel izdati zdravila. Tudi če bi se integrirane aplikacije uskladile glede osvežitve aplikacij z najnovejšim šifrantom dm+d enkrat na mesec, bi se lahko še vedno pojavila časovna okna (na primer nekaj dni, tednov), ko aplikacije ne bi imele sinhroniziranih šifrantov.

6 ZAKLJUČEK

FHIR je pomemben integracijski standard, namenjen prenosu elektronskih zdravstvenih zapisov o pacientih med dvema aplikacijama (Wagholikar et al., 2016). Specifikacija je napisana tako, da ne omejuje objektov (Mandel et al., 2016), kar lahko predstavlja problem pri zagotavljanju varnosti pacientovih podatkov (Franz et al., 2015). V članku smo poudarili pomembnost enoličnega razumevanja vsebine podatkovnih elementov tega jezika tako s strani odjemalca kot strežnika. V članku smo najprej pripravili predlog preslikave pacientovih podatkov iz podatkovnih elementov standarda openEHR v podatkovne elemente standarda za integracijo (CareConnect) FHIR. Nato smo z na primeru predstavili izzive integracije s FHIR-jem.

V prvi vrsti predstavlja izziv vsebinski prenos podatkov o pacientu iz enega v drug standard. Vsako prepisovanje podatkov iz enega jezika/standarda v drugi lahko pomeni izgubo ali popačenje podatkov. Ponudnika konkretnih aplikacij morata doseči dogovor, kako bosta interpretirala podatkovne elemente integracijskega standarda FHIR. Drugi izziv predstavlja dejstvo, da ima standard openEHR nekatere svoje podatkovne elemente vsebinsko bolj podrobno razdelane kot FHIR in obratno. Posledično ni enolične interpretacije podatkovnih objektov FHIR. Tako se je skupnost INTEROPen v Angliji zavzela za svojo izdajo FHIR standarda/specifikacije CareConnect FHIR. Ta je osredotočena tako na definiranje novih manjkajočih podatkovnih elementov kot

[3] URL do navodil, kako izgraditi CareConnect API: https://nhsconnect.github.io/CareConnectAPI/.

tudi na podrobnejše definiranje uporabe obstoječih. Tretji izziv je, da je standard FHIR še v razvoju ter pridobiva nove različice vsaj enkrat letno. Tudi njegova razširitev CareConnect FHIR se še dopolnjuje in spreminja, kar predstavlja četrti izziv. Ponudniki aplikacij, ki uporabljajo CareConnect FHIR, morajo biti pripravljeni na prehode na nove različice standarda. Kot zadnji peti izziv smo izpostavili uporabo šifranta zdravil dm+d, ki ima prav tako svoje redne posodobitve, ki jih je treba upoštevati. Ponudniki integriranih aplikacij imajo velik izziv pri zagotavljanju sočasne posodobitve uporabljenih standardov in šifrantov.

Omejitev naše študije se nanaša na tabelo preslikav iz openEHR v CareConnect FHIR. Ta namreč predstavlja le krajši nabor možnih preslikav, ki izhajajo iz naročila nekega konkretnega zdravila. Prav tako je seznam izzivov, s katerimi se srečujejo deležniki pri integraciji aplikacij s standardom FHIR, v praksi daljši. Mi smo se osredotočili na tiste težavnejše. Ostajajo odprta raziskovalna vprašanja, kot je na primer: katere so pogoste napake z zdravili pri prenosu po standardu FHIR v primerjavi z njegovo razširitvijo CareConnect FHIR? Katere napake so v preseku in katere ne? V prihodnosti so potrebne tudi kvalitativne študije, v katerih bi definirali skupine težav s strani različnih deležnikov integracije s FHIR ter predlagali strategije reševanja le teh.

LITERATURA

[1] Apperta fundation. (2018). Defining an open platform. Retrieved from https://apperta.org/assets/Apperta_Defining_an_Open_Platform.pdf

[2] Board on health care services. (2004). *Patient safety: achieving a new standard for care*. Washington: The national academies press.

[3] Franz, B., Schuler, A., & Krauss, O. (2015). Applying FHIR in an integrated health monitoring system. *European federation of medical informatics*, *11*(2), 51-56.

[4] HL7. (2020). HL7 - FHIR Release 4. Retrieved from https://www.hl7.org/fhir/overview.html

[5] INTEROPen, H. (15.6.2020). HL7 UK INTEROPen CareConnect FHIR profiles: CareConnect-MedicationRequest-1. Retrieved from https://fhir.hl7.org.uk/STU3/StructureDefinition/CareConnect-MedicationRequest-1

[6] Li, Y. (2014). 5 keys to consider when storing and transforming medical images. *Becker's hospital review*.

[7] Mandel, J. C., Kreda, D. A., Mandl, K. D., Kohane, I. S., & Ramoni, R. B. (2016). SMART on FHIR: a standards-based, interoperable apps platform for electronic health records. *Journal of the american medical informatics association, 23*(5), 899–908. doi:10.1093/jamia/ocv189

[8] Meyer, M., Levine, W. C., Brzezinski, P., Robbins, R., Lai, F., Spitz, G., & Sandberg, W. S. (2005). *Integration of hospital information systems, operative and peri-operative information systems*. Paper presented at the AMIA Annual symposium

[9] NHS Digital. (2020). FHIR dose syntax implementation guidance. Retrieved from https://nhsconnect.github.io/Dose-Syntax-Implementation/

[10] openEHR Foundation. (2020). openEHR. Retrieved from https://www.openehr.org/about/what_is_openehr

[11] Parv, L., Kruus, P., Motte, K., & Ross, P. (2016). An evaluation of e-prescribing at a national level. *Informatics for health and social care, 41*(1), 78-95. doi:10.3109/17538157.2014.948170

[12] Radović, B. (2019). *Integration between electronic prescribing system and pharmacy information system using FHIR standard.* (Master's thesis), University of Ljubljana, Faculty of electrical ingeneering, https://repozitorij.uni-lj.si/IzpisGradiva.php?id=111243.

[13] Sarita, P., Dave, P., & Yunfeng, H. (2017). *Suitability of fast healthcare interoperability resources (FHIR) for wellness data* Paper presented at the Hawaii international conference on system sciences, Hawaii.

[14] Schleyer, T. K. L., Rahurkar, S., & Schaffer, J. T. (2019). *Preliminary evaluation of the Chest Pain Dashboard, a FHIR-based approach for integrating health information exchange information directly into the clinical workflow*. Paper presented at the AMIA Joint Summits on Translational Science proceedings.

[15] Shahmoradi, L., Habibi-Koolaee, M., Ebrahimi, M., Khoy, F. P., & Soltani, A. (2017). Middleware for the integration of hospital information systems. *Iranian journal of medical informatics, 6*(1), 28-32.

[16] Wagholikar, K. B., Mandel, J. C., Klann, J. G., Wattanasin, N., Mendis, M., Chute, C. G., . . . Murphy, S. N. (2016). SMART-on-FHIR implemented over i2b2 *Journal of the american medical informatics association, 24*(2), 398-402.

▪

Marina Trkman je doktorirala iz računalništva in informatike. Trenutno je zaposlena kot podoktorska raziskovalka na Institutu Jožef Stefan, kjer je bila zadnje tri leta odgovorna za potek projekta »Tehnološki in poslovni vidiki bodočega ekosistema za e-zdravstvo[4]« odobren na »Javnem razpisu za spodbujanje raziskovalcev na začetku kariere, 2.0«. Kot partner iz gospodarstva je pri projektu sodelovalo podjetje Better.

▪

Mitja Lapajne je diplomirani inženir računalništva in informatike. Trenutno je zaposlen kot arhitekt programske opreme v podjetju Better, kjer je že približno deset let odgovoren za delo razvijalcev na aplikaciji za elektronsko predpisovanje zdravil OPENeP. Njegove raziskave se osredotočajo na razvoj informacijskih tehnologij v zdravstvu.

▪

Božidarka Radović je magistrirala na področju biomedicine in elektrotehnike. Trenutno je zaposlena kot produktni vodja v podjetju Better, kjer je odgovorna za razvoj novih funkcionalnosti v aplikaciji za elektronsko predpisovanje zdravil OPENeP. Njene raziskave so povezane z uvajanjem informacijskih tehnologij v zdravstvu. V svojem magistrskem delu se je tako osredotočila na problem integracije različnih zdravstvenih sistemov s pomočjo odprtokodnih standardov.

[4] Povezava do projekta: https://www.e5.ijs.si/teba-ecosystem-slo/?lang=sl.

Semantična segmentacija aerolaserskih oblakov točk in centriranje višin globalnih soseščin

Jernej Nejc Dougan[1,3], Krištof Oštir[2], Matej Kristan[1]
[1]Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, Ljubljana,
[2]Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo, Jamova cesta 2, Ljubljana,
[3]Flycom Technologies d.o.o., Ljubljanska cesta 24A, Kranj
nejc.dougan@flycom.si, kristof.ostir@fgg.uni-lj.si, matej.kristan@fri.uni-lj.si

Izvleček

Aerolaserski oblaki točk so pomemben vir informacij v številnih prostorskih aplikacijah, kot na primer pri izdelavi digitalnih modelov terena ali kartiranju in popisu sredstev kritične infrastrukture. Semantična segmentacija se lahko uporablja v večini procesnih tokov obdelave aerolaserskih oblakov točk. V zadnjih letih najboljše rezultate za semantično segmentacijo in klasifikacijo dosegajo metode globokega učenja. Na kakovost segmentacije med drugim vpliva izbor soseščine točk in centriranje višine. V članku predstavimo in evalviramo različne metode za centriranje višin. Preizkuse smo izvedli na podatkovni zbirki ISPRS 3D Semantic Labelling, kjer smo s preprosto metodo centriranja najmanjše višine izboljšali rezultat za skoraj dva procenta.

Ključne besede: aerolasersko snemanje, globoko učenje, oblaki točk, semantična segmentacija

Abstract

Aerial laser scanning point clouds are an important data source in many geospatial applications such as digital terrain model generation or asset mapping of critical infrastructure. Semantic segmentation can be used in the majority of point cloud processing pipelines. Current state-of-the-art methods for semantic segmentation and classification are based on deep learning. The quality of semantic segmentation depends also on the neighbourhood selection and elevation centering. In this paper, we propose and evaluate different methods for elevation centering. Experiments on ISPRS 3D Semantic Labelling show that the use of minimal elevation centering increases results by nearly two percent.

Keywords: Aerial laser scanning, deep learning, point clouds, semantic segmentation.

1 UVOD

Klasifikacija in semantična segmentacija oblakov točk aerolaserskega snemanja (ALS) sta pomembna problema, ki zahtevata znanje daljinskega zaznavanja, fotogrametrije in računalniškega vida. Številne prostorske aplikacije, na primer izdelava digitalnih modelov reliefa, zaznavanje stavb, rekonstrukcija stavb, kartiranje in popis sredstev kritične infrastrukture, temeljijo na obdelanih oblakih točk. Obdelava zajema razdelitev točk v različne razrede, na primer za izdelavo digitalnega modela terena je potrebno točke razdeliti v točke terena in ostale. Velika večin trenutno obstoječih postopkov temelji na metodah, ki ne temeljijo na stojnem učnju, na primer matematična morfologija [Mongus et al., 2014]. Velik uspeh metod strojnega učnja in predvsem globokega učnja v slikovni domeni [Krizhevsky et al., 2012] je spodbudil raziskave na področju uporabe globokega učenja za oblake točk. Tradicionalni postopki, temelječ na strojnem učenju, za semantične segmentacijo oblakov točk ALS temeljijo na ročno ustvarjenih značilnicah in klasifikatorjih. Pred kratkim so se za semantično segmentacijo in klasifikacijo oblakov točk začele uporabljati globoke nevronske mreže [Qi et al., 2017a, Qi et al., 2017b, Thomas et al., 2019] in trenutno dosegajo tudi najboljše rezultate. Vendar trenutno najbolj-

še metode ne naslavljajo vseh karakteristik oblakov točk ALS.

Oblaki točk ALS so obsežni, lahko obsegajo celotne države, za obdelavo jih je potrebno razdeliti v manjše soseščine. Izboru soseščine moramo posvetiti posebno pozornost, saj se velikosti objektov lahko razlikujejo za celotne velikostne razrede, višina nad terenom pa predstavlja eno izmed pomembnejših značilnic za uspešno segmentacijo [Niemeyer et al., 2014]. Določanje višine terena zahteva predhodno določitev točk terena. Posledično bi točke terena morali obravnavati ločeno. Z uporabo centriranja višin globalnih soseščin lahko dobimo dober približek višine nad terenom in se izognemo kompleksni arhitekturi za ločeno obravnavanje točk terena.

V tem članku predlagamo tri preproste metode centriranja višine globalnih soseščin in uporabo centriranih višin kot vhodnih značilnic mrežo.

2 PREGLED METOD GLOBOKEGA UČENJE ZA OBLAKE TOČK

Klasične metode nadzorovanega strojnega učenja za oblake točk ALS izkoriščajo ročno oblikovane značilnice. Pogosto uporabljene značilnice temeljijo na lastnih vrednostih in so linearnost, planarnost, razpršenost, omnivarianca, anizotropija, vsota lastnih vrednosti, sprememba ukrivljenosti [Weinmann et al., 2015], le-te opisujejo, kako se točke porazdeljujejo v okolici točke ocenjevanja. Zaradi svoje sposobnosti vključevanja kontekstualne informacije je eden izmed bolj uporabljanih pristopov pristop pogojno slučajnih polj (angl. conditional random fields, CRF) [Weinmann et al., 2015, Vosselman et al., 2017, Niemeyer et al., 2014].

Uspeh metod globokega učenja [LeCun et al., 2015] v preteklih letih je navdihnil nove raziskave klasifikacije in semantične segmentacije 3D oblakov točk. Ena izmed glavnih prednosti metod globokega učenja je njihova zmožnost učenja značilnic in posledično odpravljena potreba po ročnem oblikovanju le-teh. Metode globokega učenja za oblake točk lahko razdelimo v dve kategoriji: (i) projekcijske in (ii) direktne metode. Projekcijske metode projicirajo točke v regularne 2D ali 3D mreže. Ob postopku projekcije se del informacije izgubi, pojavljajo se neželeni artefakti diskretizacije prostora. Direktne metode delujejo neposredno na oblakih točk in tako niso izpostavljene omenjenim pomanjkljivostim projekcijskih metod. Ključni izzivi direktnih metod so odkrivanje značilnic, definicija konvolucije in izbira soseščine. Trenutno najboljše so direktne metode, ki jih razdelimo v metode temelječe na: (i) točkovnih več-nivojski-perceptronih [Qi et al., 2017a, Qi et al., 2017b, Zhang et al., 2019], (ii) grafih [Wang et al., 2019, Landrieu and Simonovsky, 2018, Liu et al., 2019] in (iii) točkovnih konvolucijah [Thomas et al., 2019, Wang et al., 2018, Li et al., 2018, Wu et al., 2019].

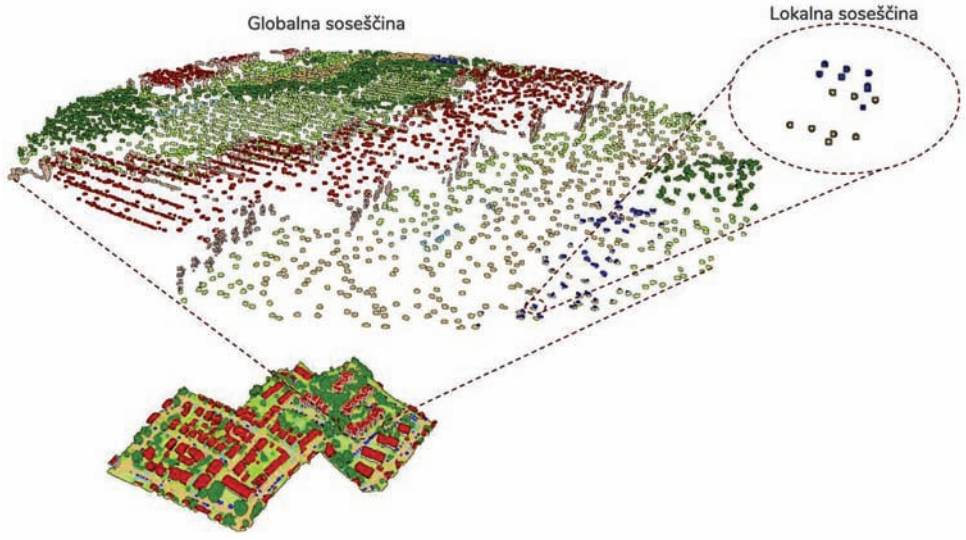
PointNet [Qi et al., 2017a] je bila prva globoka nevronska mreža za oblake točk, delujoča neposredno na točkah, ki je za odkrivanje značilnic uporabljala več točkovnih več-nivojskih-perceptronov (VNP). Trenutno najboljše rezultate dosegajo metode, ki temeljijo na točkovnih konvolucijah. Wang et al. [Wang et al., 2018] so na primer predlagali parametrično zvezno konvolucijo. Zvezna konvolucija za točke je definirana kot Monte-Carlo integracija parametrične funkcije, ki jo aproksimira VNP. Thomas et al. [Thomas et al., 2019] so predlagali novo konvolucijsko jedro definirano z jedrnimi točkami - Kernel Point Convolution (KPConv) in trenutno dosega najboljše rezultate na standardnih testih za semantično segmentacijo oblakov točk.

3 VIŠINA IN SOSEŠČINA

Oblaki točk ALS so praviloma preveliki, da bi jih lahko obdelovali naenkrat. Treba jih je razdeliti na manjša območja - soseščine. Izbor globalne in lokalne soseščine igra pomembno vlogo pri uspešnosti mreže, saj soseščina definira območni kontekst. Globalna soseščina je podmnožica točk iz celotnega oblaka točk, lokalna soseščina pa podmnožica točk za izračun enega koraka konvolucije (Slika 1). Izbira velikosti, centriranje višine in metode vzorčenja posamezne soseščine vplivajo na kakovost semantične segmentacije. Trenutno najboljša metoda KPConv [Thomas et al., 2019] uporablja fiksno sferično poizvedbo za globalno in lokalno soseščino. Višinsko so točke centrirane okoli točke poizvedbe. Velikost, oblika in vzorčenje soseščine so fiksni.

V izogib prevelike kompleksnosti mreže za določitev višine nad terenom predlagamo metode centriranja višin v globalnih soseščinah. Centrirane višine aproksimirajo višine nad terenom.

Globalna soseščina je podmnožica točk znotraj sfere s polmerom r in središčem v *središčni točki*. *Središčno točko* naključno izberemo iz množice vseh točk. Vse točke globalne soseščine centriramo po vseh treh prostorskih dimenzijah okoli *središčne toč-*



Slika 1: **Izbor globalne in lokalne soseščina vpliva na kakovost semantične segmentacije.**

ke. Višina je tako odvisna od izbrane središčne točke in je neuporabna kot značilnica. Zato predlagamo tri alternativne določitve globalne soseščine z uporabo centriranja višin: (i) centriranje s srednjo vrednostjo višine globalne soseščine, (ii) centriranje z najmanjšo vrednostjo višine globalne soseščine in (iii) centriranje z n-tim percentilom višin globalne soseščine. V ekstremnem primeru, kjer je teren popolnoma raven in velja $z = 0$ za vse točke, sta pri uporabi centriranja z najmanjšo vrednostjo centrirana višina in višina nad terenom enaki. Osnovno metodo smo dodatno razširili tako, da kot vhodno značilnico sprejme tudi centrirano višino. Lokalno soseščino smo definirali kot k-najbližjih sosedov.

4 EVALVACIJA

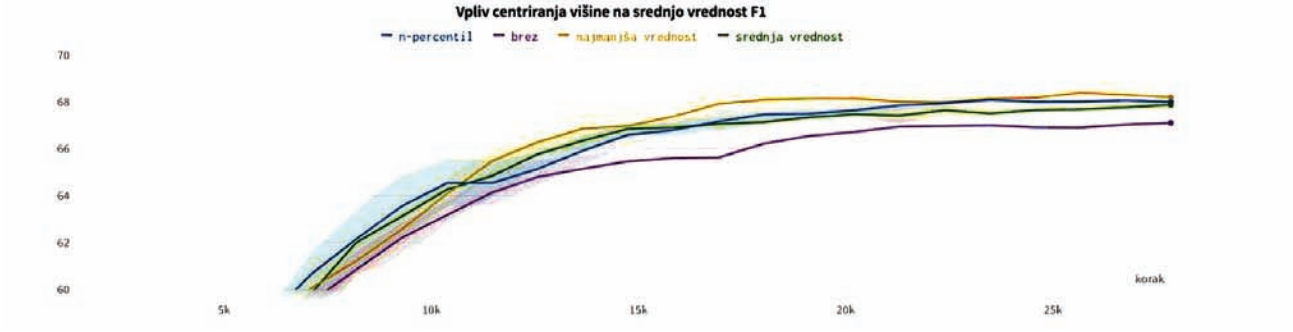
4.1 Eksperimenti

Za osnovno mrežo smo uporabili 5-nivojsko KP-Conv mrežo [Thomas et al., 2019] z deformabilnimi 15- točkovnmi jedri na zadnjih treh nivojih. Polmer globalne soseščine je 24 metrov, lokalno soseščino predstavlja 20 najbližjih sosedov. Parametre smo določili na podlagi preliminarnih eksperimentov. Osnovni mreži smo modificirali načine centriranja višine globalnih soseščin. Celoten algoritem je razdeljen na dva dela:(i) branje in priprava podatkov ter (ii) učenje. Branje in pripravo podatkov izvaja centralna procesna enota (CPU), učenje poteka na grafično procesni enoti (GPU). Zaradi velike neuravnoteženosti zastopanosti razredov smo za kriterijsko funkcijo uporabili uteženo križno entropijo (angl. weighted crossentropy) [Qi et al., 2017b]. Mrežo smo učili 500 epoh, kjer eno epoho sestavlja 50 korakov. Učenje smo izvedli na računalniku s procesorjem Intel Core i5-8400 in grafični kartici nVidia GTX 1080 Ti 11GB. Algoritem je implementiran v Python-u 3.6 z uporabo knjižnice Tensorflow 1.15.0. Povprečni čas učenja mreže in validacije je 3 ure, kjer en korak traja povprečno 350 milisekund. Znotraj enega koraka se obdela približno 72.200 točk.

Mrežo smo učili in testirali na podatkovni zbirki ISPRS 3D Semantic Labelling [Niemeyer et al., 2014]. Podatki so bili zajeti z instrumentom Leica ALS50, z višine 500 metrov nad terenom in vidnim poljem 45◦ [Cramer, 2010]. Podatki so označeni v devet semantičnih kategorij in razdeljeni v učno množico s 753.876 točkami in testno množico s 411.722 točkami. Pri izvedbi eksperimentov smo ohranili obstoječo razdelitev v učno in testno množico.

4.2 Rezultati

Za oceno rezultatov smo uporabili standardno proceduro na podatkih ISPRS 3D Semantic Labelling. Za vsako kategorijo posebej določimo oceno F1 (Enačba 1), kjer TP predstavlja pravilno pozitivne, FP nepravilno pozitivne in FN nepravilno negativne segmentirane točke. Skupna ocena je srednja vrednost ocen $F1$ vseh kategorij.



Slika 2: **Linije predstavljajo glajeno srednjo vrednost metrike mF1. Območje v ozadju predstavlja razpon med najmanjšo in največjo vrednostjo posamezne skupine.**

$$F1 = 2 * \frac{\frac{TP}{TP+FP} * \frac{TP}{TP+FP}}{\frac{TP}{TP+FP} + \frac{TP}{TP+FN}} \qquad (1)$$

Vse metode za centriranje višine globalne soseščine izboljšajo rezultate semantične segmentacije oblakov točk (Slika 2). Najboljše rezultate smo dosegli z metodo centriranja z najmanjšo vrednostjo višine in uporabo višine kot značilnice. Podrobni rezultati so prikazani v Tabeli 1. ISPRS oblak točk je pretežno ravninski, posledično je centriranje z minimalno vrednostjo zelo dober približek dejanske višine nad terenom.

V članku smo obravnavali problem izbora globalnih soseščin in centriranja višine pri semantični segmentaciji ALS oblakov točk. Zanimala sta nas vpliv višinske informacije na kakovost semantične segmentacije in zasnova učinkovite in preproste metode za centriranje višin globalne soseščine brez uporabe višine nad terenom. Ugotovili smo, da je najučinkovitejše centriranje z upoštevanjem najmanjše višine in vključitev višinske informacije kot značilnice v mrežo. Centriranje z upoštevanjem najmanjše višine v dani podatkovni zbirki najverjetneje tudi najboljše aproksimira dejansko višino točk nad terenom. S predlagano metodo smo dosegli oceno 69,19 srednje vrednosti F1. Preprosta sprememba je izboljšala rezultat v primerjavi z metodo brez centriranja za skoraj 2 odstotka srednje vrednosti F1.

V prihodnjih raziskavah bomo obravnavali problem izbora soseščin celostno, kjer bomo raziskali dodatne faktorje kot so oblika, velikost in vzorčenje. Dodatno bi bilo smiselno preveriti vpliv centriranja višine na višinsko bolj razgibani podatkovni zbirki. Prav tako bi bilo smotrno nasloviti problem nizkih osamelcev. Preproste metode centriranja bo verjetno treba nadomestiti z naprednejšo metodo, kot na primer z uporabo morfoloških profilov za aproksimacijo višine terena.

Semantična segmentacija oblakov točk je kompleksen problem, ki ga v industriji še vedno rešujejo pretežno ročno oziroma polsamodejno, kar je zamudno in neučinkovito. Vsakršne izboljšave metod imajo tako takojšen vpliv tako na znanost kot tudi industrijo.

Tabela 1: **Srednje vrednosti ocene F1 z uporabo različnih metod centriranja in brez. Uporaba višine kot značilnice je brezpredmetna, če višina ni centrirana.**

Centriranje višine	uporaba višine kot značilnice	srednje vrednosti F1
brez	ne	67,38
najmanjša vrednost	ne	67,74
srednja vrednost	ne	68,02
percentil	ne	68,32
brez	da	67,35
srednja vrednost	da	68,03
percentil	da	68,12
najmanjša vrednost	da	69,19

LITERATURA

[1] [Cramer, 2010] Cramer, M. (2010). The DGPF-test on digital airborne camera evaluation - Overview and test design. *Photogrammetrie, Fernerkundung, Geoinformation*, 2010(2):73–82.

[2] [Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and E. Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems*, 25.

[3] [Landrieu and Simonovsky, 2018] Landrieu, L. and Simonovsky, M. (2018). Large-Scale Point Cloud Se- mantic Segmentation with Superpoint Graphs. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 4558–4567.

[4] [LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep Learning. *Nature*, 521:436–444. [Li et al., 2018] Li, Y., Bu, R., Sun, M., Wu, W., Di, X., and Chen, B. (2018). PointCNN: Convolution on X-transformed points. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, number NeurIPS, pages 820–830. Curran Associates, Inc.

[5] [Liu et al., 2019] Liu, J., Ni, B., Li, C., Yang, J., and Tian, Q. (2019). Dynamic Points Agglomeration for Hierarchical Point Sets Learning. *IEEE International Conference on Computer Vision (ICCV)*, pages 7546–7555.

[6] [Mongus et al., 2014] Mongus, D., Lukač N., and Žalik, B. (2014). Ground and building extraction from LiDAR data based on differential morphological profiles and locally fitted surfaces. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93:145–156.

[7] [Niemeyer et al., 2014] Niemeyer, J., Rottensteiner, F., and Soergel, U. (2014). Contextual classification of lidar data and building object detection in urban areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87:152–165.

[8] [Qi et al., 2017a] Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017a). PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[9] [Qi et al., 2017b] Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017b). PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *CoRR*, abs/1706.0.

[10] [Thomas et al., 2019] Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., and Guibas, L. J. (2019). KPConv: Flexible and Deformable Convolution for Point Clouds. *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6410–6419.

[11] [Vosselman et al., 2017] Vosselman, G., Coenen, M., and Rottensteiner, F. (2017). Contextual segment- based classification of airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 128:354–371.

[12] [Wang et al., 2018] Wang, S., Suo, S., Pokrovsky, W.-C. M. A., and Urtasun, R. (2018). Deep parametric continuous convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2589–2597.

[13] [Wang et al., 2019] Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. (2019).

[14] Dynamic graph Cnn for learning on point clouds. *ACM Transactions on Graphics*, 38(5).

[15] [Weinmann et al., 2015] Weinmann, M., Schmidt, A., Mallet, C., Hinz, S., Rottensteiner, F., and Jutzi, B. (2015). Contextual classification of point cloud data by exploiting individual 3D neigbourhoods. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(3W4):271–278.

[16] [Wu et al., 2019] Wu, W., Qi, Z., and Fuxin, L. (2019). PointCONV: Deep convolutional networks on 3D point clouds. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:9613–9622.

[17] [Zhang et al., 2019] Zhang, Z., Hua, B.-S., and Yeung, S.-K. (2019). ShellNet: Efficient Point Cloud Convolu- tional Neural Networks using Concentric Shells Statistics. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*.

▪

Jernej Nejc Dougan je magistriral leta 2015 na Fakulteti za gradbeništvo in geodezijo Univerze v Ljubljani. Trenutno obiskuje doktorski študij na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Zaposlen je v podjetju Flycom Technologies d.o.o., kjer se raziskovalno ukvarja z metodami globokega učenja za obdelavo oblakov točk in drugimi analizami in obdelavami prostorskih podatkov.

▪

Krištof Oštir, prof. dr., je doktoriral leta 2000 na Fakulteti za gradbeništvo in geodezijo Univerze v Ljubljani. Kot predavatelj je zaposlen na Fakulteti za gradbeništvo in geodezijo Univerze v Ljubljani, kjer predava več do- in podiplomskih predmetov s področja geoinformatike, opazovanja Zemlje in obdelave podatkov. Glavno področje njegovega dela je optično in radarsko daljinsko zaznavanje. Opravljal je študije površja z radarsko interferometrijo, se ukvarjal z izdelavo digitalnih modelov višin, rabo in pokrovnostjo tal, po-obdelavo in mehko klasifikacijo. Ukvarja se z razvojem tehnologije malih satelitov za opazovanje Zemlje.

▪

Matej Kristan, izr. prof. dr., je doktoriral leta 2008 na Fakulteti za elektrotehniko Univerze v Ljubljani. Trenutno je član Laboratorija za umetne vizualne spoznavne sisteme (LUVSS) ter izredni profesor na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Njegovo raziskovalno področje obsega računalniški vid s poudarkom na vizualnem sledenju in semantični segmentaciji ter računalniškem vidu za avtonomne mobilne robote.

Interaktivna vizualizacija gosto poseljenih volumnov

Žiga Lesar, Matija Marolt
Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, Ljubljana
ziga.lesar@fri.uni-lj.si, matija.marolt@fri.uni-lj.si

Izvleček

V članku obravnavamo neposredno upodabljanje volumnov, ki vsebujejo številne podobne primerke struktur, kot so slike polikristalnih materialov, z vlakni ojačenih polimerov in znotrajceličnih struktur. V teh primerih je gostota primerkov tako velika, da obstoječe metode za upodabljanje ne omogočajo dobrega vpogleda v notranjost volumnov zaradi količine zastiranja med primerki. Predstavljamo novo metodo za upravljanje vidnosti pri vizualizaciji tovrstnih podatkov, ki nam omogoča skrivanje posameznih primerkov, poudarjanje primerkov z določenimi lastnostmi, redčenje primerkov in zlivanje segmentacije s surovimi podatki. Metodo smo preizkusili na sliki z vlakni ojačenega polimera, kjer omogoča boljši pregled notranjosti volumna.

Ključne besede: upodabljanje volumnov, gosto poseljeni volumni, upravljanje vidnosti, senčenje z usmerje- nim zastiranjem

Abstract

In this paper, we cover the direct rendering of volumes that contain numerous similar instances of structures such as images of polycrystalline materials, fibre-reinforced polymers or intracellular structures. In these cases, the density of the instances is so high that the existing rendering methods provide no insight into the volumes due to the amount of occlusion between the instances. We propose a new visibility management method for the visualization of such data, allowing us to hide individual instances, highlight instances with specific properties, sparsify instances and blend the segmentation with the raw data. We tested the method on fibre-reinforced polymer data in order to obtain a better view of the volume interior.

Keywords: Volume rendering, densely populated volumes, visibility management, directional occlusion shading.

1 UVOD

Neposredno upodabljanje volumnov (NUV) je družina metod za prikazovanje 3D skalarnih polj, ki jih pogosto najdemo v medicini, fiziki, kemiji, strojništvu ipd. Običajne metode za upodabljanje površin v sodobnih 3D aplikacijah ne omogočajo zadovoljivega vpogleda v volumetrično naravo podatkov, kar razkriva potrebo po posebnih metodah za upodabljanje volumnov. Natančna vizualizacija in interaktivnost sta bistvenega pomena pri upodabljanju volumnov, toda težko dosegljivi zaradi velike količine podatkov in zapletenosti simulacije osvetlitve. Predvsem v povezavi z interaktivnim raziskovanjem podatkov so te vizualizacijske metode pogosto združene s tehnikami *upravljanja vidnosti* za izpostavljanje ali poudarjanje določenih interesnih območij. Med tipičnimi primeri najdemo rezanje podatkov in prenosne funkcije, ki običajno potrebujejo zamudno ročno nastavljanje.

V določenih primerih nobena od obstoječih metod NUV ne deluje dovolj dobro, denimo na slikah polikristalnih materialov, z vlakni ojačenih polimerov in znotrajceličnih struktur. V tovrstnih podatkih je prostor gosto poseljen s številnimi primerki podobnih struktur, ki so postavljeni tesno skupaj, zato take volumne imenujemo *gosto poseljeni volumni*. Gostota primerkov je lahko celo tako velika, da ovira prostor-

sko zaznavo podatkov, saj je večina primerkov zakritih. Običajno nas ne zanima le en določen primerek, temveč njihova porazdelitev po prostoru, zato jih lahko postopoma odstranjujemo iz volumna, da razkrijemo več informacij o notranjosti. Obstoječe rešitve so za NUV neprimerne, saj so namenjene le upodabljanju površin.

Naš prispevek odpravi to pomanjkljivost z interaktivno metodo za upravljanje vidnosti v gosto poseljenih volumnih. Omogoča dinamično prilagajanje količine izrisanih primerkov in s tem razkrivanje tistih, ki bi sicer bili zastrti. Zanaša se na segmentacijo, ki vsakemu primerku priredi številčno oznako, opcijsko pa lahko uporabimo tudi dodatne lastnosti primerkov, ki jih moramo predložiti skupaj s segmentacijo. Z našo metodo ciljamo predvsem na interaktivno raziskovanje podatkov, kjer lahko primerke združujemo v skupine in obarvamo glede na njihove lastnosti, dobljene skupine pa nato razredčimo ali povsem odstranimo. Na mnogih področjih je tak način vizualizacije dobrodošel, saj nas pogosto zanimajo tisti primerki, ki odstopajo od povprečja ali zadoščajo nekemu danemu kriteriju. S segmentacijo izgubimo precejšnjo količino informacij iz surovih podatkov, zato naša metoda podpira tudi zlivanje segmentacije s surovimi podatki.

2 PREGLED PODROČJA

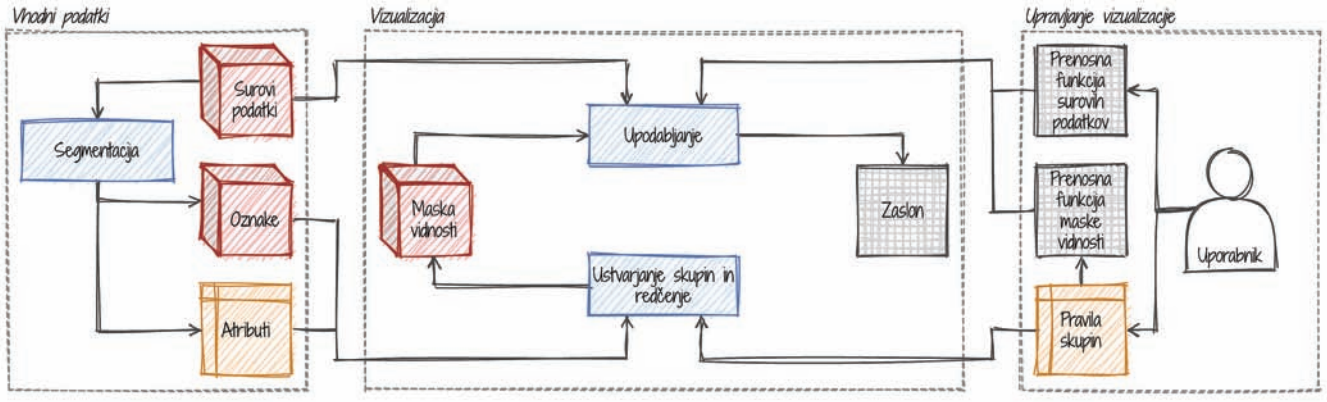
Področje NUV sega v 80. leta prejšnjega stoletja. Po dolgoletni prevladi ad-hoc metod [Max, 1995] se je celotno področje začelo preusmerjati k sledenju poti [Fong et al., 2017, Novák et al., 2018] - predvsem po zaslugi enostavne implementacije in skalabilnosti. Zaradi računske kompleksnosti simulacije svetlobe se za namene vizualizacije uporabljajo številne aproksimacije. Med najbolj razširjenimi sta sledenje poti z enkratnim sipanjem [Kroes et al., 2012] in senčenje z usmerjenim zastiranjem [Schott et al., 2009] ternjegova večsmerna razširitev [Šoltészová et al., 2010]. Jönsson idr. [Jönsson et al., 2014] v svojem poročilu opisujejo najpopularnejše osvetlitvene metode, njihovo tehnično stališče pa dopolnjujeta Lindemann in Ropinski [Lindemann and Ropinski, 2011], ki te metode predstavita iz vidika zaznavanja. Nekaj metod je bilo vključenih tudi v ogrodje VPT [Lesar et al., 2018], ki smo ga uporabili tudi za implementacijo metode, predstavljene v tem članku.

Vsebina tega članka se dotika tudi področja upravljanja vidnosti. Tu so v širši uporabi predvsem rezalni volumni in prenosne funkcije, pri čemer so slednje daleč najbolj pogoste. Podrobnejši pregled področja najdemo v preglednem članku [Viola and Gröller, 2005]. Načrtovanje dobrih prenosnih funkcij je težaven in dolgotrajen postopek, s katerim želimo izboljšati vidnost interesnih območij, pogosto s poskušanjem. Postopek je bil že delno avtomatiziran na različne načine [Ljung et al., 2016], toda v večini primerov je še vedno potrebno ročno nastavljanje.

Prenosne funkcije same zase ne rešujejo problema vizualizacije gosto poseljenih volumnov. Presli- kava med prosojnostjo in vidnostjo namreč ni enostavna, saj je zelo odvisna od zastiranja. Correa in Ma [Correa and Ma, 2009] sta v ta namen predstavila *histogram vidnosti*, ki meri delež površine zaslona, ki ga zasedajo določene strukture. Avtorja sta kasneje metodo razširila še s samodejnim generiranjem prenosnih funkcij [Correa and Kwan-Liu Ma, 2011]. Le Muzic idr. [Le Muzic et al., 2016] so predstavili *izena- čevalnik vidnosti* kot orodje za interaktivno raziskovanje mezoskopskih bioloških podatkov. Njihove podatke sestavljajo mnogi primerki proceduralno generiranih molekul, združenih v hierarhično strukturo. Vidnost posameznih nivojev v hierarhiji določi uporabnik, ocena dejanske vidnosti na zaslonu pa se uporabi za skrivanje in prikazovanje primerkov v realnem času. Koncept je odličen, toda metoda deluje le na površinskih podatkih, zato jo v tem članku razširjamo še na volumetrične podatke.

3 INTERAKTIVNO UPRAVLJANJE VIDNOSTI

Pregled naše metode začnimo z visokonivojskim opisom. Za boljše razumevanje poteka vizualizacije je zgradba metode prikazana tudi shematsko na sliki 1. Vhod v metodo predstavljajo surovi podatki in njihova predhodna segmentacija, ki vsakemu vokslu pripiše številčno oznako primerka. Poleg oznake lahko vsak primerek nosi še množico dodatnih atributov, kot so npr. volumen, površina, dolžina, orientacija ipd. Uporabnik nato združi primerke v skupine glede na tiste atribute, ki ga zanimajo ali pa predstavljajo neželeno informacijo. V eno skupino denimo združi primerke, katerih volumen presega določeno mejno vrednost, v drugo primerke z zanemarljivim volumnom, ki verjetno predstavljajo šum v segmentaciji, v tretjo pa preostale primerke. Nato vsaki skupini določi še barvo, prosojnost in stopnjo vidnosti, s katero določi delež skritih primerkov v tej skupini. Prvo skupino iz prejšnjega primera denimo obarva



Slika 1: **Pretok podatkov v predstavljeni metodi. S segmentacijo surovih podatkov dobimo volumen oznak in tabelo atributov. Z njimi generiramo masko vidnosti in ustrezno prenosno funkcijo, ki se nato skupaj s surovimi podatki in njihovo prenosno funkcijo uporabita v postopku upodabljanja na zaslon.**

rdeče in nastavi največjo stopnjo vidnosti, drugo skupino popolnoma skrije, tretjo pa obarva sivo in jo nekoliko razredči. Na podlagi opisanih pravil skupin se ustvari maska vidnosti in ustrezna prenosna funkcija, s pomočjo katere masko upodobimo na zaslon. Uporabnik lahko upodobi tudi surove podatke in jih med upodabljanjem zlije z masko vidnosti. S tem lahko poudari kontekst, oceni natančnost segmentacije ali prikaže informacije, ki so se v postopku segmentacije izgubile.

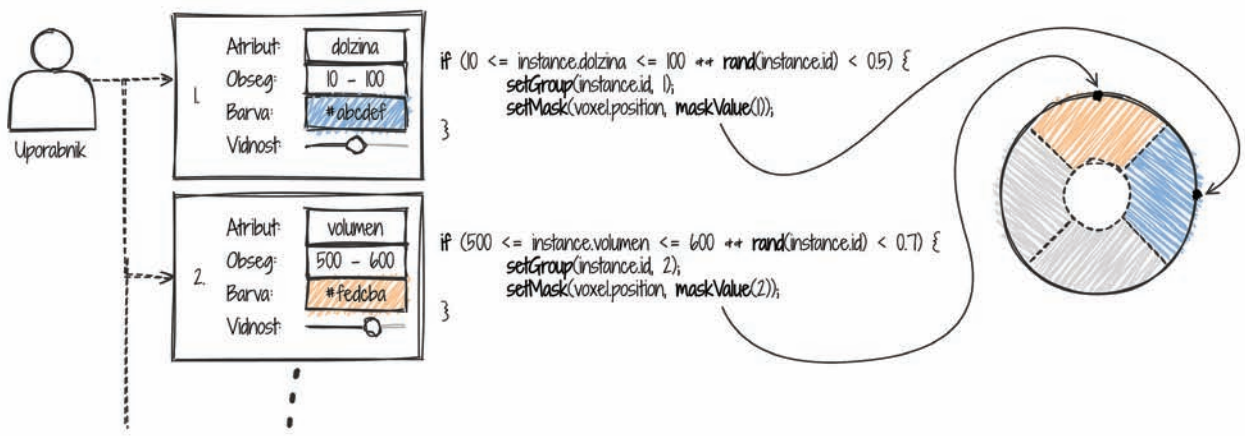
Medtem ko z upodabljanjem surovih podatkov ni posebnih težav, je upodabljanje segmentiranih podatkov malenkost bolj zapleteno. Na težave namreč naletimo pri vzorčenju in interpolaciji, saj številčnih oznak in atributov primerkov ne moremo interpolirati. V tem primeru lahko vzorčimo po metodi najbližjega soseda, toda s tem predpostavljamo, da so podatki nezvezni (so videti kockasti), slika pa je posledično nizke kakovosti.

Naša metoda ta problem reši z vpeljavo *maske vidnosti*, katere vrednosti lahko interpoliramo in klasificiramo po vzorčenju (angl. post-classification). S tem v volumnu obdržimo visokofrekvenčne komponente in hkrati ustvarimo kvalitetno končno sliko, kot navajata Engel in Ertl [Engel and Ertl, 2002]. Maska vidnosti služi dvema namenoma: omogoča strojno interpolacijo ter ločuje skupine primerkov od ozadja in med seboj. Med upodabljanjem jo vzorčimo in dobljene vrednosti z ustrezno prenosno funkcijo preslikamo v barvo in prosojnost, kot pri običajnem NUV. Vrednosti maske so točke v 2D prenosni funkciji, ki ustrezajo skupinam primerkov. Konkretne vrednosti so brez pomena, kritičen pa je njihov razpored po domeni prenosne funkcije, saj s tem neposredno vplivamo na interpolacijo in kvaliteto vzorčenja. Zaradi čim boljše izrabe domene prenosne funkcije smo se vrednosti odločili razporediti v obliki zvezde, kot je prikazano na sliki 3. Ozadje se preslika v središče prenosne funkcije, medtem ko so posamezne skupine razporejene enakomerno po obodu včrtanega kroga. Pri takem razporedu ima vsaka skupina prosto interpolacijsko pot do ozadja, kar omogoča vizualizacijo brez neželenih artefaktov na površinah primerkov [Lum and Ma, 2004], česar z enodimenzionalno prenosno funkcijo ne bi mogli doseči. Maska vidnosti in njena prenosna funkcija sta odvisni od uporabniško definiranih pravil skupin, zato ju posodabljamo dinamično ob vsaki uporabnikovi spremembi.

Preostane nam še vprašanje generiranja maske vidnosti. Vsak voksel se lahko preslika v ozadje ali v primerno skupino, kot določajo uporabniško definirana pravila. Za realizacijo redčenja naša metoda izkorišča številčno oznako primerka, ki jo z razpršilno funkcijo preslikamo na interval [0, 1), nato pa dobljeno vrednost primerjamo s stopnjo vidnosti te skupine. Glede na ta pogoj se nato odločimo, ali bomo dani primerek prikazali ali skrili. Primer uporabniško definiranih skupin in redčenja je prikazan na sliki 2.

Zadnja novost naše metode je zlivanje maske vidnosti s surovimi podatki. Med upodabljanjem vzorčimo oba volumna (V) in vzorce preslikamo prek ustreznih prenosnih funkcij (TF). Dobljene barve (C) in prosojnosti

(A) lahko nato linearno zlijemo med sabo, utež zlivanja pa določi uporabnik. V našem primeru smo se odločili uporabiti ločeni uteži za barvo (w_C) in prosojnost (w_A), saj lahko na ta način uporabnik preklaplja med barvami maske vidnosti ter surovih podat-



Slika 2: **Generiranje računalniškega senčilnika in prenosne funkcije na podlagi uporabniško definiranih pravil skupin**

kov brez spreminjanja prosojnosti. Zlivanje vpliva le na celoten volumen, čeprav včasih želimo upodobiti le razredčene surove podatke. Za redčenje surovih podatkov lahko prenesemo prosojnost iz maske vidnosti, uporabnik pa določi utež tega prenosa (w_T). Končno barvo in prosojnost posameznega vzorca na lokaciji **x** tako dobimo na sledeči način:

$$
\begin{aligned}
C_{raw},\, A_{raw} &= TF_{raw}(V_{raw}(\mathbf{x})),\\
C_{mask},\, A_{mask} &= TF_{mask}\,(V_{mask}(\mathbf{x})),\\
A' &= \mathrm{mix}(A_{\mathrm{mask}},\, A_{\mathrm{mask}}A_{\mathrm{raw}},\, w_T),\\
A &= \mathrm{mix}(A',\, A_{raw},\, w_A),\\
C &= \mathrm{mix}(C_{mask},\, C_{raw},\, w_C).
\end{aligned}
$$

4 IMPLEMENTACIJSKE PODROBNOSTI

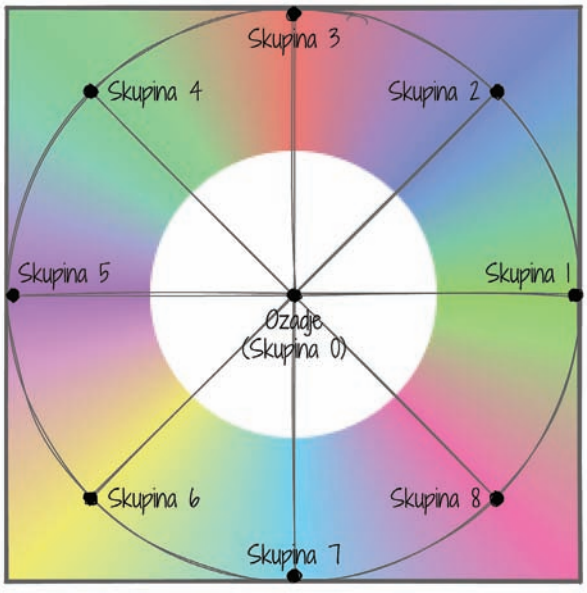
Interaktivnost vizualizacije je bila naša glavna prioriteta, zato smo jo optimizirali za izvajanje na GPE. Implementirali smo jo v ogrodju VPT [Lesar et al., 2018] z vmesnikom WebGL 2.0 Compute, ki omogoča uporabo računskih senčilnikov, napisanih v jeziku GLSL. Posebnosti GPE se v veliki meri odražajo v zgradbi metode, s katero rešujemo problem formata podatkov in strojne interpolacije.

Volumen številčnih oznak shranjujemo na GPE v obliki 3D teksture 32-bitnih nepredznačenih števil, atribute primerkov pa v pomnilniškem objektu senčilnika (angl. shader storage buffer object, SSBO). Format in razpored podatkov znotraj pomnilniškega objekta smo prilagodili uporabi v senčilniku, tako da se neposredno preslika v polje struktur, kjer vsaka struktura hrani vrednosti atributov posameznega primerka. Številčne oznake primerkov na ta način delujejo kot indeksi za dostopanje do njihovih atributov v SSBO. Ker je razpored podatkov specifičen za vsak vhodni volumen, ga moramo aplikaciji predložiti skupaj z volumnom in atributi. V našem primeru gre za dokument JSON, ki vsebuje seznam atributov z njihovimi imeni in tipi, kar je dovolj za generiranje definicije strukture v jeziku GLSL.

Do atributov primerkov dostopamo v računskem senčilniku, ko generiramo masko vidnosti. Kot je prikazano na sliki 2, se pravila skupin prevedejo neposredno v seznam pogojnih stavkov v jeziku GLSL, ki vsakemu vokslu priredijo vrednost maske vidnosti. To hranimo na GPE kot 3D teksturo štirih 8-bitnih vrednosti. Izbira formata ni naključna: gre za enega izmed formatov tekstur, v katere lahko pišemo z računskim senčilnikom, polega tega pa podpira strojno interpolacijo. Računski senčilnik generiramo dinamično ob vsaki spremembi pravil skupin ali množice atributov. Generiranje maske je neodvisno glede na voksle, zato komunikacije med delovnimi skupinami senčilnika ni, dimenzije posamezne delovne skupine pa so torej lahko poljubne. V naši implementaciji smo izbrali dimenzije 16 × 16 × 1.

5 REZULTATI

Metodo smo preizkusili na sliki z vlakni ojačenega polimera velikosti 400 401 800, ki so jo zajeli in segmentirali raziskovalci z Univerze uporabnih znanosti Zgornja Avstrija. Njihova vizualizacija volumna z namenskim orodjem FiberScout [Weissenbock et al., 2014] ni bila zadovoljiva, saj to orodje ne vsebuje funkcionalnosti, predstavljenih v tem članku. Volu-



Slika 3: **Vrednosti maske vidnosti se preslikajo v domeno prenosne funkcije v obliki zvezde. Č rte predstavljajo interpolacijske poti med skupinami primerkov in ozadjem.**

men vsebuje 3828 primerkov vlaken in 18 atributov na primerek. Za upodabljanje smo uporabili metodo senčenja z usmerjenim zastiranjem [Schott et al., 2009], ki je dovolj preprosta, da se izvaja v realnem času, hkrati pa ustvari dovolj dobre sence. Slika 4 prikazuje upodobitve volumna z različnimi nastavitvami vidnosti. Vlakna smo združili v skupine in obarvali glede na različne vrednosti atributov, s čimer smo poudarili vlakna, ki so npr. prekratka ali napačno orientirana. Slika 5 prikazuje postopek zlivanja podatkov v primeru, ko vizualizacija razredčenih segmentiranih podatkov ni dovolj informativna, zato je dodan kontekst, prosojnost pa prenešena iz surovih podatkov.

Metodo smo testirali v brskalniku Google Chrome 83 na prenosnem računalniku z integrirano GPE Intel HD Graphics 530 ter na namiznem računalniku z grafično kartico Nvidia GeForce GTX 1060. Na obeh napravah je vizualizacija tekla interaktivno in v realnem času. Večino računske moči je zahtevalo upodabljanje, medtem ko je bila vsaka posodobitev podatkov ob spremembah klasifikacijskih pravil nezaznavna.

Od raziskovalcev, ki so priskrbeli podatke, smo že prejeli pozitiven odziv na uporabnost naše metode. Zatrdili so, da je z našo metodo moč bolje vizualizirati notranjost polimerov kot s katero koli obstoječo metodo. Po njihovem mnenju je najbolj uporabna zmožnost dinamičnega prilagajanja pravil skupin in gostote primerkov ter zlivanje prikaza s surovimi podatki.

6 DISKUSIJA

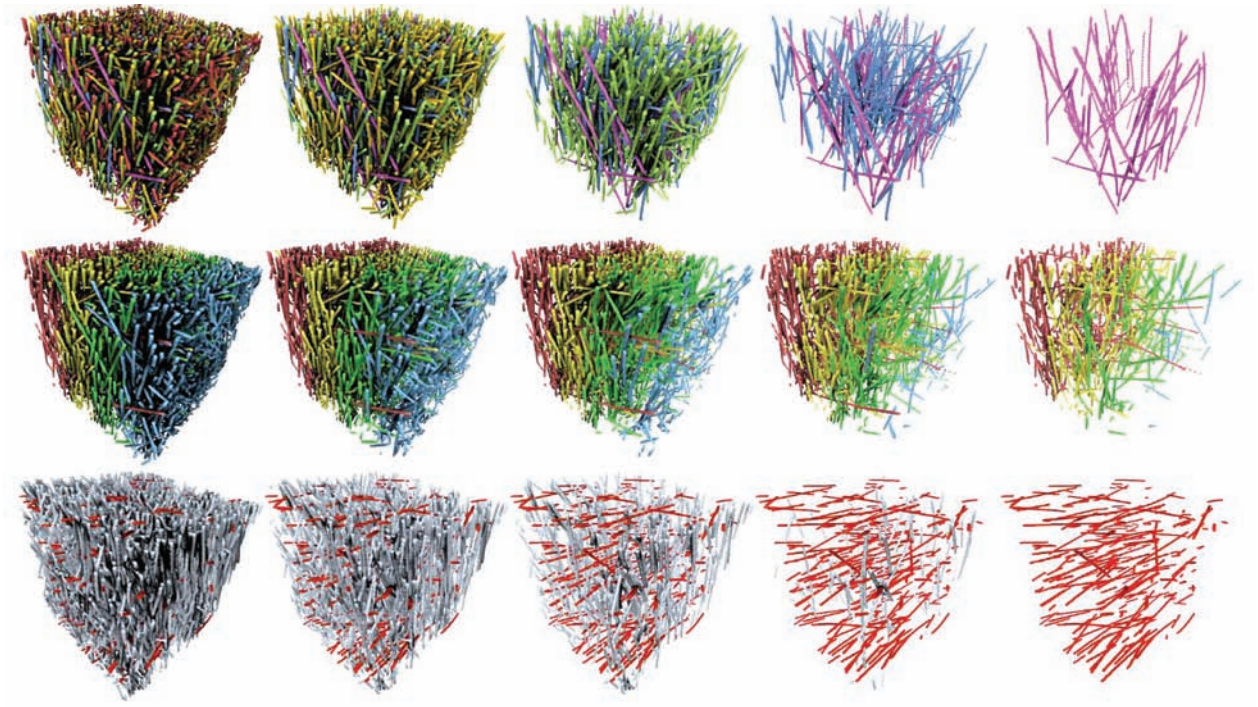
V tem članku smo predstavili metodo za interaktivno upravljanje vidnosti v gosto poseljenih volumnih. Uporabniki lahko primerke struktur združujejo v skupine glede na preprosta pravila, nato pa dinamično prilagajajo njihovo barvo in gostoto ter s tem razkrijejo več informacij o notranjosti volumnov. To smo dosegli z ločitvijo upodabljanja in razporejanja v skupine. V ta namen smo razvili postopek generiranja maske vidnosti in prenosne funkcije v proceduralno generiranem senčilniku. Dodali smo tudi možnost zlivanja surovih in segmentiranih podatkov, s katerim lahko poudarimo kontekst ali razkrijemo napake segmentacije. Naša metoda je prva, ki omenjene funkcionalnosti povezuje z NUV.

Še vedno pa ostaja nekaj odprtih vprašanj. Kljub interpolaciji lahko površine primerkov so videti nekoliko nazobčane, kar je posledica vrednosti maske vidnosti. Za vsak voksel je namreč odločitev med klasifikacijo v ustrezno skupino ali v ozadje binarna. Temu bi se lahko izognili z glajenjem maske vidnosti in s tem nazobčanost zmanjšali ali celo popolnoma odstranili. Drug problem leži v pogostem ponovnem generiranju in prevajanju računskega senčilnika, kar je lahko na nekaterih napravah počasno. Senčilnik bi bil lahko generičen: s tem bi omogočal manj raznolika klasifikacijska pravila, toda ponovno prevajanje ne bi bilo potrebno. Ali bi bila ta omejitev za praktično uporabo prestroga, še ni jasno.

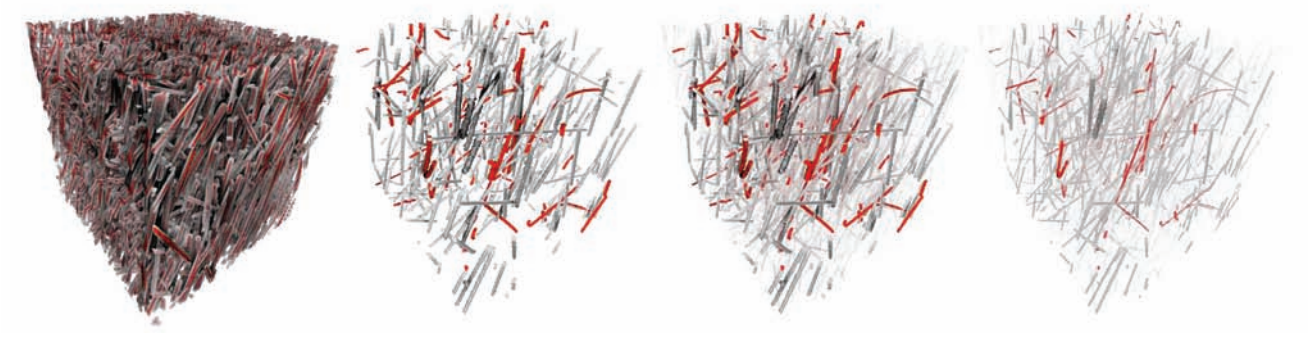
Kljub številnim možnostim za izboljšave je naša metoda velik korak v smeri interaktivne vizualizacije gosto poseljenih volumnov. Verjamemo, da se bo za uporabno izkazala še v mnogih znanstvenih disciplinah, kjer se znanstveniki s tovrstnimi podatki vsakodnevno srečujejo.

ZAHVALA

Radi bi se zahvalili Christophu Heinzlu in njegovi raziskovalni skupini z Univerze uporabnih znanosti Zgornja Avstrija za podatke ojačenega polimera in za dragocen odziv na naše delo.



Slika 4: **Primerki vlaken so razdeljeni v skupine glede na dolžino (zgoraj), lokacijo (sredina) in orientacijo (spodaj). Od leve proti desni so določene skupine vlaken postopoma razredčene, tako da do izraza pridejo najdaljša (zgoraj), napačno locirana (sredina) in napačno orientirana (spodaj) vlakna.**



Slika 5: **Zlivanje segmentiranih in surovih podatkov. Od leve proti desni: vizualizacija surovih podatkov, vizualizacija segmentiranih podatkov, dodajanje konteksta v vizualizacijo in prenos prosojnosti iz surovih podatkov.**

LITERATURA

[1] [Correa and Kwan-Liu Ma, 2011] Correa, C. D. and Kwan-Liu Ma (2011). Visibility Histograms and Visibility- Driven Transfer Functions. *IEEE Transactions on Visualization and Computer Graphics*, 17(2):192–204.

[2] [Correa and Ma, 2009] Correa, C. D. and Ma, K.-L. (2009). Visibility-driven transfer functions. In *2009 IEEE Pacific Visualization Symposium*, pages 177–184. IEEE.

[3] [Engel and Ertl, 2002] Engel, K. and Ertl, T. (2002). Interactive high-quality volume rendering with flexible consumer graphics hardware. In *Eurographics*. Eurographics Association.

[4] [Fong et al., 2017] Fong, J., Wrenninge, M., Kulla, C., and Habel, R. (2017). Production volume rendering.

[5] In *ACM SIGGRAPH*, pages 1–79, New York, New York, USA. ACM Press.

[6] [Jönsson et al., 2014] Jönsson, D., Sundén, E., Ynnerman, A., and Ropinski, T. (2014). A Survey of Volumetric Illumination Techniques for Interactive Volume Rendering. *Computer Graphics Forum*, 33(1):27– 51.

[7] [Kroes et al., 2012] Kroes, T., Post, F. H., and Botha, C. P. (2012). Exposure Render: An Interactive Photo-Realistic Volume Rendering Framework. *PLoS ONE*, 7(7):e38586.

[8] [Le Muzic et al., 2016] Le Muzic, M., Mindek, P., Sorger, J., Autin, L., Goodsell, D. S., and Viola, I. (2016). Visibility Equalizer Cutaway Visualization of Mesoscopic Biological Models. *Computer Graphics Forum*, 35(3):161–170.

[9] [Lesar et al., 2018] Lesar, Ž., Bohak, C., and Marolt, M. (2018). Real-time interactive platform-agnostic volumetric path tracing in webGL 2.0. In *Proceedings of the 23rd International ACM Conference on 3D Web Technology - Web3D '18*, pages 1–7, New York, New York, USA. ACM Press.

[10] [Lindemann and Ropinski, 2011] Lindemann, F. and Ropinski, T. (2011). About the Influence of Illumination Models on Image Comprehension in Direct Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1922–1931.

[11] [Ljung et al., 2016] Ljung, P., Krüger, J., Groller, E., Hadwiger, M., Hansen, C. D., and Ynnerman, A. (2016). State of the Art in Transfer Functions for Direct Volume Rendering. *Computer Graphics Forum*, 35(3):669– 691.

[12] [Lum and Ma, 2004] Lum, E. and Ma, K.-L. (2004). Lighting transfer functions using gradient aligned sampling. In *IEEE Visualization*, pages 289–296. IEEE Comput. Soc.

[13] [Max, 1995] Max, N. (1995). Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108.

[14] [Novák et al., 2018] Novák, J., Georgiev, I., Hanika, J., Kˇrivánek, J., and Jarosz, W. (2018). Monte Carlo methods for physically based volume rendering. In *ACM SIGGRAPH 2018 Courses on - SIGGRAPH '18*, pages 1–1, New York, New York, USA. ACM Press.

[15] [Schott et al., 2009] Schott, M., Pegoraro, V., Hansen, C., Boulanger, K., and Bouatouch, K. (2009). A Directional Occlusion Shading Model for Interactive Direct Volume Rendering. *Computer Graphics Forum*, 28(3):855–862.

[16] [Šoltészová et al., 2010] Šoltészová, V., Patel, D., Bruckner, S., and Viola, I. (2010). A multidirectional occlusion shading model for direct volume rendering. *Computer Graphics Forum*, 29(3):883–891.

[17] [Viola and Gröller, 2005] Viola, I. and Gröller, E. (2005). Smart Visibility in Visualization. In Neumann, L., Sbert, M., Gooch, B., and Purgathofer, W., editors, *Computational Aesthetics in Graphics, Visualization and Imaging*, pages 209–216. The Eurographics Association.

[17] [Weissenbock et al., 2014] Weissenbock, J., Amirkhanov, A., Weimin Li, Reh, A., Amirkhanov, A., Groller, E., Kastner, J., and Heinzl, C. (2014). FiberScout: An Interactive Tool for Exploring and Analyzing Fiber Reinforced Polymers. In *2014 IEEE Pacific Visualization Symposium*, pages 153–160. IEEE.

▪

Žiga Lesar je asistent in doktorski študent na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Ukvarja se predvsem z računalniško grafiko in visoko zmogljivim računalništvom, raziskuje pa interaktivno upodabljanje volumetričnih podatkov s spletnimi tehnologijami. Za svoje delo je leta 2014 prejel univerzitetno Prešernovo nagrado.

▪

Matija Marolt je izredni profesor na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Je predstojnik Laboratorija za računalniško grafiko in multimedije. Njegove raziskave so na področjih pridobivanja informacij iz glasbe s poudarkom na semantičnih opisih in razumevanju zvočnih signalov, pridobivanju in organizaciji glasbenih arhivov in interakcije med človekom in računalnikom.

Sistematični pregled literature agilnih in vitkih pristopov k razvoju varne programske opreme

Anže Mihelič[1,2,3], Simon Vrhovec[1], Tomaž Hovelja[3]
[1]Univerza v Mariboru, Fakulteta za varnostne vede, Kotnikova 8, 1000 Ljubljana
[2]FernUniversität in Hagen, Fakultät für Mathematik und Informatik, Universitätsstraße 47, 58097 Hagen
[3]Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana anze.mihelic@um.si, simon.vrhovec@um.si, tomaz.hovelja@fri.uni-lj.si

Izvleček

Izvedli smo sistematičen pregled literature v štirih bibliografskih zbirkah, pri čemer smo se osredotočali na pomanjkljivosti trenutnih preglednih del. Identificirali smo 23 predlaganih pristopov, ki so bili večinoma teoretični. Le 21,7 odstotkov pristopov je bilo empirično preverjenih v industrijskih okoljih. Vsi identificirani pristopi temeljijo na predpostavki, da varnost v razvojnem procesu ni zadostno upoštevana, ker varnostni elementi niso sestavni in stalni del agilnih metod. Najpogosteje dodani varnostni elementi so procesi (48 odstotkov), sledita kombinacija procesov in artefaktov (26 odstotkov) in kombinacija procesov, artefaktov in vlog (13 odstotkov).
Ključne besede: metodologija, informacijska varnost, računalniška varnost, agilne metode, razvoj varne programske opreme

Abstract

We conducted a systematic literature survey in four bibliographic databases. We focused on secure software development with special attention to the shortcomings of existing surveys. We identified 23 approaches. Most identified approaches were theoretical and only 21.7 percent were empirically tested in an industrial setting. All identified approaches are based on the assumption that security is not considered in the development process since security elements are not an integral and permanent part of agile methods. The most frequently proposed security elements are processes (48 percent), followed by combination of processes and artefacts (26 percent) and combination of processes, artefacts and roles (13 percent).
Keywords: Methodology, information security, computer security, agile methods, secure software development.

1 UVOD

Agilne metode razvoja programske opreme (npr. Scrum in Extreme Programming) so se pojavile kot odgovor na pomanjkljivosti tradicionalnih metod [Oueslati et al., 2015]. Glavne značilnosti agilnih metod so iterativni in postopni pristop, samoorganizirajoče se skupine, vsakodnevno komuniciranje med člani skupine in hitre povratne informacije [Adelyar and Norta, 2016, Pohl and Hof, 2015]. Zaradi teh značilno- sti agilne metode predstavljajo zelo prilagodljiv, učinkovit in hiter pristop k razvoju programske opreme [Gwanhoo and Weidong, 2010, Jyothi and Rao, 2011, Othmane et al., 2014]. Ker je osrednji poudarek agil- nih metod na funkcionalnih zahtevah razvite programske opreme, varnost pa ima močno nefunkcionalno kakovostno komponento, agilne metode niso povsem primerne za varen razvoj programske opreme [Tøndel and Jaatun, 2020, Bishop and Rowland, 2019]. Poleg omenjenega je manj kot 50 odstotkov tradici- onalnih dejavnosti zagotavljanja varnosti združljivih z agilnimi metodami, povsem primernih pa je manj kot 10 odstotkov [Beznosov and Kruchten, 2005].

S selitvijo trga s prodaje programske opreme kot izdelka na prodajo programske opreme kot storitve, je ideja o združevanju razvojnih in operativnih sku-

pin rezultirala v paradigmi DEVOPS [Lwakatare et al., 2016, Myrbakken and Colomo-Palacios, 2017]. Tako kot agilni, tudi DEVOPS, sam po sebi ni prilagojen za razvojne varne programske opreme [Lee, 2018]. Kot rešitev te pomanjkljivosti se je pojavil DEVSECOPS, ki DEVOPS združuje z varnostjo [Allison et al., 2020]. Zavzema se za implementacijo varnostnih elementov v vsaki fazi razvojnega procesa. Bolj kot nabor posebnih orodij in ukrepov, DEVSECOPS predstavlja ogrodje za razvoj varne programske opreme [Allison et al., 2020, Bezdedeanu, 2019, Kiuwan, 2019].

Č etudi je v literaturi mogoče zaslediti nekaj sistematičnih pregledov literature [Inayat et al., 2015, Mellado et al., 2010] in primerjalnih študij [Curcio et al., 2018, Khan and Ikram, 2017] na področju agilnih pristropov k razvoju programske upreme, so se takšni pregledi, ki se osredotočajo na razvoj *varne* pro- gramske opreme [Rindell et al., 2017, Kasauli et al., 2018, Barbosa and Sampaio, 2017] in ogrodja DEVSE- COPS [Myrbakken and Colomo-Palacios, 2017] pojavili šele pred kratkim. Pregledi literature na področju agilnega varnega razvoja programske opreme so k problemu pristopali z različnih zornih kotov. Tako po- nujajo celovite vodiče za izboljševanje varnostih ukrepov v agilnih projektih [Barbosa and Sampaio, 2017], pregled agilnih pristopov k razvoju varnostno kritičnih sistemov [Kasauli et al., 2018], pregled agilnih pri- stopov inženirstva zahtev [Villamizar et al., 2018], pregled agilin metod razvoja varne programske opreme [Rindell et al., 2017], in pregled izzivov in rešitev na tem področju [Oueslati et al., 2015, Riisom et al., 2018]. Neodvisno od agilnih pristopov, se je le en pregledni prispevek osredotočal na ogrodje DEVSECOPS [Myrbakken and Colomo-Palacios, 2017], ki pa se osredotoča na poskus iskanja ustrezne opredelitve de- finicije tega ogrodja in ne na varnostne elemente in rešitve, ki bi bile lahko vključene v proces razvoja in vzdrževanja.

Naš prispevek gradi na omenjenih pregledih literature, vendar se osredotoča na širši časovni okvir, poleg agilnih pa vključuje tudi vitke pristope, ki se pogosto omenjajo skupaj z ogrodjem DEVSECOPS. V prispevku bomo odgovorili na naslednja raziskovalna vprašanja.

- **RV1:** Katera kategorija elementov je najpogosteje predlagana kot rešitev za razvoj varne programske opreme?
- **RV2:** Kako so bili predlagani pristopi empirično preverjeni?
- **RV3:** Na kakšen način je varnost vključena v predlagane pristope?

Da bi raziskali kakšne rešitve ponuja obstoječa literatura, smo opravili pregled agilnih in vitkih pristopov k razvoju varne programske opreme. Sistematično smo pregledali relevantno literaturo v znanstvenih bibliografskih zbirkah. Identificirane elemente smo nato razvrstili v tri kategorije (artefakti, vloge, procesi) in analizirali njihovo kompatibilnost z agilnimi in vitkimi pristopi glede na njihovo osnovno predpostavko. Prav tako je cilj prispevka identificirati vse pristope k razvoju varne programske opreme, ki so bili zabeleženi v znanstveni literaturi v zadnjih enajstih letih.

2 METODA

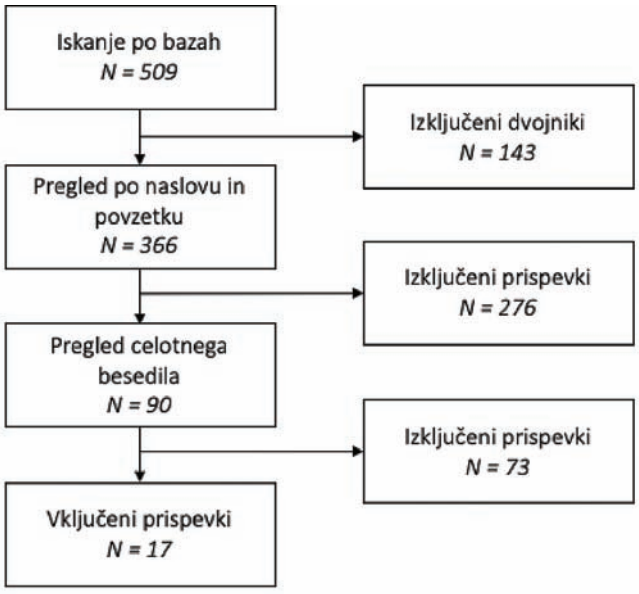
Opravili smo sistematičen pregled literature, ki je naslavljala probleme, usmerjene v razvoj varne programske opreme. Pregled je obsegal prispevke na konferencah in članke v znanstvenih revijah, objavljenih od leta 2009. Slika 1 prikazuje proces pregleda s številom vključenih del v posameznem koraku.

Pregled je bil opravljen po naslednjem postopku. V znanstvenih bibliografskih zbirkah (ACM DL, IEEE Xplore, Scopus in Web of Science) smo 8. januarja 2020 izvedli poizvedbo po naslovu in povzetku s kombinacjo ključnih besed *agilno, vitko, varno, programska oprema, razvoj, inženiring, metoda* in *upravljanje*. Za poizvedbo je bil uporabljen iskalni niz:

(agile **OR** lean) **AND** (secur* **AND** software **AND** (development **OR** engineering) **AND** (method* **OR** manag*)),

ki je bil prilagojen posamezni bibliografski zbirki. Iskanje je vrnilo skupno $N = 509$ zadetkov od leta 2009. Vse rezultate smo izvozili in jih shranili v lastno zbirko podatkov.

V naslednjem koraku smo odstranili vse podvojene zapise, kar je rezultiralo v $N = 366$ unikatnih bibliograf- skih zapisih, ki smo jih nato na podlagi pregleda naslova in povzetka izločali skladno z vključitvenimi in izključitvenimi kriteriji predstavljenimi v tabeli 1. V pregled celotnega besedila so bila vključena tudi dela, pri katerih iz naslova ali povzetka ni bilo



Slika 1: **Proces sistematičnega pregleda literature.**

jasno ali zadoščajo omenjenim kriterijem. Končno je bilo pregledano celotno besedilo $N = 66$ del in vključenih $N = 17$ del, v katerih so bili predlagani novi pristopi k naslavljanju problema razvoja varne programske opreme. Rezultati po bibliografskih bazah so predstavljeni v tabeli 2. Dodaten pregled izveden po metodi snežne kepe, je rezultiral v dodatnih $N = 6$ delih objavljenih leta 2009 ali kasneje, katerih izvor je bilo mogoče najti med deli, vključenimi v sistematični pregled literature.

3 REZULTATI IN RAZPRAVA

V nadaljevanju predstavljamo obstoječe pristope, ki predlagajo posebne rešitve za vključitev varnostnih praks v postopek razvoja programske opreme. Dela predstavljamo v tabelah 3 in 4. Posamezni pristopi so razvrščeni v skupine glede na to, na katero kategorijo elementov ali njihovo kombinacijo je osredotočen: procese (P), artefakte (A) ali vloge (V).

Odgovor na RV1: Rezultati nakazujejo, da so najpogosteje (48%) predlagani elementi, ki spadajo v kateogirijo procesov [Mougouei et al., 2013, Daud, 2010, Pohl and Hof, 2015, Rygge and Jøsang, 2018, Williams and Meneely, 2010, Tøndel et al., 2019, Yu and Le, 2012, Singh, 2018, Nguyen and Dupuis, 2019, Koc et al., 2019, Giacalone et al., 2014]. Sledi mu kombinacija procesov in artefaktov (26%) [Stålhane and Johnsen, 2017, Singhal and Singhal, 2011, Othmane et al., 2014, Maier et al., 2017,

Tabela 1: **Vključitveni in izključitveni kriteriji**

Vključitveni kriteriji	Izključitveni kriteriji
Članek objavljen v znanstveni reviji ali prispevek na konferenci	Teoretični pregledni članek
Povezan z razvojem varne programske opreme	Članek ni v angleškem jeziku ne
Povezan z nefunkcionalnimi zahtevami	Ni povezan z agilnimi ali vitkimi metodami
Povezan z razvojem varnostno-kritičnih sistemov	Objavljen leta 2009 ali kasneje
Ima izvirni znanstveni prispevek	Celotno besedilno ni dostopno

Tabela 2: **Zadetki po bibliografskih bazah.**

Biografska zbirka	Skupaj zadetkov	Pregled celotnega besedila	Vključwni
Web of Science	106	27	7
Scopus	171	30	8
IEEE Xplore	166	5	0
ACM DL	66	4	2

Tabela 3: **Predlagani pristopi, ki izvajajo iz sistematičnega pregleda literature. Črke v oglatih oklepajih nakazujejo kateri tip predlaganih elementov je dominanten (P - procesi, A - artefakti, V - vloge)**

Vir	Metoda	Kratek opis
[Ghani et al., 2014]	Študija primera v industrijskem okolju	Predlagani pristop v Scrum metodo uvaja varnostno specifičen seznam zahtev – security backlog. **[A]**
[Baca et al., 2015]	Kvazieksperiment v industrijskem okolju	Pristop Security-Enhanced Agile Software Develop- ment Process (SEAP) predlaga vključitev skupine za varnost, ki vključuje več varnostnih vlog. **[P,V]**
[Mohino et al., 2019]	Teoretičen z anketo	Secure Software Development Life Cycle (S-SDLC) predlaga številne dodatne procese, artefakte in vloge skozi na vsakem koraku razvojnega procesa. **[P,A,V]**
[Firdaus et al., 2014]	Študija primera med študenti	Secure Feature Driven Development (SFDD) gradi na znanem pristopu Feature Driven Development (FDD) z osredotočanjem na varnost vseh elementov v posameznih fazah. Metoda dodaja dve dodatni varnostni fazi in predlaga uvedbo skrbnika za varnost (security master). **[P,A,V]**
[Giacalone et al., 2014]	Študija primera v industrij- skem okolju	Predlagana metoda vključuje dva glavna procesa: (1) varnostno raziskovanje (security survey) zagotavlja celovito karakterizacijo informacijsko-komunikacijskih tehnologij in poslovnih storitev lastnika (naročnika); (2) triaža varnosti (security triage) za identifikacijo ravni ustreznosti zahtev za oceno varnosti. **[P]**
[Ionita et al., 2019]	Študija primera v industrij- skem okolju	Predlagani pristop uvaja identifikacijo, določanje prednosti in izvajanje varnostnih zahtev v treh fazah: (i) ocena tveganja, (ii) določitev prednostnih nalog varnostnih zahtev in (iii) dopolnjevanje seznama zahtev (product backlog). **[P,A]**
[Koc et al., 2019]	Anketa	Trustworthy Scrum v klasičen Scrum vnaša z varnostjo povezane aktivnosti v vsak sprint (npr. statična analiza, pregled kode). Poleg tega predlaga tudi dodaten, v varnost usmerjen sprint.**[P]**
[Maier et al., 2017]	Anketa	Secure Scrum v celoten proces vpeljuje številne prakse v varnost usmerjenega razvoja, ki jih predlagajo ISO standardi. **[P,A]**
[Maria et al., 2015]	Študija primera v akademskem okolju	ScrumS dodaja tehniko Secure Project za Scrum. Tehnika predvideva dodajanje številnih dodatnih elementov v obstoječo metodologijo (npr. varnostne uporabniške zgodbe, analiza tveganj itd.). **[P,A]**
[Nguyen and Dupuis, 2019]	Teoretičen	Technology Development Lifecycle je osnovan na ogrodju DevSecOps in skuša vzdrževati hitre povratne informacije med vsemi deležniki v razvojnem procesu. **[P]**
[Othmane et al., 2014]	Študija primera v industrijskem okolju	Pristop gradi na razširanju posameznih faz v razvojnem procesu z dodajanjem različnih z varnostjo povezanih aktivnosti. **[P,A]**
[Rindell et al., 2015]	Teoretičen	Predlagani model dodaja varnostno-specifične prilagoditve in dodatke vlogam, procesom in artefaktom v klasični metodi Scrum. **[P,A,V]**
[Singhal and Singhal, 2011]	Teoretičen	Pristop Agile Security Framework v razvojni proces vpeljuje varnost v vsaki fazi življenjskega cikla skozi uvajanje varnostnih elementov, kot so zgodbe zlorab (abuser stories), izobraževanje o varnosti in kategori- ziranje varnostnih izahtev.**[P,A]**
[Singh, 2018]	Teoretičen	Predlagani model se osredotoča na naslavljanje avtentikacije in preverbe v procesu razvoja programske opreme. **[P]**
[Yu and Le, 2012]	Teoretičen	Metoda SQUARE-R gradi na metodi SQUARE pri čemer vnaša neprekinjeno obvladovanje tveganj. Tako zagotavlja uresničevanje varnostnih poslovnih ciljev hkrati.**[P]**
[Stålhane and Johnsen, 2017]	Teoretičen	SafeScrum prilagaja Scrum tako, da seznam zahtev razdeli na dva dela (standardni in varnostni) in spremeni analizo vpliva za uspešno upravljanje z zahtevami. **[P,A]**
[Tøndel et al., 2019]	Teoretičen	Sestanki Security Intention Recap Meetings so oblikovani za evalviranje trenutnih praks povezanih z varnostnimi namerami v določenem projektu in sprejemanje odločitev kako naslavljati probleme. **[P]**

Tabela 4: **Predlagani pristopi, ki izvirajo iz pregleda z metodo snežne kepe. Črke v oglatih oklepajih nakazujejo kateri tip predlaganih elementov je dominanten (P – procesi, A – artefakti, V – vloge).**

Vir	Metoda	Kratek opis
[Williams and Meneely, 2010]	Teoretičen	Protection poker je pristop z igrifikacijskimi elementi in temelji na Wideband Delphi metodi. **[P]**
[Rygge and Jøsang, 2018]	Teoretičen	Pristop Threat poker je predlagan po vzoru prej omenjenega Protection poker z dodano kompleksnostjo. **[P]**
[Pohl and Hof, 2015]	Kvazieksperiment v akademskem okolju	Secure Scrum je razširitev tradicionalne Scrum metode. Predlaga štiri dodatne aktivnosti: (i) identifikacija varnostnih vprašanj, (ii) uvajanje varnostnih komponent, (iii) preverjanje / pregled in (iv) definicija končane zadeve (definition of done). **[P]**
[Siiskonen et al., 2014]	Teoretičen	Generic security user stories so vnaprej pripravljene zgodbe, ki jih lahko razvojne ekipe uporabijo tudi v pri- meru, ko naročnik nima zadostnega znanja, da bi var- nostne komponente eksplicitno zahteval. Predlagane varnostne zgodbe so nato skupinjene v večje enote imenovane varnostne teme. **[A]**
[Daud, 2010]	Teoretičen	Secure Software Lifecycle (S-SL) predvideva uvajanje varnostnih elementov na vsaki stopnji življenjskega cikla razvoja programske opreme z (i) varnostno ana lizo, (ii) varnostnim načrtovanjem in izvajanjem ter (iii) varnostnim testiranjem. **[P]**
[Mougouei et al., 2013]	Teoretičen	S-Scrum gradi na dveh novih aktivnostih, in sicer Security Spikes: (i) konica za varnostno analizo (ii) konica za varnostno modeliranje. Konkretne tehnike niso predlagane. [P]

Ionita et al., 2019] in nato kombinacija vseh treh kategorij elementov (13%) v obliki celostnih rešitev [Mohino et al., 2019, Firdaus et al., 2014, Rindell et al., 2015]. Najmanjši skupini sta skupina, ki se osredotoča samo na artefakte (9%) in kombinacija procesov in vlog (4%). Pristopov, ki bi bili usmerjeni le v dodajanje novih vlog nismo zasledili.

Odgovor na RV2: Med identificiranimi pristopi je le pet pristopov (21,7 %) empirično preverjenih s študijo primera v realnem industrijskem okolju [Ghani et al., 2014, Baca et al., 2015, Giacalone et al., 2014, Ionita et al., 2019], trije pristopi (13,1 %) pa so bili preverjetni v akademskem okolju s pomočjo študentov [Firdaus et al., 2014, Maria et al., 2015, Pohl and Hof, 2015]. Nadaljnji trije pristopi (13,1 %) so bili predmet empirične raziskave s pomočjo ankete [Mohino et al., 2019, Koc et al., 2019, Maier et al., 2017]. Več kot polovica predlaganih pristopov (52,1 %) ni bila empirično preverjena.

Odgovor na RV3: Vsi identificirani pristopi temeljijo na predpostavki, da varnost ne velja za nepogrešljivo kakovost programske opreme, ker ni sestavni del posamezne metode. Zato avtorji teh pristopov predlagajo, da se varnostni elementi stalno vključijo v postopek razvoja programske opreme. Identificirane pristope je mogoče razvrstiti v tri skupine. Pristope, ki se osredotočajo na: (a) dvig motivacije za razvoj varne programske opreme, (b) povečevanje znanja s področja varnosti in (c) prilagajanje metode razvoja programske opreme z varnostno-specifičnimi elementi.

Glede na zorni kot s katerega rešujejo problem razvoja varne programske opreme, lahko pristope razde- limo v tri skupine: dodajanje varnostnih elementov z namenom (a) dvigovanja *motivacije*, (b) vključevanje *varnostno-specifičnega znanja* in (c) *ostali varnostno-specifični elementi*. *Motivacija* je bila naslovljena z igri- fikacijo, kot sta *Protection poker* [Williams and Meneely, 2010] in *Threat poker* [Rygge and Jøsang, 2018], oba navdihnjena po pristopu *Planning poker*, na konseznu temelječi tehniki ocenjevanja težavnosti nalog v agilnih projektih [Grenning, 2002]. Oba pristopa temeljita na ideji vključevanja v obstoječe agilne metode in vključujeta oceno tveganj za varnost in zasebnost, ki ju določata enostavnost izvedbe napada in njegova resnost. Vključitev *varnostno-specifičnega znanja* je predlagalo več avtorjev, najpogosteje z dodaja- njem novih vlog, povezanih z varnostjo [Baca et al., 2015]. Te vloge so običajno skrbnik za varnost (angl. *Security master*) [Azham et al., 2011], varnostni strokovnjak (angl. *Security expert*) [Musa et al., 2011] ali penetracijski tester (angl. *Penetratioin tester*) [Tomanek and Klima, 2015]. Naloga varnostnih strokovnjakov je prepoznavanje in zmanjševanje varnostnih

tveganj, upravljanje varnostnih pomanjkljivosti, in izvajanje varnostnih in penetracijskih testiranj. Najbolj celovit pristop so predlagali [Baca et al., 2015] in vključuje niz varnostnih vlog, ki tvorijo varnostno skupino, sestavljeno iz štirih varnostnih vlog z različ- nimi pristojnostmi (tj. tehničnimi, netehničnimi in pravnimi). Najpogostejši predlagani *varnostno--specifični elementi* so varnostne uporabniške zgodbe (kot prilagoditev primerov zlorabe (angl. *Misuse and abuse cases*) iz tradicionalne metode razvoja varne programske opreme [Lee and Park, 2016]) ali njihove razli- čice, ki zagotavljajo pregled nad varnostnimi zahtevami in so lahko vključene v v obstoječi seznam zahtev (angl. *Product backlog*) ali pa v posebni in namenski seznam varnostnih zahtev (anlg. *Security backlog*) [Azham et al., 2011, Mougouei et al., 2013, Siiskonen et al., 2014].

Trenutna literatura ne ponuja nedvoumne in celovite rešitve za razvoj varne programske opreme z agil- nimi metodami, primernimi za manjša podjetja z omejenimi proračuni [Siiskonen et al., 2014]. Obstoječi pristopi imajo več problemov. Prvič, predlagajo niz varnostnih elementov (vloge, procesi, artefakti), ki jih je treba trajno vključiti v obstoječo agilno metodo [Baca et al., 2015]. Pogosta težava takšnih pristopov je, da ogrožajo produktivnost, agilnost in znatno povečajo stroške projektov razvoja programske opreme [Boström et al., 2006]. Drugič, ne upoštevajo situacijskih dejavnikov, kot so obstoječa agilna metoda, medo- sebni odnosi znotraj razvojne skupine in podjetja ter drugih dejavnikov, ki skladno s kontingenčno teorijo pomembno vplivajo na uspešnost projekta [Fiedler, 1964, Song et al., 2018]. Tretjič, varnostni elementi so zasnovani za točno določeno metodo (npr. Scrum ali XP), kar razvojno podjetje omeji na metodo, za katero so bili zasnovani [Ghani et al., 2014, Türpe and Poller, 2017, Azham et al., 2011]. Č etrtič, igrifikacija kot pristop k dvigu motivacije znotraj razvojne skupine lahko po daljših obdobjih uporabe povzroči stres in napetost v delovnem procesu [Platonova and Be¯ rziša, 2017]. Petič, da bi odpravili pomanjkanje znanja v razvojni skupini, opredeljeni pristopi predlagajo vključitev novih varnostnih vlog, običajno z zaposlitvijo dragih strokovnjakov za varnost [Baca et al., 2015].

Glede na ugotovitve, ki izhajajo iz sistematičnega pregleda literature in predvsem identificirane prednosti in pomanjkljivosti omenjenih pristopov, bi bilo smiselno iskati trajno rešitev, ki bi bila dosežena z le začasno prilagoditvijo agilne metode. Idealno bi morala biti predlagana rešitev prilagodljiva različnim agilnim metodam, ki jih podjetja uporabljajo. Takšno rešitev deloma ponujajo pristopi, ki uvajajo varnostno znanje skozi različne varnostne vloge (npr. [Azham et al., 2011, Musa et al., 2011, Tomanek and Klima, 2015]). Vendar temeljno pomanjkljivost tovrstnih pristopove predstavlja tudi do 500-odstotno povišanje stroškov projekta [Baca et al., 2015]. Postopno pridobivanje znanja in s tem tudi ozaveščenosti o varnosti bi bilo tako smiselno delegirati na razvojno ekipo. Začasni značaj te rešitve bi bil dosežen z implementacijo pristopa do trenutka, ko ravni varnostne ozaveščenosti in znanja razvojne ekipe ne dosežejo zadovoljive ravni. Ponovitve bi bile potrebne le za ohranitev želene ravni varnostnega znanja. Ta zasnova bi torej (a) omogočala izogibanje zaposlovanju dragih strokovnjakov za varnost, ki so običajno zadolženi za vnašanje varnostnega znanja, in (b) reševala problem togosti in višanja stroškov obstoječih pristopov, ker jih je v obstoječo agilno metod treba vključiti trajno. Poleg tega bi na podlagi predpostavke, da varnostno znanje in varnostna zavest višata motivacijo, (c) naslavljal vprašanje motivacije brez elementov igrifikacije, ki po daljši uporabi lahko v ekipo vnaša stres in napetost [Platonova and Be¯ rziša, 2017]. Takšen pristop bi predstavljal razmeroma trajno rešitev z začasnim dodajanjem elementov k obstoječi metode, pri čemer ne bi postal sestavni del obstoječe metode.

4 OMEJITVE IN NADALJNJE DELO

Kot vsaka druga ima tudi ta raziskava omejitve. Prvič, osredotočili smo se na štiri glavne bibliografske zbirke podatkov, pri čemer je nekaj manjših ostalo nepregledanih oz. so bile pregledane le posredno, skozi dela, vključena v sistematični pregled literature. Drugič, pregled je opravil en raziskovalec, kar izpostavlja možnost raziskovalčeve pristranskosti. Tretjič, pregled je bil opravljen med deli, objavljenimi od leta 2009, kar omeji širino pregleda na določeno časovno obdobje. Tako so dela, objavljena pred tem datumom, iz predgleda izpuščena.

V nadaljnjem delu bi se bilo smiselno osredotočiti na analizo prednosti in slabosti vsake od identificiranih metod, kar bi rezultiralo v bolj poglobljenem pregledu dejanskega stanja. Smiselno bi bilo tudi razširiti časovni okvir (npr. od leta 2001, ko so se agilni pristopi začeli pojavljati) in razširiti iskalni niz,

kar bi omogočalo bolj celovit pregled področja. Ob upoštevanju širšega pregleda bi bilo treba predlagane pristope razvrstiti v časovne okvirje (npr. petletna obdobja), kar bi omogočalo umestitev posameznega pristopa v določen socio-tehnološki kontekst.

5 ZAKLJUČEK

Izvedli smo sistematični pregled literature agilnih in vitkih pristopov k razvoju varne programske opreme. Pregled se je osredotočil na relativno široko časovno obdobje in temeljil na razmeroma širokem iskalnem nizu. S tem smo naslavljali odprta vprašanja preteklih del. Rezultati nakazujejo, da si obstoječi pristopi delijo skupne pomanjkljivosti. Predvsem je večina del teoretičnih, testiranih v akademskem okolju ali empiričnih z anketo. Popolnoma razumljivo je, da je testiranje pristopov v industrijskem okolju težavno, pa vendar le takšno testiranje lahko poda vpogled v izvedljivost in učinkovitost pristop ter tiste posledice, ki niso bile pričakovane pri teoretičnem modeliranju pristopa ali njegovem testiranju v kontroliranem okolju.

Identificirane pomanjkljivosti na področju agilnih in vitkih pristopov k razvoju varne programske opreme usmerjeno kliče k iskanju rešitve, ki ne bi bila trajno vključena v razvojni proces. Tak pristop bi lahko ponudil dolgotrajno rešitev, obenem pa obetal učinkovitost, brez znatnega povečanja stroškov ali ogrožanja agilnosti metode, ki jo razvojna skupina že uporablja.

LITERATURA

[1] [Adelyar and Norta, 2016] Adelyar, S. H. and Norta, A. (2016). Towards a Secure Agile Software Deve- lopment Process. In *10th International Conference on the Quality of Information and Communications Technology (QUATIC)*, pages 101–106.

[2] [Allison et al., 2020] Allison, I., Tiplitsky, J., Kennedy, S., Kersten, N., Lietz, S., Lim, F., Nikulshin, M., Price, C., Dhungel, R., Rose, K., and Sherman, B. (2020). The DevSecOps Manifesto.

[3] [Azham et al., 2011] Azham, Z., Ghani, I., and Ithnin, N. (2011). Security backlog in scrum security practices. *2011 5th Malaysian Conference in Software Engineering, MySEC 2011*, pages 414–417.

[4] [Baca et al., 2015] Baca, D., Boldt, M., Carlsson, B., and Jacobsson, A. (2015). A novel security-enhanced agile software development process applied in an industrial setting. *10th International Conference on Availability, Reliability and Security*, pages 11–19.

[5] [Barbosa and Sampaio, 2017] Barbosa, D. A. and Sampaio, S. (2017). Guide to the Support for the En- hancement of Security Measures in Agile Projects. In *Brazilian Workshop on Agile Methods*, pages 25–31.

[6] [Bezdedeanu, 2019] Bezdedeanu, A. (2019). DevSecOps is Not a Role or Technology: It's a Culture to Wholly Embrace.

[7] [Beznosov and Kruchten, 2005] Beznosov, K. and Kruchten, P. (2005). Towards agile security assurance. In *Proceedings New Security Paradigms Workshop*, pages 47–54.

[8] [Bishop and Rowland, 2019] Bishop, D. and Rowland, P. (2019). Agile and Secure Software Development: An Unfinished Story. *Issues in Information Systems*, 20(1):144–156.

[9] [Boström et al., 2006] Boström, G., Wäyrynen, J., Bodén, M., Beznosov, K., and Kruchten, P. (2006). Extending XP practices to support security requirements engineering. *2006 international workshop on Software engineering for secure systems,* page 11.

[10] [Curcio et al., 2018] Curcio, K., Navarro, T., Malucelli, A., and Reinehr, S. (2018). Requirements engineering: A systematic mapping study in agile software development. *Journal of Systems and Software*, 139(1):32– 50.

[11] [Daud, 2010] Daud, M. I. (2010). Secure software development model: A guide for secure software life cycle. In *Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010*, pages 724–728, Hong Kong.

[12] [Fiedler, 1964] Fiedler, F. E. (1964). A theory of leadership effectiveness. In Berkowitz, L., editor, *Advances in experimental social psychology.* Academic Press, New York.

[13] [Firdaus et al., 2014] Firdaus, A., Ghani, I., and Jeong, S. R. (2014). Secure Feature Driven Development (SFDD) Model for Secure Software Development. In *Procedia - Social and Behavioral Sciences,* volume 129, pages 546–553. Elsevier B.V.

[14] [Ghani et al., 2014] Ghani, I., Azham, Z., and Jeong, S. R. (2014). Integrating software security into agile-Scrum method. *Transactions on Internet and Information Systems*, 8(2):646–663.

[15] [Giacalone et al., 2014] Giacalone, M., Paci, F., Mammoliti, R., Perugino, R., Massacci, F., and Selli, C. (2014). Security Triage: An Industrial Case Study on the Effectiveness of a Lean Methodology to Identify Security Requirements Matteo. In *Symposium on Empirical Software Engineering and Measurement,* pages 1–8.

[16] [Grenning, 2002] Grenning, J. (2002). Planning poker or how to avoid analysis paralysis while release planning.

[17] [Gwanhoo and Weidong, 2010] Gwanhoo, L. and Weidong, X. (2010). Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data. *MIS Quarterly*, 34(1):87–114.

[18] [Inayat et al., 2015] Inayat, I., Salim, S. S., Marczak, S., Daneva, M., and Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior*, 51:915–929.

[19] [Ionita et al., 2019] Ionita, D., Van Der Velden, C., Ikkink, H.-j. K., and Eelko, N. (2019). Towards Risk- Driven Security Requirements Management in Agile Software Development. *Lecture Notes in Business Information Processing*, 350(628):133–144.

[20] [Jyothi and Rao, 2011] Jyothi, V. E. and Rao, K. N. (2011). Effective Implementation of Agile Practices Ingenious and Organized Theoretical Framework. *International Journal of Advanced Computer Science and Applications*, 2(3):41–48.

[21] [Kasauli et al., 2018] Kasauli, R., Knauss, E., Kanagwa, B., Nilsson, A., and Calikli, G. (2018). Safety-critical systems and agile development: A mapping study. In *44th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2018*, pages 470–477. IEEE.

[22] [Khan and Ikram, 2017] Khan, N. F. and Ikram, N. (2017). Security requirements engineering: A systematic mapping (2010-2015). In *2016 International Conference on Software Security and Assurance*, pages 31–36.

[23] [Kiuwan, 2019] Kiuwan (2019). The Benefits of a DevSecOps Approach to the SDLC.

[24] [Koc et al., 2019] Koc, G., Aydos, M., and Tekerek, M. (2019). Evaluation of Trustworthy Scrum Emplo- yment for Agile Software Development based on the Views of Software Developers. In *4th International Conference on Computer Science and Engineering*, pages 63–67.

[25] [Lee, 2018] Lee, J. S. (2018). The DevSecOps and Agency Theory. In *29th IEEE International Symposium on Software Reliability Engineering Workshops, ISSREW 2018*, pages 243–244. IEEE.

[26] [Lee and Park, 2016] Lee, K. H. and Park, Y. B. (2016). Adaption of integrated secure guide for secure software development lifecycle. *International Journal of Security and its Applications*, 10(6):145–154.

[27] [Lwakatare et al., 2016] Lwakatare, L. E., Kuvaja, P., and Oivo, M. (2016). Relationship of DevOps to Agile, Lean and Continuous Deployment: A Multivocal Literature Review Study. In Abrahamsson, P., Jedlitschka, A., Nguyen, D. A., Felderer, M., Amasaki, S., and Mikkonen, T., editors, *Lecture Notes in Computer Science*, volume 10027, pages 198–214. Springer, Cham.

[28] [Maier et al., 2017] Maier, P., Ma, Z., and Bloem, R. (2017). Towards a Secure SCRUM Process for Agile Web Application Development. In *Proceedings of the 12th International Conference on Availability, Reliability and Security - ARES '17*, pages 1–8.

[29] [Maria et al., 2015] Maria, R. E., Rodrigues, L. A., and Pinto, N. A. (2015). ScrumS - A model for safe agile development. In *7th International ACM Conference on Management of Computational and CollEctive Intelligence in Digital EcoSystems, MEDES 2015*, pages 43–47.

[30] [Mellado et al., 2010] Mellado, D., Blanco, C., Sánchez, L. E., and Fernández-Medina, E. (2010). A syste- matic review of security requirements engineering. *Computer Standards and Interfaces*, 32(4):153–165.

[31] [Mohino et al., 2019] Mohino, J. d. V., Higuera, J. B., Higuera, J. R. B., and Montalvo, J. A. S. (2019). The application of a new secure software development life cycle (S-SDLC) with agile methodologies. *Electronics (Switzerland)*, 8(11):1–28.

[32] [Mougouei et al., 2013] Mougouei, D., Sani, N. F. M., and Almasi, M. M. (2013). S-Scrum : a Secure Methodology for Agile Development of Web Services. *World of Computer Science and Information Technology Journal (WSCIT)*, 3(1):15–19.

[33] [Musa et al., 2011] Musa, S. B., Norwawi, N. M., Selamat, M. H., and Sharif, K. Y. (2011). Improved extreme programming methodology with inbuilt security. *ISCI 2011 - 2011 IEEE Symposium on Computers and Informatics*, pages 674–679.

[34] [Myrbakken and Colomo-Palacios, 2017] Myrbakken, H. and Colomo-Palacios, R. (2017). DevSecOps: A Multivocal Literature Review. In Mas, A., Mesquida, A., and O'Connor, R., editors, *Communications in Computer and Information Science*, volume 770, pages 30–42. Springer, Cham.

[35] [Nguyen and Dupuis, 2019] Nguyen, J. and Dupuis, M. (2019). Closing the feedback loop between UX design, software development, security engineering, and operations. In *Proceedings of the 20th Annual Conference on Information Technology Education*, pages 93–98.

[36] [Othmane et al., 2014] Othmane, L., Angin, L., Weffers, H., and Bhargava (2014). Extending the Agile Development Approach to Develop Acceptably Secure Software. *IEEE Transactions on Dependable and Secure Computing*, 11(6):497–509.

[37] [Oueslati et al., 2015] Oueslati, H., Rahman, M. M., and ben Othmane, L. (2015). Literature Review of the Challenges of Developing Secure Software Using the Agile Approach. In *10th International Conference on Availability, Reliability and Security*, pages 540–547.

[38] [Platonova and Be¯rziša, 2017] Platonova, V. and Be¯rziša, S. (2017). Gamification in Software Development Projects. *Information Technology and Management Science*, 20(1):58–63.

[39] [Pohl and Hof, 2015] Pohl, C. and Hof, H.-J. (2015). Secure Scrum: Development of Secure Software with Scrum. In *The Ninth International Conference on Emerging Security Information, Systems and Technologies Secure*, pages 15–20.

[40] [Riisom et al., 2018] Riisom, K. R., Hubel, M. S., Alradhi, H. M., Nielsen, N. B., Kuusinen, K., and Jabangwe, R. (2018). Software security in agile software development: A literature review of challenges and solutions. In *ACM International Conference Proceeding Series*, pages 1–5.

[41] [Rindell et al., 2015] Rindell, K., Hyrynsalmi, S., and Leppänen, V. (2015). Securing scrum for VAHTI. In *CEUR Workshop Proceedings*, pages 236–250.

[42] [Rindell et al., 2017] Rindell, K., Hyrynsalmi, S., and Leppänen, V. (2017). Busting a myth: Review of agile security engineering methods. In *ACM International Conference Proceeding Series,* pages 1–10.

[43] [Rygge and Jøsang, 2018] Rygge, H. and Jøsang, A. (2018). Threat Poker : Solving Security and Privacy Threats in Agile Software Development. (November):1–15.

[44] [Siiskonen et al., 2014] Siiskonen, T., Sars, C., Vah Sipila, A., and Pietikain, A. (2014). Generic Security User Stories. In Pietikinen, P. and Rning, J., editors, *Handbook of the Secure Agile Sotware Development Life Cycle*, chapter 9, pages 9–14. University of Oulu, Oulu.

[45] [Singh, 2018] Singh, A. (2018). Integrating the Extreme Programing Model with Secure Process for Re- quirement Selection. In *2nd International Conference on Electronics, Communication and Aerospace Technology*, pages 423–426.

[46] [Singhal and Singhal, 2011] Singhal, S. and Singhal, A. (2011). Development of Agile Security Framework Using a Hybrid Technique for Requirements Elicitation. In Unnikrishnan, S., Surve, S., and Bhoir, D., editors, *Advances in Computing, Communication and Control*, pages 178–188.

[47] [Song et al., 2018] Song, M., Wang, P., and Yang, P. (2018). Promotion of secure software development assimilation. *Chinese Management Studies,* 12(1):164–183.

[48] [Stålhane and Johnsen, 2017] Stålhane, T. and Johnsen, S. O. (2017). Resilience and safety in agile development. In *27th European Safety and Reliability Conference*, pages 945–954.

[49] [Tomanek and Klima, 2015] Tomanek, M. and Klima, T. (2015). Penetration Testing in Agile Software Development Projects. *International Journal on Cryptography and Information Security,* 5(1):01–07.

[50] [Tøndel et al., 2019] Tøndel, I. A., Cruzes, D. S., Jaatun, M. G., and Rindell, K. (2019). The Security Intention Meeting Series as a way to increase visibility of software security decisions in agile development projects. In *International Conference on Availability, Reliability and Security*, pages 1–8, Canterbury. ACM Press.

[51] [Tøndel and Jaatun, 2020] Tøndel, I. A. and Jaatun, M. G. (2020). Towards a Conceptual Framework for Security Requirements Work in Agile Software Development. *International Journal of Systems and Software Security and Protection*, 11(1):33–62.

[52] [Türpe and Poller, 2017] Türpe, S. and Poller, A. (2017). Managing security work in scrum: Tensions andchallenges. *CEUR Workshop Proceedings*, 1977(SecSE):34–49.

[53] [Villamizar et al., 2018] Villamizar, H., Kalinowski, M., Viana, M., and Fernández, D. M. (2018). A systematic mapping study on security in agile requirements engineering. In 44th *Euromicro Conference on Software Engineering and Advanced* Applications, pages 454–461.

[54] [Williams and Meneely, 2010] Williams, L. and Meneely, A. (2010). Protection poker: The new software security »game«. *IEEE Security & Privacy*, 8(3):pp. 14–20.

[55] [Yu and Le, 2012] Yu, W. D. and Le, K. (2012). Towards a secure software development lifecycle with SQUARE+R. In *International Computer Software and Applications Conference*, pages 565–570.

▪

Anže Mihelič je doktorski kandidat na Univerzi v Ljubljani, Fakulteti za računalništvo in informatiko ter Pravni fakulteti. Kot asistent je zaposlen na Univerzi v Mariboru, Fakulteti za varnostne vede, kot raziskovalec pa na Univerzi v Hagnu, Fakulteti za matematiko in računalništvo. Za svoje raziskovalno delo je prejel nagrado Fakultete za varnostne vede in rektorjevo nagrado Univerze v Mariboru. Bil je predsednik organizacijskega odbora mednarodne konference Central European Cybersecurity Conference 2019, ki se je odvijala v Münchnu. Prav tako je sodeloval in sodeluje pri nacionalnih in mednarodnih projektih s področja informacijske in kibernetske varnosti. Njegovi raziskovalni interesi obsegajo tehnične, zasebnostne, pravne ter psihološke vidike informacijske in kibernetske varnosti.

▪

Simon Vrhovec je docent na Fakulteti za varnostne vede Univerze v Mariboru. Leta 2015 je doktoriral na Fa- kulteti za računalništvo in informatiko Univerze v Ljubljani. V letih 2018 in 2019 je sopredsedoval mednarodni konferenci Central European Cybersecurity Conference (CECC),od leta 2019 pa je član usmerjevalnega od- bora European Interdisciplinary Cybersecurity Conference (EICC). Je član uredniškega odbora znanstvenih revij Journal of Cyber Security and Mobility, International Journal of Cyber Forensics and Advanced Threat Investigations in EUREKA: Social and Humanities. Bil je oz. je gostujoči urednik v znanstvenih revijah IEEE Security Privacy, Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications in Journal of Universal Computer Science. Njegova glavna raziskovalna področja so človeški dejavniki v kibernetski varnosti, razvoj varne programske opreme, agilne metode, odpor do sprememb in zdravstvena informatika.

▪

Tomaž Hovelja je doktoriral iz organizacije in managementa na Ekonomski fakulteti Univerze v Ljubljani. Zaposlen je kot izredni profesor na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Njegova raziskovalna področja so družbeni, gospodarski in organizacijski dejavniki uvajanja IT v podjetja in uspešnost IT projektov. Objavlja v revijah, kot so Business information systems engineering, International journal of project management, International journal of engineering education, Assessment evaluation in higher education

Iz Islovarja

Islovar je spletni terminološki slovar informatike, ki ga objavlja jezikovna sekcija Slovenskega društva INFORMATIKA in ga najdete na naslovu http://www.islovar.org. Tokrat objavljamo izbor izrazov iz zbirke »objekt«. Vabimo vas, da tudi vi prispevate svoje pripombe, predloge ali nove izraze.

dédovanje -a s (angl. inheritance system, inheritance) značilnost nekaterih objektnih jezikov, v katerih lahko objekt ali razred prevzame značilnosti in metode (2) nadrejenih objektov ali razredov; prim. enkratno dedovanje, večkratno dedovanje

deklarácija -e ž (angl. declaration) opredelitev lastnosti identifikatorja v programiranju

inkapsulácija -e ž (angl. encapsulation) 1. združevanje podatkov in metode (2) za njihovo upravljanje; sin. ovijanje 2. dodajanje prometnih podatkov k podatkovnemu bloku; sin. ovijanje

konstrúktor -rja m (angl. constructor) metoda (2) za ustvarjanje objekta v objektnem programiranju

metóda -e ž (angl. method) 1. postopki, pravila za izvajanje določene naloge 2. ukaz (1), procedura v objektnem programiranju, lastna enemu objektu ali več objektom

nádrázred -éda m (angl. superclass, basic class, parent class) razred, iz katerega so izpeljani drugi razredi, ki podedujejo njegove lastnosti in metode (2); prim. podrazred

objékt -a m (angl. object) entiteta v objektnem programiranju, ki vsebuje podatke in metode

objéktni jezik -ega -íka m (angl. object oriented language) programski jezik, ki podpira objekte z identiteto, vgrajenimi lastnostmi in metodami (2), npr. java

objéktni podátkovni modél -ega -ega -a m (angl. object oriented data model) podatkovni model, ki temelji na objektih in metodah (2), npr. CORBA

objéktno povezovánje in vgrajevánje -ega -- -- -- s (angl. object linking and embedding, OLE) vključitev in povezava zunanjega objekta v program ali dokument; sin. OLE

objéktno programiranje -ega -a s (angl. object oriented programming) programiranje, pri katerem so koncepti realnega sveta predstavljeni z objekti, razredi in metodami (2); sin. objektno usmerjeno programiranje

pót do razrédov -í -- -- ž (angl. classpath) parameter (2), ki določa lokacijo uporabniško definiranih razredov

prekrívanje2 -a s (angl. overriding) redefinicija podedovanih metod pri objektno usmerjenem programiranju

preobláganje -a s (angl. overloading) deklaracija več metod (2) z enakim imenom, a drugačnimi parametri (2) pri objektnem programiranju

primérek -rka m *(angl. instance)* konkretna izvedba objekta v razredu

prototípno programíranje -ega -a *(angl. prototype-oriented programming, prototypal programming, prototype-based programming, instance-based programming, classless programming)* objektno programiranje, pri katerem se dedovanje izvede s kloniranjem obstoječih objektov, ki služijo kot prototipi

rázred -éda m *(angl. class)* predloga za ustvarjanje objektov z enakim naborom spremenljivk, metod (2) v objektnem programiranju

Izpitni centri ECDL

ECDL (European Computer Driving License), ki ga v Sloveniji imenujemo evropsko računalniško spričevalo, je standardni program usposabljanja uporabnikov, ki da zaposlenim potrebno znanje za delo s standardnimi računalniškimi programi na informatiziranem delovnem mestu, delodajalcem pa pomeni dokazilo o usposobljenosti. V Evropi je za uvajanje, usposabljanje in nadzor izvajanja ECDL pooblaščena ustanova ECDL Fundation, v Sloveniji pa je kot član CEPIS (Council of European Professional Informatics) to pravico pridobilo Slovensko društvo INFORMATIKA. V državah Evropske unije so pri uvajanju ECDL močno angažirane srednje in visoke šole, aktivni pa so tudi različni vladni resorji. Posebno pomembno je, da velja spričevalo v 148 državah, ki so vključene v program ECDL. Doslej je bilo v svetu izdanih že več kot 11,6 milijona indeksov, v Sloveniji več kot 17.000, in podeljenih več kot 11.000 spričeval. Za izpitne centre v Sloveniji je usposobljenih osem organizacij, katerih logotipe objavljamo.

AGORA

LJUDSKA UNIVERZA MURSKA SOBOTA

ŠOLSKI CENTER Novo mesto

spin

ISER

LJUDSKA UNIVERZA CELJE

ACADEMIA

Micro Team

Znanstveni prispevki

Strokovni prispevki

Kratki znanstveni prispevki

Informacije

ISSN 1318-1882
9 771318 188001