



04 U P O R A B N A INFORMATIKA

2024 < ŠTEVILKA 4 < LETNIK XXXII < ISSN 1318-1882

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 pooblaščen 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 v program certificiranja ECDL vključenih že preko 16 milijonov oseb, ki so uspešno opravile preko 80 milijonov izpitov in pridobile ustrezne certificate. V Sloveniji je bilo doslej v program certificiranja ECDL vključenih več kot 18.000 oseb in opravljenih več kot 92.000 izpitov. V Sloveniji sta akreditirana dva izpitna centra ECDL, ki imata izpostave po vsej državi.



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

2024 ŠTEVILKA 4 OKT/NOV/DEC LETNIK XXXII ISSN 1318-1882

Znanstveni prispevki

Yauhen Unuchak, Mirjana Kljajić Borštnar, Tatyana Unuchak
Uporaba dnevniških zapisov v namen profiliranja obremenitev spletne aplikacije 155

Strokovni prispevki

Gal Gantar, Matevž Pesek
Analiza napadov na pametne pogodbe osnovane na okoljih ethereum virtualnega stroj 170

Maksim Nikitashin
Umetna inteligenca in razvoj uporabniških vmesnikov: Študija primera mobilne aplikacije X 180

Pregledni znanstveni prispevki

Staša Blatnik, Andreja Pucihar, Mirjana Kljajić Borštnar
Spodbude in ovire uporabe odprtih podatkov v podjetjih – pregled literature 188

Informacije

Iz Islovarja 204

Ustanovitelj in izdajatelj

Slovensko društvo INFORMATIKA
Litostrojska cesta 54, 1000 Ljubljana

Predstavniki

Slavko Žitnik

Odgovorni urednik

Mirjana Kljajić Borštnar

Uredniški odbor

Andrej Kovačič, Anton Manfreda, Evelin Krnac, Jan Mendling, Jan von Knop, John Taylor, Lili Nemeč Zlatolas, Marko Hölbl, Miodrag Popović, Mirjana Kljajić Borštnar, Mirko Vintar, Pedro Simões Coelho, Saša Divjak, Sjaak Brinkkemper, Stevanče Nikoloski, Tatjana Welzer Družovec, Timotej Knez, Vesna Bosilj-Vukšič, Vida Groznik, Vladislav Rajkovič

Recenzentski odbor

Alenka Baggia, Alenka Brezavšček, Andrej Brodnik, Andrej Kovačič, Andreja Pucihar, Anton Manfreda, Benjamin Urh, Blaž Rodič, Damjan Fujs, Damjan Strnad, Dejan Lavbič, Denis Trček, Domen Mongus, Drago Bokal, Eva Jereb, Gregor Lenart, Jernej Vičič, Jure Žabkar, Jurij Mihelič, Luka Pavlič, Luka Tomat, Maja Meško, Meja Pušnik, Marina Trkman, Marina Trkman, Marjeta Marolt, Marko Hölbl, Martina Šestak, Matej Klemen, Matevž Pesek, Mirjam Sepesy Maučec, Mirjana Kljajić Borštnar, Mladen Borovič, Muhamed Turkanović, Niko Schlamberger, Ratko Pilipovic, Samed Bajrić, Sandi Gec, Saša Divjak, Stevanče Nikoloski, Tilen Medved, Tina Jukić, Uroš Rajkovič, Živa Rant

Tehnični urednik

Timotej Knez

Lektoriranje angleških izvlečkov

Marvelingua (angl.)

Oblikovanje

KOFEIN DIZAJN, d. o. o.

Prelom in tisk

Boex DTP, d. o. o., Ljubljana

Naklada

110 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).

Izid publikacije je finančno podprla Javna agencija za znanstvenoraziskovalno in inovacijsko dejavnost Republike Slovenije.

© Slovensko društvo INFORMATIKA

Vabilo avtorjem

V reviji Uporabna informatika objavljamo kakovostne izvirne prispevke domačih in tujih avtorjev z najširšega področja informatike, ki se nanašajo tako na poslovanje podjetij, javno upravo, družbo in posameznika. Prispevki so lahko znanstvene, strokovne ali informativne narave, še posebno spodbujamo objavo interdisciplinarnih prispevkov. Zato vabimo avtorje, da prispevke, ki ustrezajo omenjenim usmeritvam, pošljejo uredništvu revije po elektronski pošti na naslov ui@društvo-informatika.si.

Avtorje prosimo, da pri pripravi prispevka upoštevajo navodila, ki so objavljena na naslovu <http://www.uporabna-informatika.si>.

Za kakovost prispevkov skrbi mednarodni uredniški odbor. Prispevki 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 prispevek ponovno prejmejo v pregled. Sprejeti prispevki so pred izidom revije objavljeni na spletni strani revije (predobjava), še prej pa končno verzijo prispevka avtorji dobijo v pregled in potrditev. Uredništvo lahko še pred recenzijo zavrne objavo prispevka, če njegova vsebina ne ustreza vsebinski usmeritvi revije ali če prispevek ne ustreza kriterijem za objavo v reviji.

Pred objavo prispevka mora avtor podpisati izjavo o avtorstvu, s katero potrjuje originalnost prispevka in dovoljuje prenos materialnih avtorskih pravic. 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 pomagali 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 ustreznih priporočamo uporabo spletnega terminološkega slovarja Slovenskega društva Informatika, Islovar (www.islovar.org).

Znanstveni prispevek naj obsega največ 40.000 znakov, kratki znanstveni prispevek do 10.000 znakov, strokovni članki do 30.000 znakov, obvestila in poročila pa do 8.000 znakov.

Prispevek naj bo 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 presledek, pri odstavkih ne uporabljajte zamika.

Naslovu prispevka naj sledi polno ime vsakega avtorja, 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 prispevka. Sledi naj prevod naslova povzetka in ključnih besed v angleškem jeziku. V primeru, da oddajate prispevek v angleškem jeziku, velja obratno. Razdelki naj bodo naslovljeni in oštevilčeni z arabskimi številkami.

Slike in tabele vključite v besedilo. Opremite jih z naslovom in oštevilčite z arabskimi številkami. Na vsako sliko in tabelo se morate v besedilu prispevka sklicevati in jo pojasniti. Če v prispevku 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. Slikam zaslonov se v prispevku izogibajte, 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 IEEE navajanja bibliografskih referenc, v besedilu to pomeni zaporedna številka navajenega vira v oglatem oklepaju (npr. [1]). Na koncu prispevka navedite samo v prispevku uporabljeno literaturo in vire v enotnem seznamu, urejeno po zaporedni številki vira, prav tako v skladu s pravili IEEE. Več o sistemu IEEE, katerega uporabo omogoča tudi urejevalnik besedil Word 2007, najdete na strani https://owl.purdue.edu/owl/research_and_citation/ieee_style/ieee_general_format.html.

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

Uporaba dnevniških zapisov v namen profiliranja obremenitev spletne aplikacije

Yauhen Unuchak, Mirjana Kljajić Borštnar, Tatyana Unuchak
Fakulteta za organizacijske vede, Univerza v Mariboru, Kidričeva cesta 55a, 4000 Kranj
yauhen.unuchak@student.um.si, mirjana.kljajic@um.si, tatyana.unuchak@student.um.si

Izvleček

Zmogljivost spletnih aplikacij je pomembna za učinkovito delovanje in se običajno obravnava s testiranjem z ustreznimi orodji za testiranje zmogljivosti. Običajno se simulirajo pričakovane obremenitve, da bi ocenili delovanje in stabilnost sistema. V tem prispevku predstavljamo novo metodo za podporo profiliranju obremenitev, ki izboljšuje načrtovanje obremenitev spletnih aplikacij z analizo dejanskega vedenja uporabnikov. Na podlagi podatkov iz strežniških dnevnikov in algoritmov strojnega učenja identificiramo uporabniške vzorce, kar omogoča natančnejšo prilagoditev aplikacije realnim potrebam. Ta pristop retrospektivnega testiranja omogoča bolj realistično simulacijo uporabniških dejanj ter optimizacijo uporabe virov. Združevanje ekspertnih predpostavk z analizo dejanske uporabe vodi do bolj zanesljivega načrtovanja in optimizacije infrastrukture, kar izboljša delovanje aplikacij v realnih pogojih. Razvita metoda za podporo profiliranju obremenitev spletnih aplikacij na podlagi dnevniških zapisov omogoča zbiranje, čiščenje in pripravo dnevniških podatkov ter prepoznavanje skupin uporabnikov in njihovega obnašanja.

Ključne besede: testiranje zmogljivosti, profiliranje obremenitev, analiza obnašanja uporabnikov, strojno učenje, dnevniški zapisi spletnega strežnika.

Using Log Records for Web Application Load Profiling

Abstract

The performance of Web applications is important for the efficient operation and is usually addressed by testing with appropriate performance testing tools. Typically, the expected loads are simulated to evaluate system performance and stability. In this paper, we present a new method to support load profiling that improves load planning of web applications by analyzing actual user behaviour. Based on server log data and machine learning algorithms, we identify user patterns that allow us to more accurately adapt the application to real-world needs. This retrospective testing approach allows for more realistic simulation of user actions and optimization of resource usage. Combining expert assumptions with the analysis of actual usage leads to more robust infrastructure design and optimization, improving application performance under realistic conditions. The methodology developed to support log-based load profiling of Web applications enables the collection, cleansing, and preparation of log data as well as the identification of user groups and their behaviour.

Keywords: performance testing, load profiling, user behaviour analysis, machine learning, Web server logs.

1 UVOD

Iz analitičnega poročila družbe Siteefy LLP [1] je razvidno, da v letu 2024 je na internetu registriranih več kot 193 milijonov aktivnih spletnih mest. Spletne

aplikacije so še vedno prevladujoča vrsta aplikacij, saj je do njih mogoče dostopati s katere koli naprave, povezane z internetom, vključno z računalniki, tabličnimi računalniki in pametnimi telefoni, ter z različnimi

operacijskimi sistemi brez potrebe po prilagajanju. Takšne aplikacije zahtevajo manj virov za razvoj, posodabljanje in vzdrževanje v primerjavi z izvornimi aplikacijami za različne platforme (iOS, Android, Windows itd.). Hkrati je spletne aplikacije lažje razširjati, posodabljati in zagotavljati visoko stopnjo razpoložljivosti. Vendar se moramo za ohranjanje visoke ravni zadovoljstva uporabnikov osredotočiti na kakovost teh aplikacij. Na primer, Amazon in Google sta poročala o izgubah med 0,6 in 1,2 %, ko sta zamuda nalaganja strani povečala med 0,4 in 1 sekundo, medtem ko je Shopzilla poročala o 12-odstotnem povečanju prihodkov ob 5-sekundnem skrajšanju časa nalaganja strani [2].

Zagotavljanje visoke kakovosti in zmogljivosti spletnih aplikacij je nemogoče brez temeljitega testiranja. Strukturiran pristop k ocenjevanju in izboljševanju kakovosti spletnih aplikacij (model kakovosti programske opreme) omogoča razvijalcem, da ustvarijo zanesljivejše, učinkovitejše in varnejše rešitve. V skladu s standardom ISO/IEC 25010:2023 – »Sistemsko in programsko inženirstvo« model kakovosti programskega izdelka vključuje številne značilnosti, med katerimi velja omeniti zmogljivostno učinkovitost (sposobnost programskega izdelka, da opravlja svoje funkcije v danem času in pri danih zmogljivostnih parametrih ter učinkovita uporaba virov v danih pogojih) [3]. Testiranje zmogljivosti je smiselno uporabiti kot orodje za ocenjevanje te lastnosti. Posebno vlogo pri tem ima avtomatizirano testiranje. Po podatkih Gartnerjevega analitičnega poročila o sprejemanju in trendih avtomatiziranega testiranja za leto 2023 so med najpogostejšimi vrstami avtomatiziranega testiranja programske opreme testiranje API (56 %), integracijsko testiranje (45 %) in testiranje zmogljivosti (40 %) [4].

Osnovni namen testiranja zmogljivosti je določiti ali preveriti hitrost, razširljivost in/ali stabilnost programske aplikacije. Z njim lahko ugotovimo, ali bo aplikacija v prihodnosti lahko obvladovala pričakovano obremenitev ali pa je pred začetkom delovanja potrebna optimizacija zmogljivosti ali nadgradnja strojne opreme. Testiranje zmogljivosti aplikacije vključuje uporabo specializirane programske opreme (Apache JMeter, Gatling, HP LoadRunner, K6 itd.), ki simulira delo velikega števila hkratnih uporabnikov. Ključna vprašanja so določitev števila in sprememb sočasnih uporabnikov ter narave njihovih dejanj.

Glavne vrste testiranja zmogljivosti vključujejo [5]:

- obremenitveno testiranje (angl. Load testing) – preverjanje delovanja aplikacije pri običajni in največji dovoljeni obremenitvi v skladu z omejitvami;
- stresno testiranje (angl. Stress testing) – preverjanje delovanja aplikacije, ko so ravni obremenitev presežene, da se ugotovijo njene omejitve in stabilnost pri preobremenitvah;
- testiranje kapacitete (angl. Capacity testing) – preverjanje največje zmogljivosti sistema, da se določi meja zmogljivosti v normalnih pogojih delovanja, preden se sistem začne slabšati.

Praviloma testiranje zmogljivosti izvaja inženir za testiranje zmogljivosti. Glavni izzivi pri določanju delovne obremenitve in testiranju zmogljivosti so naslednji:

- potreba po poglobljenem razumevanju poslovne logike aplikacije in primerov uporabe, ki jih je treba preizkusiti;
- potreba po ponovitvi dejanskih uporabniških dejavnosti;
- potreba po ponovitvi dejanskih pogojev delovanja.

Napačna izbira števila sočasnih uporabnikov in napačni scenariji uporabe funkcij aplikacije lahko ne dajo popolne slike o zmogljivosti ali pa preobremenijo sistem preko realnih pogojev. Testiranje zmogljivosti aplikacij se lahko izvede pred izdajo različice programske opreme ali po njej. Za testiranje zmogljivosti pred izdajo nove različice programske opreme (z novimi lastnostmi in funkcionalnostmi) se lahko delovna obremenitev aplikacije določi na podlagi rezultatov pregleda mnenja uporabnikov in strokovnjakov ter rezultatov uporabe beta različice aplikacije s strani skupine reprezentativnih uporabnikov. Testiranje zmogljivosti po izdaji nove različice programske opreme je mogoče izvesti s posodobljenimi ključnimi scenariji uporabe aplikacije, ki temeljijo na podatkih iz dnevniških zapisov spletnega strežnika in odražajo dejansko uporabo aplikacije (dejansko pogostost zahtevkov in zaporedje dejanj uporabnikov). To omogoča natančnejše modeliranje delovnih obremenitev in ocenjevanje zmogljivosti v pogojih, ki so blizu dejanski uporabi.

Predlagana metoda ni namenjena iskanju funkcionalnih napak v programski kodi, temveč določanju ali potrjevanju značilnosti hitrosti, razširljivosti in/ali

stabilnosti testirane programske opreme, pri čemer se upoštevajo podatki o različici programske opreme, ki že deluje v realnih pogojih in je izpostavljena realni obremenitvi. Takšna rešitev omogoča sestavo natančnejšega profila obremenitev (profiliranje obremenitev) ob upoštevanju dejanskega obnašanja uporabnikov ter prepoznavanje težav, ki se morda ne pokažejo v fazi testiranja pred izdajo. Pri izbiri obdobja za analizo podatkov je treba upoštevati skladnost dnevniških zapisov s testno različico (funkcionalnostjo) aplikacije.

Cilj raziskave je razviti metodo za ustvarjanje profilov obremenitev, ki natančno odražajo resnične uporabniške scenarije in omogočajo natančnejše testiranje zmogljivosti spletnih aplikacij. To bo prispevalo k odkrivanju ozkih grl v sistemu in zagotavljanju zanesljivosti v različnih pogojih delovanja. Napredek na področju obravnav dnevniških datotek spletnega strežnika [6], [7], [8], [9], [10] spodbujajo naslednje raziskovalno vprašanje: »Ali je možno izboljšati načrtovanje obremenitev za testiranje zmogljivosti spletnih aplikacij z uporabo metod strojnega učenja in podatkov dnevniških datotek spletnega strežnika?«.

V nadaljevanju članka najprej predstavljamo pregled literature in orodij za analizo dnevniških datotek, ki jima sledi raziskovalna metodologija. Nadaljujemo z razumevanjem problema in podatkov, razvojem metode za ustvarjanje profilov obremenitev, pripravo podatkov, modeliranjem ter ocenjevanjem rezultatov modeliranja in kakovosti modelov. Na koncu zaključimo z oceno pridobljenih rezultatov, omejitvami in implikacijami za prakso.

2 SORODNA DELA

2.1 Pristopi k obravnavi dnevniških datotek spletnega strežnika

Pridobivanje znanja iz dnevniških datotek – računalniških dnevnikov spletnega strežnika je pomembno področje raziskav, saj te datoteke vsebujejo dragocene informacije o uporabnikih, njihovih navadah in vzorcih obnašanja na spletnem mestu. Veliko število avtorjev in obstoječih programskih rešitev za obdelavo podatkov iz dnevniških datotek spletnega strežnika se osredotoča na statistično obdelavo teh podatkov [6], [11], [12]. To vključuje analizo obdobj največje aktivnosti uporabnikov, identifikacijo značilnosti uporabnikov in ugotavljanje najbolj prilju-

bljenih strani. Na primer, s statistično analizo lahko ugotovimo, ob katerih urah je naše spletno mesto najbolj obiskano, katere strani so najbolj obiskane in kateri uporabniki so najpogostejši obiskovalci. Vendar pa ta pristop pogosto zanemarija odnose med posameznimi obiskanimi stranmi, kar pomeni, da ne upošteva, kako uporabniki »premikajo« po spletnem mestu in katere spletne strani pogosto obiskujejo v določenem zaporedju. Poleg statistične obdelave obstajajo tudi druge naloge za obdelavo podatkov iz dnevniških datotek spletnega strežnika, ki se osredotočajo na kompleksnejše analize. Ena izmed takih nalog je rudarjenje asociacijskih pravil, ki vključuje iskanje nizov povezanih strani [9], [11], [12], [13]. To pomeni, da lahko odkrijemo, katere strani so pogosto obiskane skupaj, kar nam lahko pomaga pri optimizaciji strukture spletnega mesta. Druga naloga je rudarjenje zaporednih vzorcev, kjer ugotavljamo urejene verige obiskov strani. Na ta način lahko razumemo, kako se uporabniki premikajo po spletnem mestu in katere poti najpogosteje izberejo. Gručenje je še ena pomembna naloga, pri kateri združujemo uporabnike s podobnimi značilnostmi v skupine [11], [12], [14]. Na primer, lahko združimo uporabnike glede na njihove demografske podatke ali glede na strani, ki jih najpogosteje obiskujejo. Klasifikacija je podobna naloga, vendar se osredotoča na uvrščanje uporabnikov v vnaprej določene skupine na podlagi njihovih značilnosti. Na primer, uporabnike lahko uvrstimo v skupine glede na njihove nakupovalne navade ali glede na to, kako pogosto obiskujejo spletno mesto.

Določanje optimalne strukture navigacije po spletnem mestu je še ena ključna naloga, ki pomaga izboljšati uporabniško izkušnjo [11]. S pomočjo analiz dnevniških datotek lahko ugotovimo, katere strani so težko dostopne in katere povezave bi bilo smiselno dodati, da bi uporabnikom olajšali navigacijo. Vse te naloge prispevajo k boljšemu razumevanju uporabnikov in njihovih potreb, kar je ključno za učinkovito upravljanje in optimizacijo spletnih mest [15], [16], [17].

Za našo raziskavo je še posebej zanimiv koncept rudarjenja spletne uporabe, ki pomaga pridobiti zanimive vzorce iz dnevniških zapisov dostopa do spletnega strežnika [10]. Ta pristop omogoča razvrščanje uporabniških zahtev v skupine brez človeškega sodelovanja na podlagi merila podobnosti. Na primer, lahko avtomatsko združimo uporabnike, ki

pogosto obiskujejo določene strani ali ki imajo podobne vzorce obnašanja na spletnem mestu. S tem pristopom lahko pridobimo dragocene vpogleda v obnašanje uporabnikov in optimiziramo spletno mesto glede na njihove potrebe.

Tipičen postopek obdelave dnevniških datotek spletnega strežnika vključuje več korakov, ki so med seboj povezani in se dopolnjujejo [7], [17], [18], [19].

Najbolj pomembna med njimi sta:

- predhodno čiščenje in preoblikovanje podatkov: ta korak vključuje odstranitev večpredstavno-stnih zahtev, kot so slike, videoposnetki, datoteke CSS in JS, ter zahtev s 4xx, 3xx, 5xx odgovoroma (napake pri odjemalcu, preusmeritev, napake na strežniku) in drugih nepomembnih zahtev. Cilj je pridobiti čiste in relevantne podatke, ki so pomembni za nadaljnjo analizo. S tem zagotovimo, da se v analizo vključijo samo pomembni podatki, kar poveča natančnost rezultatov [17], [20];
- združevanje zapisov, ki se nanašajo na posameznega uporabnika: ta korak vključuje združevanje zapisov v dnevniških datotekah, ki pripadajo istemu uporabniku, v eno skupino s podskupinami na podlagi najdaljšega omejenega časa seje (30 minut). To nam omogoča, da spremljamo uporabniške seje in razumemo, kako dolgo uporabniki ostanejo na spletnem mestu in katere strani obiščejo v tem času [8], [13], [21].

Ti pristopi omogočajo globlje razumevanje obnašanja uporabnikov in optimizacijo spletnih mest glede na njihove potrebe in pričakovanja. Pridobljene informacije so ključne za izboljšanje uporabniške izkušnje, povečanje učinkovitosti spletnega mesta in doseganje boljših poslovnih rezultatov. S temi pristopi lahko natančno analiziramo obnašanje uporabnikov, prepoznamo vzorce in trende ter sprejmemo informirane odločitve o tem, kako izboljšati naše spletno mesto in zagotoviti, da ustreza potrebam naših uporabnikov.

2.2 Podobna orodja

Obstoječe programske rešitve, ki omogočajo obdelavo podatkov dnevniških datotek, kot so GoAccess (<https://goaccess.io/>), Graylog (<https://graylog.org/>), AWStats (<https://www.awstats.org/>), Coralogix (<https://coralogix.com/>), Zebrium (<https://www.zebrium.com/>), so po svoji funkcionalnosti usmerjene v zbiranje in zagotavljanje statističnih informacij o delovanju aplikacij in uporabnikov, tudi z namenom

ugotavljanja groženj in incidentov. Primeri takih funkcionalnosti vključujejo informacije o številu unikatnih obiskovalcev, trajanju sej, porazdelitvi obremenitev po dnevih in urah, informacije o najbolj obiskanih straneh. Nekatere aplikacije, kot so Coralogix, Zebrium, vključujejo uporabo algoritmov strojnega učenja, kot so samodejno odkrivanje incidentov programske opreme in ugotavljanje temeljnih vzrokov. Vendar te zmogljivosti niso namenjene prepoznavanju uporabnikov in njihovega obnašanja, kar lahko bi pomagalo pri profiliranju obremenitev spletne aplikacije.

Analiza literature je pokazala, da obstajajo splošne zasnove problema in potreba po razvoju metode za podporo profiliranju obremenitev z uporabo algoritmov strojnega učenja za podatke dnevniških datotek, vendar pripravljenih rešitev, ki bi bili namenjeni temu, nismo našli.

3 METODOLOGIJA

V predstavljeni raziskavi smo uporabili naslednje metode in orodja:

- za splošni raziskovalni pristop smo izbrali strategijo načrtovanja in razvoja (angl. Design Science Research, DSR) [22];
- v okviru razvojnega cikla smo sledili standardu CRISP-DM (angl. Cross Industry Standard Process for Data Mining) methodology [23];
- v fazi modeliranja, ki je del CRISP-DM, smo za podatkovno rudarjenje uporabili programski jezik Python in integrirano razvojno okolje PyCharm Community Edition;
- kot metrike za ocenjevanje kakovosti gručenja smo uporabili Silhouette (meri, kako blizu so točke znotraj skupine in kako daleč so od drugih skupin), Davies-Bouldin (ocenjuje podobnost med skupinami), Calinski-Harabasz (izračunava razmerje med notranjo in zunanjo disperzijo skupin) in Dunnov Indeks (meri ločitev in gostoto skupin). Vsaka od teh metrik ocenjuje različne vidike gručenja, od gostote gruč (Silhouette in Dunnov index) do ločljivosti gruč in razlik med njimi (Davies-Bouldin in Calinski-Harabasz).

3.1 Raziskovalni pristop razvoja in načrtovanja

Metodologija načrtovanja in razvoja je učinkovito orodje za reševanje kompleksnih problemov, kot je »profiliranje obremenitev spletne aplikacije na podlagi dnevniških zapisov spletnega strežnika«. Osredotoča

se na ustvarjanje in vrednotenje artefaktov, ki obravnavajo ugotovljene probleme ter zagotavljajo znanstvene prispevke in praktične koristi [22], [24], [25].

DSR ne omogoča le razvoja verodostojnih in učinkovitih metod za profiliranje obremenitev spletne aplikacije, temveč tudi formalizacijo znanja, pridobljenega med njihovim razvojem in testiranjem [26].

Ciklični proces metodologije DSR pomeni, da je pri profiliranju obremenitev spletne aplikacije mogoče nenehno izboljševati razvite metode in modele na podlagi dejanskih podatkov in povratnih informacij. Ta ponavljajoči se postopek zagotavlja, da so razvita orodja in metode ustrezne in učinkovite.

Ocenjevanje in utemeljevanje ustvarjenih metod in modelov poudarja metrike in merili, ki bodo pomagali objektivno oceniti kakovost in učinkovitost nastalih metod in modelov. Ta pristop je ključen za praktično vrednost in uporabnost rezultatov v praksi [25].

3.2 CRISP-DM metodologija

Metodologija CRISP-DM zagotavlja sistematičen in strukturiran proces, ki zajema vse faze projekta profiliranja obremenitev spletne aplikacije na podlagi dnevniških zapisov obiskov spletnega strežnika [23], [27], [28].

Prvi korak CRISP-DM je poslovno razumevanje. Pomembno je razumeti, katera konkretna vprašanja želimo rešiti z analizo dnevniških zapisov. V našem primeru je glavni cilj te faze ugotoviti in analizirati obnašanje uporabnikov na spletnem mestu.

Druga faza je razumevanje podatkov. Na tej stopnji zberemo in pregledamo dnevniški zapisi spletnega strežnika. To vključuje preverjanje popolnosti in kakovosti podatkov ter začetne analize za boljše razumevanje, kateri podatki so na voljo in kako jih je mogoče uporabiti. Razumeti je treba strukturo dnevniške datoteke, določiti polja, oblike podatkov, razmejitev, časovne oznake, zahteve URL in odzivne kode strežnika, ki so pomembne z vidika analize.

Tretja faza je priprava podatkov. Na tej stopnji podatki očistimo in preoblikujemo za nadaljnjo analizo. To vključuje filtriranje nepomembnih zapisov, združevanje zapisov v skupine, obdelavo manjkajočih vrednosti, obdelavo besedilnih vrednosti in pretvorbo podatkov v obliko, ki je primerna za analizo.

Četrta faza je modeliranje. Tu ustvarimo in preizkusimo različni modeli za analizo dnevniških zapisov. Za našo nalogo so to modeli za gručenje. Algorit-

me za gručenje lahko uporabimo za prepoznavanje skupin uporabnikov s podobnim obnašanjem.

Peta faza je ocenjevanje. Na tej stopnji ocenimo kakovost in natančnost ustvarjenih modelov. Prepričati se je treba, da modeli resnično rešujejo naloge in dosegajo poslovne cilje.

Šesta faza je uvajanje. Po uspešni oceni so modeli pripravljene za uporabo v proizvodnji. Pomembno je, da so rezultati analize lahko dostopni in razumljivi zainteresiranim stranem.

3.3 Metoda za podporo profiliranju obremenitev

Razvita metoda za podporo profiliranju obremenitev vsebuje naslednje korake (slika 1):

1. Izbira časovnega obdobja. V prvi fazi določimo časovno obdobje, za katero se podatki analizirajo. Pomembno je izbrati reprezentativno obdobje ob upoštevanju največjih obremenitev sistema, tako da rezultat odraža najbolj kritične trenutke delovanja aplikacije.
2. Čiščenje podatkov. Ta korak vključuje odstranitev nepravilnih ali nepopolnih zapisov, ki vsebujejo informacije, ki bi lahko popačile rezultate nadaljnjih analiz.
3. Združevanje podatkov po uporabnikih. To omogoča sistematizacijo dejanj posameznega uporabnika v okviru analize njegovih interakcij s spletno aplikacijo.
4. Združevanje podatkov po sejah. Ta pristop omogoča analizo dejanj vsakega uporabnika v okviru ene seje, kar je potrebno za natančno modeliranje obnašanja uporabnikov.
5. Vektorizacija zahtevkov. V tem koraku pretvorimo uporabniške zahteve v numerične vektorje za nadaljnjo analizo. Ta korak je ključnega pomena za nadaljnje gručenje in analizo podatkov.
6. Gručenje IP-naslovov. Za nadaljnjo analizo razdelimo IP-naslovi uporabnikov v gruče, kar omogoča prepoznavanje podobnih skupin uporabnikov in analizo njihovega obnašanja.
7. Označevanje sej. Vsako sejo označimo z lastnostmi, ki omogočajo natančnejšo razlago obnašanja uporabnikov znotraj posameznih gruč.
8. Združitev in vektorizacija zahtevkov. Uporabniške zahteve iz različnih sej združimo in pretvorimo v vektorje za natančnejšo analizo vzorcev obnašanja v preučevanem časovnem obdobju.
9. Gručenje sej. S tehnikami strojnega učenja seje uporabnikov razvrstimo v skupine na podlagi podob-



Slika 1: Postopek analize podatkov in oblikovanja profilov obremenitev.

nih značilnosti in obnašanja. To omogoča prepoznavanje glavnih vzorcev interakcije z aplikacijo.

10. Interpretacija vsebine gruč. Na podlagi pridobljenih gruč interpretiramo njihovo vsebino, da določimo značilni vzorci obnašanja uporabnikov in vrste sej.
11. Izračun deleža scenarijev za profile. Zadnji korak vključuje izračun deleža različnih scenarijev uporabe spletnih aplikacij za vsak profil obremenitev. To omogoča oblikovanje natančnih profilov obremenitev za testiranje zmogljivosti programske opreme.

Razvita metoda zagotavlja natančno in strukturirano oblikovanje profilov obremenitev, kar pripomore k večji zanesljivosti pridobljenih podatkov iz preskusa zmogljivosti.

4 REZULTATI

4.1 Razumevanje problema in podatkov

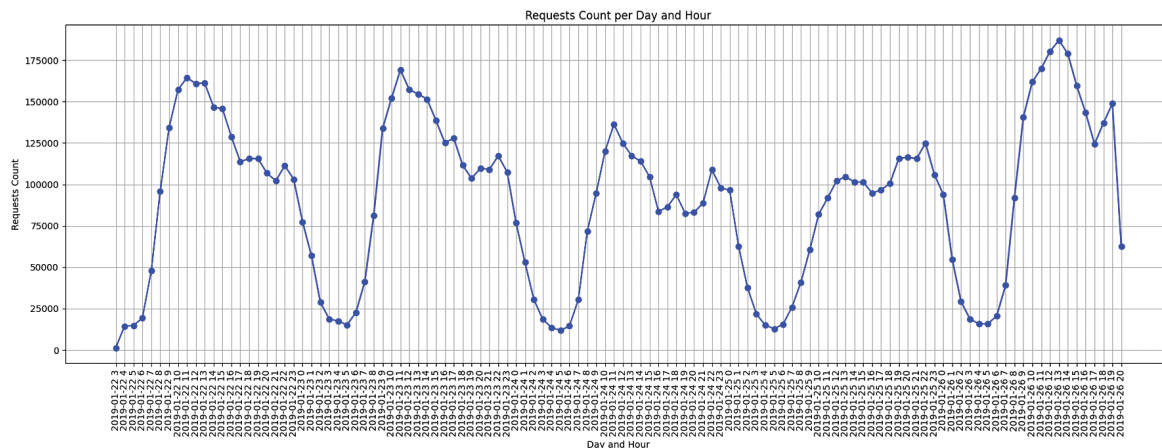
Podatke smo pridobili na spletnem mestu Kaggle (<https://www.kaggle.com/datasets/eliasdabbas/web-server-access-logs>) [29]. To je nabor 3,3 GB zapisov iz spletne trgovine – spletnega strežnika zanbil.ir. Struktura datoteke ustreza formatu CLF (angl. Common Log Format) [30]. Ta nabor podatkov je bil izbran zaradi njegove dostopnosti in razpoložljivosti podatkov o dejanskih zahtevah na spletnem mestu. Vendar se lahko metoda, predlagana v tem članku,

uporabi tudi za druga spletna mesta, saj so ključni analizirani parametri, kot so IP-naslov klienta, datum in čas zahteve, URL in koda stanja odgovora, skupni večini spletnih mest ne glede na vsebino ali posebnosti spletnega mesta.

Podatki vsebujejo 10 atributov in 10365152 zapisov v obliki CSV-datoteke. Znak presledka se uporablja kot ločilo, pri kompleksnih vrednostih nizov pa se upošteva njihovo okvirjanje z narekovaji. Primer in struktura sta navedena v tabeli 1.

Tabela 1: Primer in struktura podatkov.

Parameter	Vrednost
remotehost	54.36.149.41
rfc931	–
authuser	–
date	[22/Jan/2019:03:56:14 +0330]
method	GET
request	/image/60844/productModel/200x200
protocol	HTTP/1.1
status	200
bytes	5667
extra_field	https://www.zanbil.ir/m/filter/b113« »Mozilla/5.0 (Linux; Android 6.0; ALE-L21 Build/HuaweiALE-L21) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.158 Mobile Safari/537.36« »-«



Slika 2: Število zahtev po datumu in času.

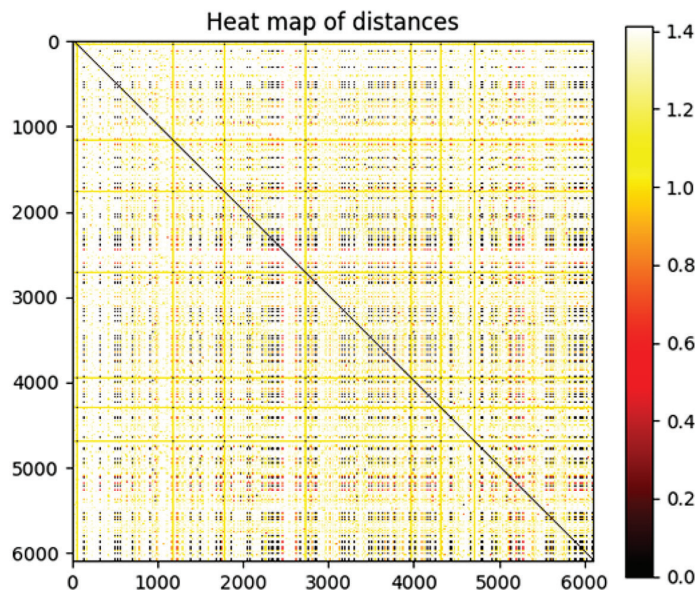
Datoteka vsebuje podatke za obdobje od 22. 01. 2019, in sicer od 03:56:14 naprej, do 26. 01. 2019, in sicer do 20:29:13 (prikazano na sliki 2).

Iz slike 2 je razvidno, da se število uporabniških zahtev spreminja vsak dan na enak ciklični način. Opazimo jasne vzorce, ki kažejo na dnevne cikle v obiskanosti spletnega mesta, na primer, da se število zahtev znatno poveča ob določenih urah dneva, kot so dopoldanske ure, ko je aktivnost uporabnikov najvišja.

Razumevanje teh cikličnih vzorcev je zelo pomembno za profiliranje obremenitev. S tem pristopom lahko natančneje prepoznamo vzorce obnašanja uporabnikov in njihove zahteve, kar nam omogoča usmeritev pri sestavljanju profilov obremenitev.

Izvorni podatki niso predprocesirani in očiščeni. Vse podatke je nesmiselno analizirati v okviru oblikovanja enega samega modela. Z izbiro podatkov v bližini največjih obremenitev se lahko osredotočimo na najpomembnejše in najbolj kritične točke, da ustvarimo natančne profile obremenitev in ugotovimo ključne težave, povezane z delovanjem in obnašanjem sistema pri največji obremenitvi.

Slika 3 prikazuje razdaljo med vektorji, pridobljenimi s tokenizacijo besedila, kjer diagonalna črta prikazuje podobne ali enake primere in svetlo obarvani deli večjo razliko med besedili. Iz slike 3 so razvidni vzorci blokov, ki nakazujejo obstoj skupine besedil, ki so si znotraj skupine podobna, vendar se razliku-



Slika 3: Toplotni zemljevid.

jejo od besedil v drugih skupinah. Razlika v intenzivnosti barve znotraj vsakega bloka kaže, da vsa besedila v isti gruči niso enaka in da obstajajo stopnje podobnosti.

4.2 Priprava podatkov

Dnevniška datoteka vsebuje informacije o vseh zahtevah aplikaciji in odgovorih nanjo. Za pravilne šteje samo tiste zahtevke, katerih status odgovora je enak 2xx (uspešno). Zahteve z drugimi statusi odgovora (preusmeritev, neveljaven naslov, notranja napaka strežnika in druge) ne bomo upoštevali.

Interakcija uporabnika s katero koli spletno aplikacijo je organizirana tako, da se po dostopu do določene strani odjemalcu prenesejo številni pomožni podatki, ki ne nosijo nobene semantične obremenitve, ki bi opisovala obnašanje uporabnika, zato jih ni smiselno analizirati. To so lahko datoteke CSS, datoteke JavaScript, multimedijske datoteke, zahteve robotov-iskalnikov in tako naprej. Prisotnost vsaj ene besede iz nabora ključnih besed povzroči izbris takega zapisa.

Obnašanje uporabnikov v različnih dnevih, vendar v istih časovnih intervalih je precej podobno, zato izberimo samo zapise, ki pripadajo samo enemu časovnemu intervalu v bližini največjih obremenitev: v našem primeru so to vsi dnevi in čas v razponu od »10:00:00« do »10:59:59«.

Prva predobdelava vrne slovar, ki vsebuje dnevniške zapise, združene po IP-naslovu in razvrščene v podkategorije po 20-minutnih intervalih neaktivnosti in po času. Osnovna zamisel je, da uporabnikov ni mogoče enoznačno identificirati samo na podlagi IP-naslava, saj jih lahko skriva strežnik NAT (angl. Network Address Translation) ali proxy-strežnik. Za analizo bomo uporabili seje uporabnikov (zaporedje obiskov spletnega mesta). Če je med dvema najbližjima zahtevkoma z enega IP-naslava preteklo več kot 20 minut (1200 sekund), bo drugi od njiju obravnavan kot zahtevka, ki odpira novo sejo interakcije uporabnika z aplikacijo. Nekateri avtorji [8], [21] predlagajo, da se upošteva prag trajanja seje 25-30 minut, vendar je lahko čas interakcije uporabnika s sodobnimi spletnimi aplikacijami, zlasti z mobilnih naprav, krajši zaradi višjih hitrosti dostopa in uporabe hitrih tehnologij, kot je AJAX. Predhodna analiza je pokazala, da nam zmanjšanje intervala na 20 minut omogoča ustrezno razvrstitev podatkov za analizo obremenitev.

Z uporabo razreda TfidfVectorizer iz knjižnice scikit-learn smo na seznamu sej izvedli oceno pomembnosti besede v dokumentu (seji) glede na celoten korpus. Metodo TfidfVectorizer smo izbrali, ker je dobra pri izločanju pomembnih izrazov za analizo spletnih sej, pri čemer upošteva pogostost besed v seji in njihovo edinstvenost v naboru podatkov, podpira pa tudi n-grame. Nastalo matriko pa je mogoče uporabiti za različne naloge strojnega učenja, kot so klasifikacija besedila, gručenje, iskanje itd.

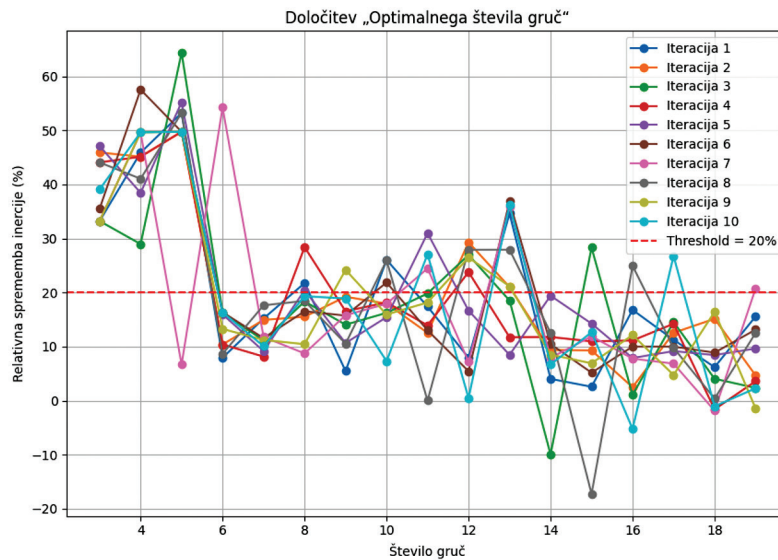
4.3 Modeliranje in ocenjevanje pridobljenih rezultatov

Po zaporedni izbiri algoritmov gručenja in analizi rezultatov gručenja smo vse uporabnike (»IP-naslovi«) razdelili v 2 gruči. Uporaba več metrik nam omogoča celovitejšo sliko kakovosti gručenja, kar poenostavi analizo in interpretacijo rezultatov. Spektralni (Silhouette=0,1587, Davies-Bouldin=1,3163, Calinski-Harabasz=1036,3703, Dunnov Index=0,0993) in aglomerativni (Silhouette=0,1536, Davies-Bouldin=1,2549, Calinski-Harabasz=991,6595, Dunnov Index=0,3056) algoritma gručenja sta izkazala najboljše rezultate za razdelitev v 2 gruči.

Pred izvedbo modeliranja in analizo obnašanja uporabnikov spletne aplikacije na podlagi dnevniških zapisov spletnega strežnika, smo izvedli naslednjo transformacijo: prvotne seje uporabnikov smo povezali z IP-naslovi in vsako od nastalih dveh gruči.

Za ponovno vektorizacijo besedila smo ustvarili objekt TfidfVectorizer s parametrom ngram_range=(1, 2) in pripravili unigrame (posamezne besede, npr. »mačka«, »pes«) kot bigrame (zaporedja dveh besed, npr. »mačka, pes«, »pes laja«) in parametrom min_df=0,1. Vrednost 0,1 za min_df določa najmanjšo pogostost dokumentov, da se izraz vključi v slovar. Vrednost 0,1 pomeni, da se izrazi, ki se pojavljajo v manj kot 10 % dokumentov, ne upoštevajo. Če imamo na primer 100 dokumentov, se mora beseda pojaviti v vsaj 10 od njih, da bo vključena v slovar. Ta odstotek omogoča izključitev elementov, ki so zelo redki in so lahko šum, ki ne vpliva na celotno sliko.

Opazovanje spreminjanja porazdelitve podatkov v gruče je pokazalo, da ko je dosežena razdelitev podatkov v določeno število gruči N, nadaljnje povečevanje števila gruči ne vpliva bistveno na kakovost rezultatov gručenja, saj nove gruče vsebujejo tako majhno količino podatkov, da jih lahko zanemarimo



Slika 4: Izračun povprečja optimalnega števila gruč.

brez večje izgube informacijske sestavine. Ta pristop pomaga najti razpon vztrajnosti κ , pri katerem nadaljnje povečanje števila gruč bistveno ne zmanjša vztrajnosti, kar kaže na optimalno število gruč za ta problem. Sprememba vztrajnosti je bila vzeta kot osnovna metrika za spremembo v izgubi informacij. Vztrajnost κ je metrika, ki meri vsoto kvadratov razdalj od vsake točke do najbližjega centroida. Uporablja se za ocenjevanje kakovosti gručenja, saj pokaže, kako kompaktne so gruče. Celoten postopek lahko opišemo na naslednji način:

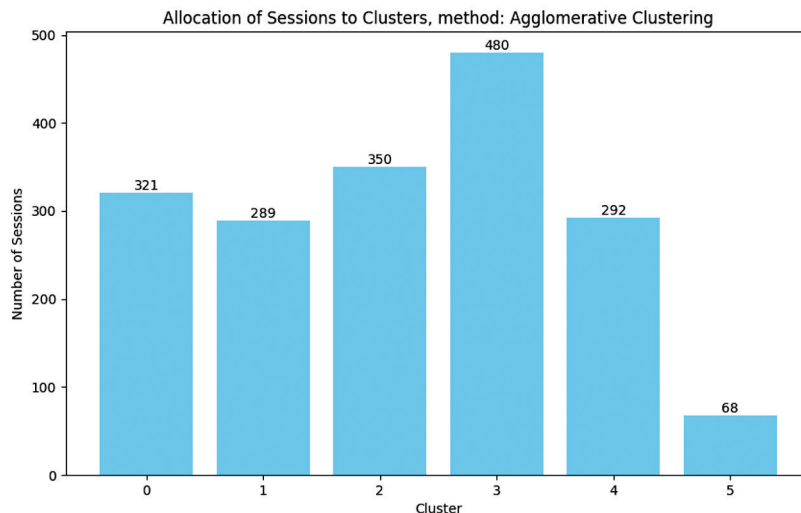
1. Opredelitev razpona vrednosti κ , za katere bo izvedeno gručenje.
2. Izvedba gručenja in ohranjanje vztrajnosti: za vsako vrednost κ ustvarimo in usposobimo model K-Means. Vztrajnost se nato shrani na seznam vztrajnosti.
3. Izračun relativne spremembe vztrajnosti: za vsako vrednost κ izračunamo relativno spremembo vztrajnosti v odstotkih glede na preteklo vrednost κ .
4. Izpis rezultatov relativne spremembe vztrajnosti.

Optimalno število gruč se določi na podlagi mejne vrednosti relativne spremembe vztrajnosti. Če je relativna sprememba vztrajnosti manjša od mejne vrednosti, se ustrezna vrednost κ izbere kot optimalna vrednost. Med izvajanjem eksperimentalne analize smo preskusili različne mejne vrednosti in ugotovili, da je 20-odstotna vrednost najpogosteje povzročila nastanek gruč, ki so bile smiselne z vidika razlage in uporabe v nadaljnjih analizah.

Po določitvi najboljšega števila gruč smo izvedli gručenje sej z metodo aglomerativnega gručenja. Ocenili smo metrike tega modela, dobljene gruče pa smo tudi interpretirali. V tem koraku je bilo pomembno ugotoviti, ali so si seje v isti skupini po svoji strukturi bolj podobne v primerjavi s sejami iz drugih skupin. Za prvo skupino uporabnikov ni bilo mogoče določiti skupin sej, saj so bile vse seje in posledično zahteve v njih zelo podobne.

V naslednjem koraku smo analizirali značilnosti sej, ki so razvrščene v isto gručo. Uporabniki aktivno uporabljajo mobilne različice strani in optimizirane strani AMP (angl. Accelerated Mobile Pages) za hiter dostop do vsebine ter iščejo določene izdelke, kar počnejo z iskalnikom. AMP je Googlova pobuda, namenjena izboljšanju zmogljivosti in hitrosti nalaganja mobilnih spletnih strani. Glavni cilj AMP je uporabnikom zagotoviti najhitrejši možni dostop do vsebine v mobilnih napravah. Strani AMP imajo običajno preprosto zasnovo brez zapletenih elementov, kar prav tako prispeva k hitrejšemu nalaganju.

Na sliki 4 prikazujemo rezultate določanja optimalnega števila gruč na podlagi relativne spremembe vztrajnosti za drugo skupino uporabnikov s spreminjanjem števila gruč od 2 do 20. Izračun smo ponovili desetkrat, da bi dobili aritmetično sredino optimalnega števila gruč glede na metriko »relativna sprememba vztrajnosti«. Os X prikazuje število gruč, os Y pa spremembo vztrajnosti v odstotkih. Posamezne krivulje (slika 4) ustrezajo eni od 10 iteracij algoritma gručenja, rdeča črtkana premica pa označuje



Slika 5: Porazdelitve sej znotraj gruĉ v drugi skupini uporabnikov.

prag 20%, pri katerem postane nadaljnje poveĉevanje števila gruĉ neuĉinkovito. Videti je, da se pri številu gruĉ okoli 6 vrednost vztrajnosti znatno spremeni, ĉemur sledi stabilizacija pod 20-odstotnim pragom, kar kaŹe na optimalno število gruĉ za ta problem.

Rezultati porazdelitve sej znotraj gruĉ v drugi skupini uporabnikov so prikazani na sliki 5 in sliki 6. Aglomerativno gruĉenje je dobro razdelilo podatke v veĉ homogenih skupin.

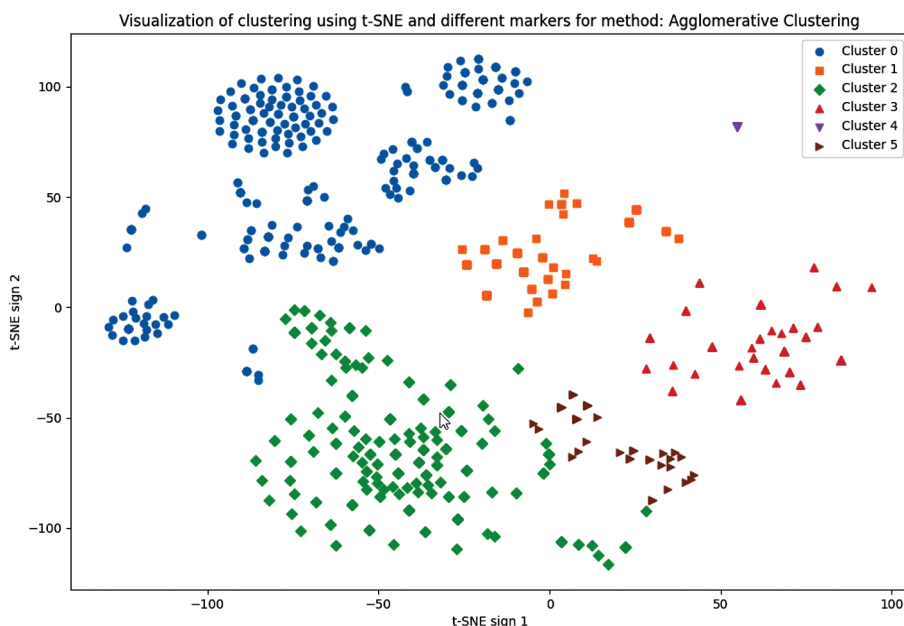
Koeficient silhuete 0,7659 in indeks Kalinskega-Harabasha 2771,7697 kaŹeta na jasno loĉitev gruĉ in

visoko homogenost znotraj gruĉ. Tudi Davis-Bouldinov indeks 0,5800 potrjuje kakovostno gruĉenje z nizko stopnjo prekrivanja med gruĉami. Nizka vrednost Dunnovega indeksa (0,1152) pa kaŹe na morebitne teŹave z gostoto in loĉevanjem gruĉ.

V nadaljevanju opisujemo glavne vzorce in znaĉilnosti sej za vsako nastalo gruĉo.

Gruĉa 0:

- skoraj vse seje vkljuĉujejo zahteve na stran '/site/enamad', ki je verjetno kljuĉna stran;



Slika 6: Vizualizacija znotraj gruĉ v drugi skupini uporabnikov.

- uporabniki pogosto obiskujejo kategorije, kot so gospodinjski aparati, avdio in video oprema, mobilni telefoni;
- številne seje vključujejo zahteve za filtriranje izdelkov in uporabo AJAX zahtev;
- nekatere seje vključujejo ogled podrobnosti o izdelku, dodajanje izdelkov v košarico in oddajo naročila;
- v več sejah uporabniki uporabljajo iskanje in samodejno dopolnjevanje.

Priporočljivo je štetje zahtev, da se ugotovijo najbolj obiskane strani, priljubljene kategorije izdelkov, uporaba filtrov, AJAX zahtev in dejanja s košarico ter nakupi za gručo 0.

Gruča 1:

- uporabniki pogosto obiskujejo različne kategorije izdelkov;
- mnoge seje uporabljajo mobilno različico spletnega mesta in strani AMP;
- uporabniki aktivno uporabljajo filtre za iskanje izdelkov;
- nekatere seje vključujejo naročanje in spreminjanje različic izdelkov.

Priporočljivo je štetje zahtev za določitev najbolj obiskanih kategorij izdelkov, uporabe mobilne različice in strani AMP, filtrov ter dejanj s košarico in blagajno za gručo 1.

Gruča 2:

- uporabniki obiskujejo določene strani z izdelki in kategorije izdelkov;
- Mnoge seje uporabljajo mobilno različico spletnega mesta in strani AMP;
- uporabniki aktivno uporabljajo filtre za iskanje izdelkov;
- nekatere seje vključujejo naročanje in spreminjanje različic izdelkov.

Priporočljivo je štetje zahtev za določitev najbolj obiskanih kategorij izdelkov, uporabe mobilne različice in strani AMP, filtrov ter dejanj s košarico in blagajno za gručo 2.

Gruča 3:

- uporabniki aktivno uporabljajo filtre za iskanje izdelkov in izvajajo zahteve AJAX;
- mnoge seje vključujejo uporabo mobilne različice spletnega mesta in strani AMP;
- uporabniki obiščejo strani s kategorijami izdelkov in uporabljajo filtre;
- pogost je dostop do strani s popusti in promocijami.

Priporočljivo je štetje zahtev za določitev najbolj obiskanih kategorij izdelkov, uporabe filtrov, mobilne različice spletnega mesta, strani AMP ter dejanj z brskanjem in dodajanjem izdelkov v košarico za gručo 3.

Gruča 4:

- uporabniki pogosto obiskujejo domačo stran in strani s promocijami;
- mnoge seje vključujejo uporabo mobilne različice spletnega mesta in strani AMP;
- nekatere seje vključujejo sledenje naročilom in spletna plačila;
- uporabniki aktivno uporabljajo filtre za iskanje izdelkov in izvajajo zahteve AJAX;
- pogosto dostopajo do strani z vsebino in članki.

Priporočljivo je štetje zahtev za določitev najbolj obiskanih strani, uporabe mobilne različice in strani AMP, dejanj s sledenjem naročilom, spletnimi plačili, uporabo filtrov in ogledom vsebine za gručo 4.

Gruča 5:

- uporabniki aktivno uporabljajo filtre in zahteve AJAX za iskanje in razvrščanje izdelkov;
- pogosto dostopajo do strani z izdelki in jih primerjajo;
- veliko sej poteka prek mobilne različice spletnega mesta;
- uporabniki pogosto uporabljajo iskalnik za iskanje izdelkov;
- izdelke pogosto dodajajo v košarico, brskajo in naročajo.

Priporočljivo je štetje zahtev za določitev najpogostejše uporabljenih funkcij, kot so filtri, zahteve AJAX, brskanje, primerjanje izdelkov, uporaba mobilne različice spletnega mesta, iskalne zahteve in dejanja v košarici za gručo 5.

5 DISKUSIJA

Glavni cilj raziskave je bil izboljšanje kakovosti oblikovanja profilov obremenitev za namene testiranja zmogljivosti spletnih aplikacij. Ta cilj smo dosegli z razvojem metode za podporo profiliranju obremenitev spletne aplikacije na podlagi dnevniških zapisov obiskov spletnega strežnika za nadaljnje testiranje zmogljivosti.

Razvita metoda za podporo profiliranju obremenitev ponuja učinkovit in prilagodljiv pristop k analizi uporabniških interakcij s spletnimi aplikacijami. Njena ključna prednost je v tem, da omogoča natančno modeliranje obnašanja uporabnikov, kar je ključno za testiranje zmogljivosti sistemov. Z uporabo različnih tehnik za obdelavo in analizo podatkov razvita metoda zagotavlja, da so profili obremenitev realistični in temeljijo na dejanskih uporabniških vzorcih. Poleg tega metoda omogoča prilagodljivost pri uporabi na različnih spletnih straneh in domenah, saj temelji na splošnih parametrih, ki so prisotni pri večini spletnih aplikacij.

Za doseg zastavljenega cilja smo izvedli več ključnih korakov. Najprej smo preučili ključne procese in pristope k pridobivanju znanja iz dnevniških datotek. Nato smo zbrali dnevniške datoteke in druge relevantne podatke iz produkcijskega okolja spletne aplikacije in izvedli začetno analizo teh podatkov. Podatke smo pripravili, da so bili primerni za nadaljnjo analizo. Na podlagi teh podatkov smo razvili algoritme in modele gručenja.

V raziskavi smo uporabili metode in orodja, ki so standardna za podatkovno analitiko in strojno učenje. Na voljo so v priljubljenih knjižnicah in orodjih, kot je Scikit-learn za Python, zato jih je mogoče zlahka uporabiti za različne vrste podatkov in scenarije. Ti pristopi so dokazano uspešni in so bili uporabljeni v številnih raziskovalnih in komercialnih aplikacijah, kar poudarja njihovo vsestranskost.

Veljavnost naših rezultatov, smo preverili z uporabo metrik koeficient Silhouette, indeks Kalinskega-Harabasha, Davies-Bouldinov indeks, Dunnov indeks in algoritmov gručenja (K-Means, Aglomerativno, Spektralno, DBSCAN in OPTICS) ter primerjali dobljene rezultate. Tako smo lahko potrdili natančnost in uporabnost rezultatov modeliranja v okviru resničnih podatkov o spletni aplikaciji.

Rezultati gručenja sej kažejo na dobro kakovost gručenja: koeficient Silhouette 0,7659 in indeks Kalinskega-Harabasha 2771,7697 kažeta na jasno ločitev

gruč in visoko homogenost znotraj gruče. Davies-Bouldinov indeks 0,5800 potrjuje kakovostno gručenje z nizko stopnjo prekrivanja med gručami. Nizka vrednost Dunnovega indeksa (0,1152) pa kaže na morebitne težave z gostoto in ločevanjem gruč. Glede na dobljene izračune je najboljšo število gruč za delitev šest. Vizualizacija na sliki 6 prikazuje dobro porazdelitev sej v gruče, ki potrjujejo jasne meje med njimi.

Analiza sej v posameznih gručah razkriva različne vzorce uporabniškega obnašanja. Uporabniki, ki uporabljajo mobilno različico spletnega mesta in strani AMP, se pogosto uvrščajo v iste gruče zaradi podobnih zahtev in vedenjskih vzorcev. Glavni vzorci vključujejo uporabo filtrov, zahtev AJAX, pregledovanje podrobnosti o izdelku in upravljanje naročil. Na podlagi teh podatkov lahko razvijemo niz testnih scenarijev, ki odražajo resnično obnašanje uporabnikov in pričakovano obremenitev sistema. Če na primer v testne scenarije vključimo zahteve, povezane s filtriranjem ali pregledovanjem podrobnosti o izdelku, lahko ocenimo, kako se bo aplikacija odzvala na take zahteve v pogojih največjih obremenitev.

Rezultati te raziskave potrjujejo pomembnost preteklih raziskav tako na področju ustvarjanja profilov za testiranje obremenitev [31], [32], [33], [34] kot na področju pridobivanja znanja iz dnevniških datotek [6], [8], [9], [10]. Medtem ko so obstoječe programske rešitve za obdelavo podatkov iz dnevniških datotek osredotočene predvsem na pridobivanje statističnih informacij, razvita metoda odpira nove perspektive pri uporabi gručenja za analizo dnevniških datotek. Ti rezultati so namenjeni izboljšanju kakovosti in preglednosti dela strokovnjakov, ki se ukvarjajo s testiranjem obremenitev IT-produktov, in prikazujejo možnost določanja obnašanja uporabnikov na podlagi analize dnevniških datotek z uporabo algoritmov strojnega učenja.

Čeprav naša raziskava temelji na podatkih določene spletne aplikacije, so uporabljene algoritmi in razvita metoda splošno uporabni. Rezultate je mogoče prilagoditi in uporabiti za analizo dnevniških zapisov drugih aplikacij.

6 ZAKLJUČEK

V prispevku smo obravnavali problem izboljšanja kakovosti oblikovanja profilov obremenitev za namene testiranja zmogljivosti spletnih aplikacij.

Odgovor na raziskovalno vprašanje, ali je možno izboljšati načrtovanje obremenitev za testiranje zmo-

gljivosti spletnih aplikacij z uporabo metod strojnega učenja in podatkov dnevniških datotek spletnega strežnika, smo demonstrirali na primeru podatkov iz spletne trgovine. Pokazali smo, da je z uporabo razvite metode za podporo profiliranju obremenitev spletnih aplikacij strojnega učenja moč analizirati obnašanje uporabnikov in na podlagi tega izvesti zanesljivejše modeliranje profilov obremenitev spletne aplikacije.

Raziskava je pokazala, da izbrane tehnike za analizo podatkov in modeliranje obremenitev, vključno s knjižnico TfidfVectorizer in različnimi metrikami vrednotenja gručenja, pomagajo pri ustvarjanju natančnih profilov obremenitev za testiranje zmogljivosti spletnih aplikacij. To potrjujejo visoke vrednosti koeficienta silhuete in kazalnikov vrednotenja gručenja. Uspeli smo identificirati več jasnih gruč, ki odražajo skupine uporabnikov. Ugotovili smo, da uporabniki znotraj posameznih gruč resnično kažejo značilne vzorce obnašanja. Razvita metoda omogoča zanesljivo pomoč pri oblikovanju profilov obremenitev spletnih aplikacij, kar je ključno za izboljšanje zmogljivosti in zanesljivosti sistemov. Z uporabo razvite metode lahko sestavimo niz testnih scenarijev, ki odražajo resnično obnašanje uporabnikov in pričakovano obremenitev sistema, kar bo pripomoglo k boljšemu razumevanju in optimizaciji spletnih aplikacij v produkcijskem okolju. Rezultati predstavljajo pomemben prispevek k praksi oblikovanja profilov obremenitev za testiranje zmogljivosti.

Uporabnost razvite metode smo pokazali na primeru dnevniških zapisov strežnika ene spletne trgovine, kar predstavlja omejitev raziskave. Razvito metodo za podporo profiliranju obremenitev je kljub temu mogoče uporabiti za različne spletne aplikacije in scenarije.

ZAHVALA

Raziskava je finančno podprla Javna agencija za znanstvenoraziskovalno in inovacijsko dejavnost Republike Slovenije v okviru raziskovalnega programa P5-0018.

LITERATURA

- [1] NJ, »How Many Websites Are There in the World? (2024)«, Sitefy. Pridobljeno: 15. julij 2024. [Na spletu]. Dostopno na: <https://sitefy.com/how-many-websites-are-there/>
- [2] E. Bocchi, L. De Cicco, in D. Rossi, »Measuring the Quality of Experience of Web users«, v *Proceedings of the 2016 workshop on QoE-based Analysis and Management of Data Communication Networks*, v Internet-QoE '16. New York, NY, USA: Association for Computing Machinery, avg. 2016, str. 37–42. doi: 10.1145/2940136.2940138.
- [3] ISO/IEC 25010:2023(en), *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model*, International standard 25010:2023, november 2023. Pridobljeno: 15. julij 2024. [Na spletu]. Dostopno na: <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:25010:ed-2:v1:en>
- [4] Gartner Peer Community, »Automated Software Testing Adoption and Trends«, Gartner, One-Minute Insight Report, jun. 2023. Pridobljeno: 14. september 2024. [Na spletu]. Dostopno na: <https://www.gartner.com/peer-community/home>
- [5] Microsoft Corporation, *Performance Testing Guidance for Web Applications*, 1st edition. United States? Microsoft Press, 2007.
- [6] K. . Dharmarajan in M. A. Dorairangaswamy, »Discovering User Pattern Analysis from Web Log Data using Weblog Expert«, *Indian Journal of Science and Technology*, let. 9, št. 42, nov. 2016, doi: 10.17485/ijst/2016/v9i42/99300.
- [7] H. Karthakula in N. Gayatri, »Customer Behaviour Prediction Using Web Usage Mining«, *Journal of Cardiovascular Disease Research*, let. 12, št. 03, 2021.
- [8] A. Neetu in H. Saba, »ijcsit2012030251.pdf«, *International Journal of Computer Science and Information Technologies*, let. 3 (2), str. 3536–3539, mar. 2012.
- [9] O. Prakash in A. Jaya, »Analyzing and predicting user behavior pattern from weblogs«, *International Journal of Applied Engineering Research*, let. 11, str. 6278–6283, jan. 2016.
- [10] G. Sudhamathy, S. B. Illango, in J. C. Venkateswaran, »A Comparative Study on Web Log Clustering Approaches«, v *GLOBAL TRENDS IN COMPUTING AND COMMUNICATION SYSTEMS, PT 1*, P. V. Krishna, M. R. Babu, in E. Ariwa, Ur., v Communications in Computer and Information Science, vol. 269. Berlin: Springer-Verlag Berlin, 2012, str. 402–+. Pridobljeno: 11. november 2023. [Na spletu]. Dostopno na: <https://www.webofscience.com/wos/woscc/summary/b7f7fef5-90cb-436e-a3be-ed08b07844b5-b366957b/relevance/1>
- [11] L. K. Joshila Grace, V. Maheswari, in D. Nagamalai, »Analysis of Web Logs And Web User In Web Mining«, *IJNSA*, let. 3, št. 1, str. 99–110, jan. 2011, doi: 10.5121/ijnsa.2011.3107.
- [12] J. Srivastava, R. Cooley, M. Deshpande, in P.-N. Tan, »Web usage mining: discovery and applications of usage patterns from Web data«, *SIGKDD Explor. Newsl.*, let. 1, št. 2, str. 12–23, jan. 2000, doi: 10.1145/846183.846188.
- [13] M. Grcar, »USER PROFILING: WEB USAGE MINING«, predstavljeno na Conference on Data Mining and Warehouses (SiKDD 2004), Ljubljana, Slovenia: J.Stefan Institute, okt. 2004. [Na spletu]. Dostopno na: <https://aile3.ijs.si/dunja/SiKDD2004/Papers/MihaGrcar-WebUsageMining.pdf>
- [14] M. H. A. Elhiber in A. Abraham, »Access Patterns in Web Log Data: A Review«, *Journal of Network and Innovative Computing*, let. Vol. 1 (2013), str. 348–355, okt. 2013.
- [15] Z. Pabarskaite in A. Raudys, »A process of knowledge discovery from web log data: Systematization and critical review«, *J. Intell. Inf. Syst.*, let. 28, št. 1, str. 79–104, feb. 2007, doi: 10.1007/s10844-006-0004-1.
- [16] Y. K. Woon, W. K. Ng, X. Li, in W. F. Lu, »Efficient web log mining for product development«, v *2003 INTERNATIONAL CONFERENCE ON CYBERWORLDS, PROCEEDINGS*, T. L. Kunii, S. H. Soon, in A. Sourin, Ur., Los Alamitos: IEEE Computer Soc, 2003, str. 294–301. doi: 10.1109/CYBER.2003.1253468.
- [17] Z. L. Yang, Y. T. Wang, in M. Kitsuregawa, »An effective system for mining web log«, v *FRONTIERS OF WWW RESE-*

- ARCH AND DEVELOPMENT – APWEB 2006, PROCEEDINGS, let. 3841, X. F. Zhou, J. Li, H. T. Shen, M. Kitsuregawa, in Y. Zhang, Ur., v Lecture Notes in Computer Science, vol. 3841., Berlin: Springer-Verlag Berlin, 2006, str. 40–52. Pridobljeno: 11. november 2023. [Na spletu]. Dostopno na: <https://www.webofscience.com/wos/woscc/summary/73145dd1-9bcf-421e-a023-1cde96a73ed4-b3669701/relevance/1>
- [18] Z. X. Chen, A. W. C. Fu, in F. C. H. Tong, »Optimal algorithms for finding user access sessions from very large web logs«, *World Wide Web*, let. 6, št. 3, str. 259–279, sep. 2003, doi: 10.1023/A:1024606901978.
- [19] Z. Pabarskaite, »Implementing advanced cleaning and end-user interpretability technologies in Web log mining«, v *ITI 2002: PROCEEDINGS OF THE 24TH INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY INTERFACES*, V. Glavinica, V. H. Dobric, in D. Simic, Ur., Zagreb: Univ Zagreb, 2002, str. 109–113. Pridobljeno: 11. november 2023. [Na spletu]. Dostopno na: <https://www.webofscience.com/wos/woscc/summary/458315da-0901-4ce0-bdaf-3f388d9f1836-b3669ae5/relevance/1>
- [20] O. R. Zaiane, »Web Usage Mining for a Better Web-Based Learning Environment«, University of Alberta, Edmonton, Alberta, Canada, Technical report TR01-05, 2001. Pridobljeno: 8. junij 2024. [Na spletu]. Dostopno na: <https://era.library.ualberta.ca/items/0a182195-ce39-4b5d-a1c1-291ed91a0f36/view/336c2a34-b149-4d9d-95a1-ad78c08ee35c/TR01-05.pdf>
- [21] J. Lorenzo in J. Domenech, »A Tool for Web Usage Mining«, let. 4881, str. 704, dec. 2007, doi: 10.1007/978-3-540-77226-2_70.
- [22] A. Hevner *idr.*, »Design Science in Information Systems Research«, *Management Information Systems Quarterly*, let. 28, str. 75, mar. 2004.
- [23] U. Kannengiesser in J. S. Gero, »MODELLING THE DESIGN OF MODELS: AN EXAMPLE USING CRISP-DM«, *Proc. Des. Soc.*, let. 3, str. 2705–2714, jul. 2023, doi: 10.1017/pds.2023.271.
- [24] M. Kljajić Borštnar, *RAZISKOVANJE INFORMACIJSKIH SISTEMOV. UČBENIK*. UNIVERZA V MARIBORU, FAKULTETA ZA ORGANIZACIJSKE VEDE, 2021.
- [25] K. Peffers, T. Tuunanen, M. Rothenberger, in S. Chatterjee, »A design science research methodology for information systems research«, *Journal of Management Information Systems*, let. 24, str. 45–77, jan. 2007.
- [26] A. Hevner, J. vom Brocke, in A. Maedche, *Design Science Research. Cases*, 1. izd. v Progress in IS. Springer International Publishing, 2020. doi: 10.1007/978-3-030-46781-4_1.
- [27] J. Brzozowska, J. Pizoń, G. BAYTIKENOVA, A. Gola, A. ZAKIMOVA, in K. Piotrowska, »DATA ENGINEERING IN CRISP-DM PROCESS PRODUCTION DATA – CASE STUDY«, *Applied Computer Science*, let. 19, str. 83–95, sep. 2023, doi: 10.35784/acs-2023-26.
- [28] V. Plotnikova, M. Dumas, in F. Milani, »Adapting the CRISP-DM Data Mining Process: A Case Study in the Financial Services Domain«, v *Research Challenges in Information Science*, let. 415, S. Cherfi, A. Perini, in S. Nurcan, Ur., v Lecture Notes in Business Information Processing, vol. 415., Cham: Springer International Publishing, 2021, str. 55–71. doi: 10.1007/978-3-030-75018-3_4.
- [29] E. Dabbas, »Web Server Access Logs«. <https://www.kaggle.com/datasets/eliasdabbas/web-server-access-logs>. Pridobljeno: 29. april 2024. [Csv]. Dostopno na: <https://www.kaggle.com/datasets/eliasdabbas/web-server-access-logs>
- [30] A. Foundation, »Log Files – Apache HTTP Server Version 2.4«. Pridobljeno: 29. april 2024. [Na spletu]. Dostopno na: <https://httpd.apache.org/docs/2.4/logs.html>
- [31] A. Amira, A. M. Huda, in B. Nagwa, »Automation of Performance Testing: A Review«, *IJICIS*, str. 1–16, dec. 2022, doi: 10.21608/ijicis.2022.161846.1219.
- [32] G. Din, I. Schieferdecker, in R. Petre, »Performance Test Design Process and Its Implementation Patterns for Multi-services Systems«, predstavljeno na International Workshop on Formal Approaches to Software Testing, Berlin, Heidelberg: Springer, jan. 2008, str. 135–152. doi: 10.1007/978-3-540-68524-1_11.
- [33] S. Pargaonkar, »A Comprehensive Review of Performance Testing Methodologies and Best Practices: Software Quality Engineering«, *International Journal of Science and Research (IJSR)*, let. 12, str. 2008–2014, nov. 2023, doi: 10.21275/SR23822111402.
- [34] H. Sarojadevi, »Performance Testing: Methodologies and Tools«, *Journal of Information Engineering and Applications*, let. Vol 1, No.5, 2011, 2011.

■

Yauhen Unuchak je magister organizator informatik in strokovnjak na področju avtomatiziranega testiranja in testiranja zmogljivosti programske opreme. Raziskovalni interesi so povezani z avtomatiziranim testiranjem, sistemi menedžmenta kakovosti, korporativnimi informacijskimi sistemi, velepodatki in napovedno analitiko. Eden od avtorjev knjige »IT-Startup: 10 nasvetov za začetnike«.

■

Mirjana Kljajić Borštnar je redna profesorica za področje informacijskih sistemov na Fakulteti za organizacijske vede Univerze v Mariboru. Njeno raziskovalno delo je usmerjeno v sisteme za podporo odločanju, odkrivanje znanja v podatkih in strojno učenje ter organizacijsko učenje. Je glavna urednica revije *Uporabna informatika*, podpredsednica Slovenskega društva INFORMATIKA, sovodja programskih odborov mednarodnega simpozija operacijskih raziskav in Blejske e-konference ter članica izvršnega odbora AI4Slovenia.

■

Tatyana Unuchak je magistrica organizatorica informatičarka in strokovnjakinja na področju razvoja spletnih in mobilnih rešitev. Raziskovalni interesi so povezani s strojnimi učenjem in z izboljšanjem procesov projektnega vodenja.

Analiza napadov na pametne pogodbe osnovane na okoljih Ethereum virtualnega stroja

Gal Gantar, Matevž Pesek

Univerza v Ljubljani, Fakulteta za Računalništvo in informatiko, Večna pot 113, 1000 Ljubljana

gg5490@student.uni-lj.si, matevz.pesek@fri.uni-lj.si

Izvelek

V zadnjem desetletju je zanimanje za kriptovalute močno naraslo. S tem je prišlo tudi do porasta kapitala, zaklenjenega v pametnih pogodbah, kar pa je hkrati povečalo tveganje za napade. Za preprečevanje napadov na pametne pogodbe je potrebna analiza programskih vzorcev in okolja, v katerem se pametne pogodbe izvajajo. V prispevku obravnavamo varnost omrežja Ethereum in njegovo vlogo v razvoju decentraliziranih aplikacij (angl. decentralised Applications – dApps). Pregledamo uporabo Ethereumovega Virtualnega stroja (EVM), programskega jezika Solidity in zbirnika pri gradnji pametnih pogodb. Poleg tega opišemo varnostne izzive, s katerimi se sooča Ethereumov ekosistem, in analiziramo tri pretekle napade na decentralizirane platforme. Prispevek prinaša tudi konkreten primer napada na pametno pogodbo, ki izkorišča ranljivost v njeni implementaciji. Rezultat prispevka je prikaz pametne pogodbe, ki ob standardni uporabi deluje pravilno, vendar vsebuje ranljivost, ki jo je mogoče izkoristiti za neavtorizirano spreminjanje njenega stanja.

Ključne besede: binarni napadi, Ethereum, kibernetika, varnost, Solidity

Analysis of smart contract exploits in Ethereum virtual machine environment

Abstract

Over the past decade, interest in cryptocurrencies has grown significantly. As a result, there has been an increase in the capital locked in smart contracts, which has simultaneously increased the risk of attacks. Preventing attacks on smart contracts requires an analysis of coding patterns and the environment in which smart contracts operate. This article examines the security of the Ethereum network and its role in the development of decentralized applications (dApps). We review the use of the Ethereum Virtual Machine (EVM), the Solidity programming language, and the compiler in the construction of smart contracts. Additionally, we describe the security challenges faced by the Ethereum ecosystem and analyze three past attacks on decentralized protocols. This article also presents a concrete example of an attack on a smart contract exploiting a vulnerability in its implementation. The results of this article are the demonstration of a smart contract that functions correctly under normal use, but contains a vulnerability that can be exploited to alter its state without authorization.

Keywords: binary exploit, cyber security, Ethereum, Solidity

1 UVOD

Kriptovaluta **Ethereum** je prinesla revolucijo v načinu, kako razumemo in uporabljamo tehnologijo veriženja blokov. Ethereum, ki je bil ustanovljen leta 2015, je zasnovan kot odprtokodna platforma, ki omogoča razvijalcem, da zgradijo in izvajajo de-

centralizirane aplikacije. Ethereum je platforma, na kateri lahko vsak uporabnik neodvisno od centralnih avtoritet razvija in izvaja decentralizirane aplikacije ter pametne pogodbe. To omogoča in spodbuja inovacije na področju decentraliziranih finančnih storitev, digitalnih identitet ter drugih aplikacij [1, 2].

Osnova Ethereumovega ekosistema je **Ethereumov virtualni stroj** (angl. Ethereum Virtual Machine – EVM). EVM je virtualno okolje, ki omogoča izvajanje posebnih programov oziroma pametnih pogodb, ki so napisani v notranjem jeziku Ethereum, imenovanem **Solidity**. Solidity je visokonivojski jezik, ki je bil posebej zasnovan za pisanje pametnih pogodb in je zelo podoben JavaScriptu, vendar pa zaradi cene računanja na Ethereumovih pametnih pogodbah postaja vse pogostejša praksa prepisovanje delov pametnih pogodb v Solidity zbirnik. **Solidity zbirnik** (angl. Solidity assembly oziroma Yul) je nizkonivojski programski jezik, v katerem programer neposredno komunicira s pomnilnikom, kar omogoča pisanje bolj optimiziranih in posledično cenejših aplikacij [13, 19].

Z vpeljavo pametnih pogodb smo v zadnjih letih priča vrsti varnostnih izzivov, saj so pametne pogodbe tarča napadalcev, ki iščejo načine za neavtorizirano spreminjanje njihovega stanja. Napadi na pametne pogodbe, napisane v Solidityju, postajajo vse pogostejši, kar ogroža varnost in stabilnost celotnega Ethereumovega ekosistema. Na pametnih pogodbah rešitev na Ethereumu z visokim prometom ali veliko kapitalizacijo je pogosto zaklenjenih več milijard dolarjev, zaradi česar so tovrstne pogodbe izjemno privlačne za napadalce [5].

V tem prispevku se osredotočamo na raziskovanje varnostnih izzivov, povezanih z EVM in Solidity pametnimi pogodbami. V drugem poglavju predstavimo EVM okolje. Nato se v tretjem poglavju osredotočimo na pregled napadov na platforme Ronin Network, Poly Network in Coincheck, ki so se zgodili v preteklosti, ter analiziramo njihove posledice in razpravljamo o možnih strategijah za preprečevanje takšnih incidentov v prihodnosti. V tretjem poglavju prikažemo primer napada na pametno pogodbo, napisano v zbirniku. Članek zaključimo s kratkim povzetkom vplivov varnostnih lukenj na tehnologijo veriženja blokov in povzamemo lasten pogled na prihajajoče izzive v prihodnosti.

2 OSNOVE ETHEREUMA IN EVM OKOLJA

Takoj za Bitcoinom je Ethereum druga najbolj priljubljena kriptovaluta na svetu glede na tržno kapitalizacijo [7]. Ethereum omrežje je sestavljeno iz številnih **vozlišč** (angl. nodes), ki jih predstavljajo validatorji, povezani v omrežje. Vsak validator neodvisno izvaja operacije v EVM, kar omogoča delovanje pametnih

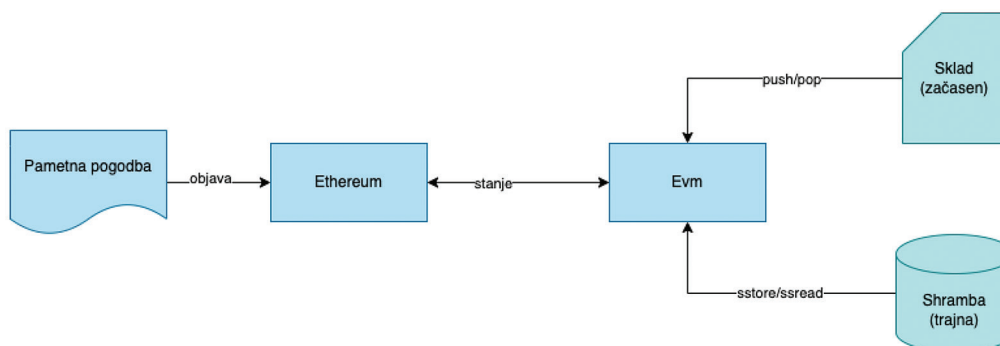
pogodb. Ta decentralizirana narava omrežja zagotavlja njegovo varnost in odpornost proti napadam. Validatorji so odgovorni za preverjanje in potrjevanje transakcij, ki se zabeležijo v blokih na verigi [9].

Solidity programi se prevedejo v binarno kodo, ki se nato izvede znotraj EVM. Ta proces izvajanja vključuje tudi interakcijo z globalnim stanjem Ethereumovega omrežja, kar omogoča, da pametne pogodbe berejo in pišejo podatke, ki so shranjeni v vsakem validatorju znotraj Ethereumovega omrežja (septembra leta 2023 je imel Ethereum več kot 800 tisoč aktivnih validatorjev [14]).

EVM je osrednji del Ethereumovega ekosistema, ki omogoča izvajanje pametnih pogodb. Vsaka pametna pogodba je program, ki predstavlja determinističen dogovor med pogodbo in uporabniki. Ko je pogodba enkrat objavljena na verigi blokov, jo je nemogoče popravljati ali spreminjati. Interakcije s pogodbo uporabniki dosežejo s klicanjem funkcij pogodbe. Primer pametne pogodbe je program, ki ima funkcijo *izplacajNagrada(Dokaz p)*, v katero uporabnik kot parameter poda dokaz, da je s svojim računalnikom pripomogel k treniranju nevronske mreže, pametna pogodba pa nato preveri verodostojnost podanega dokaza in izplača nagrado uporabniku.

Podatki v EVM se shranjujejo v dveh glavnih oblikah: v pomnilniku (angl. memory) in v shrambi (angl. storage), kot je prikazano na sliki 1. Pomnilnik je začasen in se izbriše med transakcijami, medtem ko je shramba trajna in se uporablja za shranjevanje stanja pametnih pogodb. Shema na sliki 1 prikazuje proces, ki se izvede na vsakem od validatorjev, rezultati tega procesa (spremenjeno stanje shrambe) pa se nato zapišejo v bloke. Vsaka operacija, ki se izvaja v EVM, stane določeno količino **goriva** (angl. gas). Gorivo je notranja enota za merjenje dela, ki se izvaja, in se uporablja za preprečevanje zlorab omrežja. Uporabniki plačajo transakcijske stroške v obliki goriva, ki se nato pretvori v Ethereumovo domačo valuto ETH. Nekatere transakcije, kot so na primer klici funkcij pametnih pogodb s kompleksno logiko, porabijo večje količine goriva in so zato dražje [21].

V zadnjem desetletju je zanimanje za decentralizirane aplikacije močno naraslo, pri čemer je Ethereum postal osrednja platforma za razvoj in izvajanje teh aplikacij. Ključni dejavnik v tej rasti so veliki protokoli in platforme, ki so bili razviti na Ethereumu in so postali ključni gradniki ekosistema. V tabeli 1 je predstavljenih sedem največjih rešitev glede na



Slika 1: Shema izvajanja procesa znotraj EVM

Tabela 1: Tabela z osnovnimi finančnimi podatki o največjih rešitvah na Ethereumu [8]

Rešitev	Uporaba	Vrednost žetona [€]	Zaklenjena vrednost [€]	Dnevni volumen [€]
LIDO	zastavljanje	2.43B	43.9B	17.32M
EigenLayer	zastavljanje	/	10.17B	/
Aave V3	izposoja	1.8B	7.76B	1.17M
MakerDAO	kovanec DAI	3.31B	6.67B	/
Uniswap V3	trgovanje	8.59B	5.07B	3.79B
Summer.fi	izposoja	5.26B	4.48B	951.95k
RocketPool V3	zastavljanje	582.7M	4.47B	404.39k

zaklenjeno vrednost (angl. total value locked – TVL) skupaj z vrednostjo žetona (angl. market capitalization – CAP) in dnevni volumenom. Kot je razvidno iz tabele, je v nekaterih pametnih pogodbah zaklenjeno več milijard dolarjev. Kljub vsem varnostnim ukrepom, bi zlonamerni napadalec lahko odkril ranljivost, ki bi mu omogočila klicanje funkcij za prenašanje zaklenjenih sredstev. V tem primeru bi napadalec lahko ukradel ogromno vsoto denarja.

Izmed vseh pogodb na Ethereumu je 72,9% visoko aktivnih pogodb namenjenih funkcijam žetonov, kar predstavlja izjemno tržno kapitalizacijo v višini 12,7 milijarde dolarjev. Izmed teh je 60,4% preverjenih pogodb o žetonih, ki se pogosto uporabljajo za upravljanje delnic podjetja, na primer prek začetnih ponudb kovancev (angl. initial coin offering – ICO) [16]. Pogodbe tega tipa običajno uporabljajo preverjeni ERC-20 standard za implementacijo žetona, ki implementira osnovno funkcionalnost pošiljanja in prejemanja žetonov v 312 vrsticah kode Solidity, brez uporabe zbirnika. ERC-20 je najpopularnejša pametna pogodba na Ethereumu – predstavlja kar 45,9% vseh objavljenih pametnih pogodb, vsak izmed prvih 92 ERC-20 žetonov ima skupno vrednost večjo

od 1 milijarde dolarjev [20, 6]. ERC-20 standard, ki ga je razvilo in pregledalo eno izmed največjih podjetij za iskanje ranljivosti v pametnih pogodbah, OpenZeppelin, je v uporabi že od leta 2015 in ga večina razvijalcev obravnava kot popolnoma varno pametno pogodbo [17].

23% vseh Ethereumovih pogodb vsebuje vsaj 1 blok Solidity zbirnika [3]. Uporaba zbirnika je v primerjavi z uporabo navadne Solidity kode manj spominsko varna in zahtevnejša za razvijalce. Med razvijanjem pametnih pogodb je standardna praksa izogibanje neposrednemu dostopanju do pomnilnika, če to ni absolutno nujno. Ne glede na trud, vložen v preverjanje kode, napisane v zbirniku, izvajanje strojnih ukazov na pogodbi predstavlja varnostno luknjo.

Razvijalci vse pogosteje vključujejo zbirnik v svoje pogodbe, kar je povezano tudi z razvojem jezika Solidity, saj vsaka nova posodobitev v načinu dostopanja in pisanja v pomnilnik odpre možnosti za nove optimizacije. Uporaba zbirnika v pametnih pogodbah je začela strmo naraščati že leta 2015, ko je rast vrednosti žetona ETH povzročila, da so narasle tudi cene transakcij. Višja cena transakcij je sprožila

vse večjo potrebo po optimizaciji pametnih pogodb. Trend kaže, da razvijalci pri prevajanju pogodb, ki vsebujejo zbirnik, pogosto uporabljajo najnovejše različice prevajalnika Solidity. To kaže na potrebo po izkoriščanju najnovejših funkcionalnosti in izboljšav, ki jih prinašajo posodobitve jezika [3].

Pogodbe, ki vsebujejo zbirnik, imajo običajno več transakcij, edinstvenih klicateljev in vrstic kode v

primerjavi s pogodbami brez zbirnika. To nakazuje, da je zbirnik pogosteje uporabljen v pogodbah s kompleksno logiko, kjer je cena transakcij visoka. Toda prenosi žetonov so pogostejši v pogodbah, ki ne uporabljajo zbirnika. Z vidika potencialnega napadalca to pomeni, da bi ob morebitni varnostni luknji manj verjetno lahko klical funkcije, ki so odgovorne za prenos žetonov in s tem ukradel sredstva zaklenjena na pogodbi. Pomembno je omeniti, da so pogodbe z zbirnikom običajno večje od povprečnih pogodb, s povprečjem 248,94 vrstic kode v primerjavi s povprečno 62,36 vrsticami pri pogodbah brez zbirnika. Poleg tega so pogodbe, ki prenašajo nezamenljive žetone (angl. non-fungible tokens – NFT), nagnjene k uporabi zbirnika pogosteje kakor pogodbe, ki prenašajo ERC-20 žetone [3]. Zaradi večje varnosti razvijalci za svoj produkt pogosto objavijo dve ločeni pogodbi. Prva pogodba skrbi za interno logiko produkta, druga pa je ločen ERC-20 kovanec, ki nima nobene dodatne logike in ne uporablja zbirnika. Na tak način so stanja kovancev uporabnikov shranjena na ločeni pogodbi, do katere ne moremo neposredno dostopati iz okolja prve.

Več kot tri četrtine vseh blokov zbirnika (80,86%) vsebuje vsaj en ukaz za pridobivanje in shranjevanje informacij iz shrambe (SLOAD, SSTORE).

V primeru napada ali napačnega delovanja zbirnika bi torej v večini primerov ogrozili tudi trajno stanje pogodbe, saj bi se spremenjene vrednosti v pomnilniku zapisale v shrambo. Operacije nadzora toka (ISZERO, JUMP) so prisotne le v 24,17% fragmentov, kar je pričakovano, saj preverjanje logičnih pogojev ni računsko zahtevno in ga ni smiselno prepisovati v zbirnik [3].

Citirana sorodna dela na področju analize varnosti Ethereumovih pametnih pogodb in Solidity zbirnika se osredotočajo na širšo analizo načina uporabe zbirnika. Z vidika varnosti je pomembno, da poznamo, kakšne so programerske prakse, kateri vzorci se pojavljajo in na kakšen način se uporabljao pametne pogodbe, ki vključujejo Solidity zbirnik. To delo

se najprej osredotoča na opis preteklih napadov na decentralizirane protokole, v drugem delu pa opisujemo specifičen način napada na pogodbo v Solidity zbirniku z uporabo nedovoljene modifikacije vrednosti registra ter obrambo pred tovrstnim napadom.

3 PRETEKLI NAPADI NA DECENTRALIZIRANE PROTOKOLE

V tem poglavju obravnavamo tri odmevne primere napadov na decentralizirane protokole, ki so povzročili velike finančne izgube in opozorili na nekatere ključne ranljivosti v ekosistemu kriptovalut. Prvi primer je napad na Coincheck, japonsko borzo kriptovalut, ki je bila leta 2018 tarča napada zaradi pomanjkljivih varnostnih ukrepov in uporabe vročih denarnic. Drugi primer je napad na Poly Network leta 2021, kjer so hekerji izkoristili slabše upravljanje dostopnih pravic med pametnimi pogodbami. Tretji primer pa je napad na Ronin Network Bridge leta 2022, ko je prišlo do kompromitacije zasebnih ključev v modelu Proof of Authority. Vsi trije primeri poudarjajo ranljivosti decentraliziranih protokolov ter potrebo po strožjih varnostnih ukrepih pri shranjevanju zasebnih ključev in upravljanju dostopnih pravic.

3.1 Coincheck

Coincheck je vodilna borza s kriptovalutami na Japonskem. Napad, ki se je zgodil v začetku leta 2018, je povzročil, da je borza takrat utrpela za približno 533 milijonov dolarjev škode. Več kot 500 milijonov žetonov NEM je bilo prenesenih na 19 različnih naslovov napadalcev znotraj omrežja. Napadalci so uspeli napasti borzo preko elektronskih sporočil, ki so vsebovala viruse, kar jim je omogočilo dostop do zasebnih ključev borze. S pridobljenimi zasebnimi ključi je bil postopek kraje sredstev za napadalce izjemno enostaven, še posebej, ker Coincheck ni uporabljal avtorizacijskih pametnih pogodb ali tehnologije večjih podpisov. Hkrati je podjetje vse kovance shranjevalo v isti vroči denarnici (tip denarnice, pri katerem je zasebni ključ shranjen neposredno v spominu naprave, ki izvede proces podpisovanja), kar je krajo sredstev še dodatno olajšalo. Coincheck je novembra 2018 po napadu spet začel ponujati polne storitve in še vedno posluje [4].

3.2 Poly Network Bridge

Podobno kot Ronin Network je tudi Poly Network decentraliziran most med različnimi omrežji. 10. av-

gusta 2021 je Poly Network poročal, da je neznani napadalec vdrl v pametno pogodbo omrežja, prenesel ekvivalent približno 610 milijonov dolarjev (predvsem v žetonih ETH, BNB in USDC) ter jih premaknil na svoje denarnice. Po poročanju kibernetkega podjetja SlowMist in varnostnega raziskovalca Kelvina Fichterja je bil napad omogočen zaradi nepravilnega upravljanja dostopnih pravic med dvema pomembnima pametnima pogodbama znotraj Poly Networka. Prva je EthCrossChainManager, druga pa EthCrossChainData. EthCrossChainData je izjemno privilegirana pogodba, ki naj bi bila namenjena izključno klicem lastnikov. Razlog je, da je ta pogodba odgovorna za nastavitev in upravljanje seznama javnih ključev avtentikacijskih naslovov, ki upravljajo denarnice v osnovnih likvidnostnih bazenih. Ker je napadalcu uspelo poklicati funkcijo putCurEpochConPubKeyBytes znotraj EthCrossChainData, napad na privatni ključ avtentikacijskega naslova ni bil potreben. Napadalec je tako lahko preprosto zamenjal javni ključ avtentikacijskega naslova in s tem dobil pravice do razpolaganja s sredstvi v likvidnostnih bazenih [18, 12].

3.3 Ronin Network Bridge

Ronin Network je vmesno decentralizirano omrežje (most), zgrajeno na platformi EVM in uporabljeno v večjem omrežju Axie Infinity. 23. marca 2022 je v napadu na omrežje napadalec ukradel za približno 624 milijonov dolarjev sredstev v obliki žetonov ETH in USDC. Ključni dejavnik napada je bila kompromitacija zasebnega ključa, ki so jo napadalci izkoristili za ponarejanje lažnih dvigov in krajo sredstev iz omrežja. Napad je bil izveden preko modela Proof of Authority, ki je zahteval pet od devetih potrditev, ki jih je napadalec uspešno ponaredil in s tem pridobil nadzor nad omrežjem. Po napadu je ekipa omrežja napovedala ukrepe, kot je povečanje števila potrditev transakcij na osem od devetih validacij. Rešitev za zmanjšanje tveganja tovrstnega napada bi bila uporaba stroge varnostne prakse, kot je hranjenje ključev na denarnicah, ki niso povezane z internetom, in uporaba fizičnih večpodpisnih denarnic [18].

4 PRIKAZ NAPADA NA PAMETNO POGODBO NAPIŠANO V SOLIDITY ZBIRNIKU

V tem poglavju obravnavamo napad na pametno pogodbo UnsafeMemoryWrite, ki temelji na nepravilnem upravljanju spomina v zbirniku Solidity. Implementacija te pametne pogodbe ob pričakovani upo-

rabi deluje brezhibno in pri prevajanju ne povzroča nobenih napak ali opozoril, vendar pa zlonamerna uporaba vstopnih parametrov omogoča nepooblaščen spreminjanje shranjenih podatkov v pametni pogodbi. V praktičnem delu analiziramo, kako zloraba parametrov funkcije changeStoredValues omogoča napadalcu, da prepíše stanje uporabnika in si pridobi nedovoljeno finančno korist. Poleg tega predstavimo, kako se tovrstnim napadom lahko izognemo z ustreznimi obrambnimi tehnikami.

4.1 Metodologija

Za implementacijo pametne pogodbe UnsafeMemoryWrite je bil uporabljen Solidity zbirnik. Za razvoj in objavo pogodbe na testnem omrežju Sepolia je bilo upodobljeno orodje Hardhat [11]. Postopek razvoja je vključeval uporabo Solidity zbirnika za simulacijo nedovoljenih operacij s spominom. Za simulacijo in analizo napada je bil uporabljen raziskovalec blokov Etherscan za spremljanje interakcij z verigo blokov ter orodje Hardhat Tracer za spremljanje stanja sklada in shrambe pogodbe med izvajanjem [10, 22]. Za analizo varnostnih lukenj pogodbe ter pridobitev strojne kode je bilo uporabljeno orodje Dedub, napredno orodje za analizo ranljivosti pametnih pogodb [15]. Simulacija napada je vključevala pošiljanje zlonamernih parametrov funkciji changeStoredValues, kar je povzročilo nenamerno spremembo uporabnikovega stanja. Na podlagi teh opazovanj smo ugotovili kritične varnostne pomanjkljivosti, ki omogočajo napad. Na koncu smo s pomočjo analize delovanja pametne pogodbe prikazali načine za obrambo pred prikazanim napadom.

4.2 Pametna pogodba UnsafeMemoryWrite

Praktični del tega prispevka predstavlja implementacija pametne pogodbe UnsafeMemoryWrite, ki vsebuje Solidity zbirnik. Celotna koda pametne pogodbe je prikazana na sliki 2. Pametna pogodba z identično implementacijo je objavljena na testnem omrežju Sepolia [10]. Namen pogodbe je prikazati implementacijo zbirnika, ki v primeru pravilnih vhodnih podatkov deluje kot je pričakovano, vendar pa ob zlonamerno izbranih parametrih napadalcu omogoča nedovoljeno spreminjanje stanja. Ko pogodbo prevedemo in objavimo, nobeno od standardnih orodij ne javi opozorila ali napake. Implementacija je neodvisna od različice Solidity prevajalnika in deluje tudi v najnovejših različicah.


```

// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract UnsafeMemoryWrite {
    uint256 public userBalance = 1;

    function initializeValues(uint256 replacementStart, uint256 a, uint256 b) public {
        uint256 globalMemoryStart;
        assembly {
            globalMemoryStart := mload(0x40)
            mstore(add(mload(0x40), 0x00), add(sload(userBalance.slot), 1))

            mstore(add(mload(0x40), 0x20),
                ← 0x583facc679f76c52a1127a11024efb42583facc679f76c52a1127a11024efb42)
            mstore(add(mload(0x40), 0x40),
                ← 0x8d4e5f9b3a6c8e1f0b2d9c8a7e6f5d8c583facc679f76c52a1127a11024efb42)
            mstore(add(mload(0x40), 0x60),
                ← 0x9a1b2c3d4e5f6a7b8c9d0e1f2a3b4c5d583facc679f76c52a1127a11024efb42)

            mstore(0x40, sub(add(mload(0x40), 0x80), replacementStart))
        }

        changeStoredValues(a, b);

        assembly {
            sstore(userBalance.slot, mload(globalMemoryStart))
        }
    }

    function changeStoredValues(uint256 a, uint256 b) public pure {
        assembly {
            mstore(add(mload(0x40), 0x00),
                ← 0xe1f2a3b4c5d6e7f8a9b0c1d2e3f4a5b6583facc679f76c52a1127a11024efb42)
        }
        add(a, b);
    }

    function add(uint256 x, uint256 y) public pure returns (uint256) {
        uint256 memoryStart;
        uint256 res;
        assembly {
            memoryStart := mload(0x40)

            mstore(add(memoryStart, 0x00), add(x, y))
            res := mload(add(memoryStart, 0x00))

            memoryStart := sub(memoryStart, 0x20)
        }
        return res;
    }
}

```

Slika 2: Izvorna koda UnsafeMemoryWrite

4.3 Implementacija UnsafeMemoryWrite

Pogodba ima eno spremenljivko `userBalance`, ki predstavlja uporabnikovo stanje internega žetona. V resničnem primeru bi uporabnik na menjalnici lahko ta žeton zamenjal za drugo valuto, na primer ETH ali ameriške dolarje. Funkcija `initializeValues` najprej poveča uporabnikovo stanje za 1 (v praksi bi to lahko bila nagrada za računsko delo), potem pa v spomin zapiše trikrat po 32 bajtov spomina (na primer inicializacijski podatki za Pedersonovo zgoščevalno funkcijo). Na koncu funkcija spremeni vrednost registra `0x40` na vrednost parametra `replacementStart` tako, da se naslavljanje lokacije spremembe dogaja od konca proti začetku. Register `0x40` je odgovoren za shranjevanje kazalca na naslednji prosti bajt, kamor

lahko alociramo spomin v skladu. Funkcija `changeStoredValues` zamenja blok bajtov na lokaciji `replacementStart` z drugimi vrednostmi in kliče funkcijo `add`. Funkcija `add` je standardna implementacija seštevanja v zbirniku.

4.4 Primer pravilnega delovanja

Interakcije s pogodbo so mogoče preko raziskovalcev blokov Etherscan ali Blockscout [10]. Če na začetku kličemo funkcijo `performCalculations` s parametri `replacementStart=0x20`, `a=10` in `b=20`, se v spomin zapišejo pravi podatki, zamenjava bloka bajtov se zgodi na predzadnjem mestu in rezultat funkcije `add(10, 20)` je 30. Na koncu izvajanja se je `userBalance` povečal z 1 na 2.

```

...
0x196: PUSH32    0x9a1b2c3d4e5f6a7b8c9d0e1f2a3b4c5d583facc679f76c52a1127a11024efb42
0x1b7: PUSH1     0x60
0x1b9: PUSH1     0x40
0x1bb: MLOAD
0x1bc: ADD
0x1bd: MSTORE
0x1be: DUP4
0x1bf: PUSH1     0x80
0x1c1: PUSH1     0x40
0x1c3: MLOAD
...
0x1c6: PUSH1     0x40
0x1c8: MSTORE
0x1c9: PUSH2     0x1d2
0x1cc: DUP4
0x1cd: DUP4
0x1ce: PUSH2     0xff
0x1d1: JUMP
0x1d2: JUMPDEST
0x1d3: DUP1
0x1d4: MLOAD
0x1d5: PUSH0
0x1d6: SSTORE
...

```

Slika 3: Strojna koda MemoryWriteExample

4.5 Napad na UnsafeMemoryWrite

Med izvajanjem pogodbe je na začetku v spominu shranjen `userBalance` (32B) in takoj za njim inicializacijske vrednosti (3x32B). Če kot parameter `replacementStart` podamo vrednost `0x80`, se bo register `0x40` nastavil tako, da bo ob klicu `changeStoredValues` kazalec na naslednji prosti bajt kazal na shranjeno vrednost uporabnikovega stanja. Ko se bo poklicala funkcija `add`, bo njen vmesni rezultat prepisal prejšnjo vrednost uporabnikovega stanja. Ker sta parametra `a` in `b` prosto nastavljiva, ju lahko napadalec nastavi na visoki vrednosti in na tak način nedovoljeno poveča svoje stanje. Klicanje funkcije `initializeValues` s parametri `replacementStart=0x80`, `a=1000` in `b=1000` spremeni stanje pogodbe tako, da je nova vrednost `userBalance` enaka 2000. Če pogledamo strojno kodo prevedene pogodbe na sliki 3, vidimo, da se vrednost na `0x40` spremeni na vrednost podanega parametra ter da se ob koncu klicanja funkcij nova vrednost uporabnikovega stanja shrani v shrambo pogodbe z ukazom `SSTORE`. Prevedena strojna koda je bila iz omrežja Sepolia pridobljena s pomočjo orodja Dedaub – naprednega orodja za iskanje ranljivosti v pametnih pogodbah, ki prav tako ni javilo nobenega opozorila ali napake ob pregledu objavljene pametne pogodbe [15].

4.6 Obramba pred napadom

Obramba pred opisanim napadom je mogoča na več nivojih. Vsakič, ko je zaloga vrednosti parametra omejena, je potrebno z ukazom `require` preveriti, ali

je parameter znotraj dovoljenih omejitev, preden ga začnemo uporabljati. Prav tako bi bila dobra praksa shranjevanje uporabnikovega stanja zunaj spomina, dostopnega na zbirniku, na primer v ločeni ERC-20 pogodbi, do katere znotraj okolja zbirnika nimamo dostopa. Zadnja rešitev varnostne luknje pa bi bila prepoved spreminjanja vrednosti registra `0x40` znotraj zbirnika ter izvajanje pisanja in branja iz spomina relativno na `globalMemoryStart`. Tak način programiranja zahteva več pozornosti razvijalcev, vendar pa eliminira možnost, da bi se v register `0x40` zapisala neveljavna vrednost.

5 SKLEPNE UGOTOVITVE IN ZAKLJUČEK

Raziskave varnostnih vprašanj, povezanih z EVM in Solidity pametnimi pogodbami, so ključne za izboljšanje varnosti in stabilnosti celotnega Ethereumovega ekosistema. Analiza preteklih napadov na decentralizirane platforme, kot so Coincheck, Poly Network Bridge in Ronin Network, ponuja pomembne uvide o možnih ranljivostih in potrebnih ukrepih za preprečevanje podobnih incidentov v prihodnosti.

Napad na Coincheck je dokaz, da uspešen napad ni nujno posledica napake v zasnovi protokolov, ampak tudi socialnega inženiringa. Kljub varnostnim mehanizmom je šibkost vsake organizacije lahko tudi oseba, ki v trenutku nepozornosti odpre priponko sumljivega elektronskega sporočila. Napad velike razsežnosti, kakršen se je zgodil na Coincheck, je po navadi posledica kombinacije večih ranljivosti, ki jih napadalec izkoristi hkrati. Varnost je potrebno

izvajati na vseh nivojih, saj na tak način v primeru varnostne luknje na enem od nivojev morebitnega napadalca čim bolj omejimo.

V primeru Poly Network Bridge je bil ključni del napada nepreverjena povezava med posameznimi komponentami. Vsaka pametna pogodba, ki je sestavljala omrežje, je bila večkrat pregledana s strani različnih neodvisnih varnostnih podjetij. Med pregledi so bili spregledani načini interakcije med posameznimi pogodbami. Tudi če je sistem sestavljen iz komponent, ki so same po sebi varne, je potrebno vseeno preveriti tudi načine interakcije med njimi. Prav zaradi tega so vmesna omrežja, kot sta Poly Network Bridge in Ronin Network, tako pogosto tarča napadov. Med dvema sicer varnima omrežjema je pogosto, da je ravno vmesno omrežje najšibkejši člen.

Podobno kot napad na Coincheck, tudi napad na Ronin Network ni bil posledica slabo zasnovane arhitekture protokola, ampak načina upravljanja z njim. Dodatni varnostni ukrepi, kot je uporaba fizičnih večpodpisnih denarnic in zvišanje števila potrebnih potrditev, bi tudi v tem primeru lahko omejili škodo, ki jo je povzročil napad.

V pričujočem članku delu smo prikazali načine uporabe in izkoriščanja varnostnih lukenj v zbirniku na primeru lastne pogodbe. V nasprotju z ostalimi opisanimi napadi je slednji posledica slabo zasnovane pametne pogodbe. Teoretični prikaz napada je razkril kompleksnost iskanja ranljivosti v zbirniku in razkril potencialno višjo stopnjo doveznosti za ranljivosti. Razumevanje kompleksnosti uporabe Solidity zbirnika in njegovih potencialnih tveganj je ključno za razvijalce in raziskovalcem ki se ukvarjajo z izboljšanjem varnosti decentraliziranih aplikacij na platformi Ethereum.

Naše nadaljnje delo zajema analizo uspešnosti odpravljanja tovrstnih napak z orodji za statično preverjanje kode, na primer Echinda. Poleg ranljivosti, povezanih z zbirnikom, je znotraj Ethereuma še vrsta drugih varnostnih vprašanj, ki jih v tem prispevku nismo naslovili, na primer izkoriščanje urejenosti transakcij v blokkih, manipulacija časa bloka, ponovni vstop med klicanjem transakcije in napadi na orakle. Skupaj s stalnim razvojem in izboljšavami varnostnih praks so varnostni pregledi in teoretični prikazi napadov temeljno orodje za odkrivanje ranljivosti in reševanje varnostnih izzivov.

V prihodnosti bo tehnologija veriženja blokov postala še pomembnejša, saj se bo njena uporaba širila v

različne industrije, vključno s finančnimi storitvami, dobavnimi verigami, zdravstvom in upravljanjem identitete. S tem povečanjem uporabe se bo neizogibno povečala tudi vrednost sredstev, ki jih upravljajo decentralizirani protokoli, kar bo hkrati pritegnilo več pozornosti napadalcev. Pričakovati je torej, da se bodo napadi na infrastrukturo verig blokov množili, postajali bodo bolj sofisticirani in ciljali bodo na bolj kompleksne ranljivosti. To poudarja nujnost nadaljnega razvoja varnostnih praks in orodij za preprečevanje napadov, kot tudi rednega izvajanja varnostnih pregledov in revizij kode. Le s proaktivnim pristopom k varnosti lahko ohranimo zaupanje v varnost protokolov in zagotovimo stabilnost ter integriteto celotnega Ethereum ekosistema.

LITERATURA

- [1] Vitalik Buterin. Ethereum: Platform review opportunities and challenges for private and consortium blockchains. *Opportunities and challenges for private and consortium blockchains*, 2016.
- [2] Vitalik Buterin et al. Ethereum white paper. *GitHub repository*, 1:22–23, 2013.
- [3] Stefanos Chaliasos, Arthur Gervais, and Benjamin Livshits. A study of inline assembly in solidity smart contracts. *Proc. ACM Program. Lang.*, 6(OOPSLA2), October 2022.
- [4] Ben Charoenwong and Mario Bernardi. A decade of cryptocurrency 'hacks': 2011 – 2021. October 1 2021. Available at SSRN: <https://ssrn.com/abstract=3944435> or <http://dx.doi.org/10.2139/ssrn.3944435>.
- [5] Huashan Chen, Marcus Pendleton, Laurent Njilla, and Shouhuai Xu. A survey on ethereum systems security: Vulnerabilities, attacks, and defenses. *ACM Comput. Surv.*, 53(3), jun 2020.
- [6] Coingecko. <https://www.coingecko.com/>. Dostopano: 2. Maj 2024.
- [7] CoinMarketCap. Coinmarketcap. <https://coinmarketcap.com/>. Dostopano: 2. Maj 2024.
- [8] Dapp radar. <https://dappradar.com/rankings/defi/chain/ethereum>. Dostopano: Apr 29, 2024.
- [9] Ethereum. <https://ethereum.org/en/>. Dostopano: 2. Maj 2024.
- [10] Etherscan. <https://sepolia.etherscan.io/address/0xf72a4939282D4E947dA521bAA340186d8966DeEc>, če navedeni naslov ne deluje uporabiti <https://eth-sepolia.blockscout.com/address/0xf72a4939282D4E947dA521bAA340186d8966DeEc>.
- [11] Nomic Foundation. Hardhat: Ethereum development environment for professionals, 2024.
- [12] Tommaso Gagliardini. The poly network hack explained, August 12 2021. Research at Kudelski Security.
- [13] Everett Hildenbrandt, Manasvi Saxena, Nishant Rodrigues, Xiaoran Zhu, Philip Daian, Dwight Guth, Brandon Moore, Daejun Park, Yi Zhang, Andrei Stefanescu, and Grigore Rosu. Kevm: A complete formal semantics of the ethereum virtual machine. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 204–217, 2018.
- [14] Christine Kim. The most pressing issue on ethereum is validator size growth. <https://www.coindesk.com/consensus-magazine/2023/09/29/the-most-pressing-issue-on-ethe>

- reum-is-validator-size-growth/, 2023. Dostopano: Sep 29, 2023.
- [15] Oliver Lutz, Huili Chen, Hossein Fereidooni, Christoph Sender, Alexandra Dmitrienko, Ahmad-Reza Sadeghi, and Farinaz Koushanfar. ESCORT: ethereum smart contracts vulnerability detection using deep neural network and transfer learning. *CoRR*, abs/2103.12607, 2021.
- [16] Gustavo A Oliva, Ahmed E Hassan, and Zheng Ming Jiang. An exploratory study of smart contracts in the ethereum blockchain platform. *Empirical Software Engineering*, 25:1864–1904, 2020.
- [17] OpenZeppelin. OpenZeppelin Contracts – ERC20.sol. <https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC20/ERC20.sol>, Dostopano: Apr 29, 2024.
- [18] Jason Scharfman. *Decentralized Finance (DeFi) Fraud and Hacks: Part 2*, pages 97–110. Springer International Publishing, Cham, 2023.
- [19] Solidity Assembly Dokumentacija. <https://docs.soliditylang.org/en/latest/assembly.html>. Dostopano: 2. Maj 2024.
- [20] Friedhelm Victor and Bianca Katharina Lüders. Measuring ethereum-based erc20 token networks. In *Financial Cryptography and Data Security: 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18–22, 2019, Revised Selected Papers*, page 113–129, Berlin, Heidelberg, 2019. Springer-Verlag.
- [21] Dejan Vujičić, Dijana Jagodić, and Siniša Randić. Blockchain technology, bitcoin, and ethereum: A brief overview. In *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–6, 2018.
- [22] Zemse. Hardhat tracer. <https://github.com/zemse/hardhat-tracer>, 2023. Dostopano: 2024-09-18.

■

Gal Gantar je absolvent dodiplomskega interdisciplinarnega študija računalništva in informatike. Njegovo raziskovalno delo se osredotoča na kriptografijo, predvsem na dokaze brez razkritja znanja. Prav tako raziskuje področja umetne inteligence, s posebnim poudarkom na globokem učenju na grafih. Strokovno je aktiven na področju razvoja decentraliziranih protokolov in pametnih pogodb.

■

Matevž Pesek je docent in raziskovalec na Fakulteti za računalništvo in informatiko Univerze v Ljubljani, kjer je diplomiral (2012) in doktoriral (2018). Od leta 2009 je član Laboratorija za računalniško grafiko in multimedije. Od leta 2024 izvaja predmet Varnost programov.

Premikamo meje za bolnike.

Smo Sandoz,
vodilno farmacevtsko
podjetje v svetu za generična
in podobna biološka zdravila.
In smo Lek, pionirji farmacevtske industrije
v Sloveniji.

Naša strast so odličnost in vrhunska kakovost zdravil.
Navdušujejo nas biotehnološki postopki za razvoj in
proizvodnjo podobnih bioloških zdravil ter najvišji standardi
farmacevtske proizvodnje.

SANDOZ



Lek farmacevtska družba d. d.
Verovškova ulica 57
1526 Ljubljana, Slovenija
www.lek.si

Umetna inteligenca in razvoj uporabniških vmesnikov: Študija primera mobilne aplikacije X

Maksim Nikitashin

University of Primorska, Faculty of Management, Izolska vrata 2, 6000 Koper, Slovenia

max.nikitashin@gmail.com

Izvleček

Danes je umetna inteligenca postala najbolj priljubljen trend na vseh strokovnih področjih, vključno s področjem razvoja informacijskih rešitev. Njena uporaba prinaša številne koristi in priložnosti. Kljub temu pa še vedno obstaja veliko pomislekov o njeni uporabi, zlasti med strokovnim in vodstvenim kadrom. Zaradi tega je pomembno, da se na strukturiran način izpostavijo že omenjene koristi in izboljšave, najboljši način za to pa je študija primera. Ta članek predstavlja tovrstno študijo primera za prenovo uporabniškega vmesnika mobilne aplikacije. Najprej vsebuje pregled podobne študijske literature. Nato je opisan in analiziran primer tipične Android aplikacije, vključno s tehničnimi in poslovnimi podrobnostmi. V primerjavi z razvojem izvirne različice rešitve je uporaba umetne inteligence pomagala izboljšati razvojni proces v različnih pogledih. Na primer, s skoraj 50-odstotnim prihrankom časa na podlagi primerjave porabe časa enakih dejavnosti med razvojem prvotne in nove različice ter z izboljšavami splošne kakovosti končnega izdelka na podlagi primerjave števila zrušitev in težav, ugotovljenih v primerljivem času. Izkazalo se je, da so dosežene izboljšave zelo obetavne. Predstavitve primera je lahko koristna tako za tehnike, ki uvajajo orodja umetne inteligence v proces in tudi kot primer za upravičenost investicij v tovrstno integracijo za odločevalce.

Ključne besede: umetna inteligenca, mobilna aplikacija, razvoj programske opreme, testiranje programske opreme, projektni management, uporabniški vmesnik.

AI and UI development: a Case study of mobile application x

Abstract

Artificial intelligence (AI) has become the most popular trend in all professional spheres, including information solutions development. Its utilization brings a number of benefits and opportunities. However, there still many concerns about its use, especially among professionals and managers. It is therefore important to highlight the previously mentioned benefits and improvements in a structured manner, and the best way for this are case studies. The paper presents such a case study for the redesign of the user interface of a mobile application. First, it reviews the literature on similar studies. Then, a case study of a typical Android application is described and analyzed, including both technical and business-related details. Compared to the development of the original version of the solution, AI integration has helped to improve various aspects of the development process, such as design, development, and testing. For example, with nearly 50% time saved based on the comparison of the time spent on the same activities during the development of the original and the new versions as well as improvements in the general quality of the final product based on the comparison of the number of crashes and problems identified in the comparable time frame. The results and improvements attained appear to be very promising. The presentation of the case study may be useful both for the technicians implementing AI tools into the process and as an example justifying investments into such an integration for decision makers.

Keywords: artificial intelligence, mobile application, software development, software testing, project management, user interface.

1 INTRODUCTION

Artificial Intelligence (AI) has been the most significant trend in almost all areas of human life in recent years [1]. It is reshaping nearly every professional sphere. Software development is not an exception here. At the moment, it is going through significant changes tackling almost all its sub spheres from project management to the development and deployment of the applications. These changes raise both opportunities and concerns. However, some employees working in the area, primarily non-technicians see only the potential risks and problems avoiding benefits and chances the AI opens. Especially this problem is faced in Slovenia [2]. For this reason, it is important to analyze and present successful cases of AI implementation in order to show potential benefits and justify potential investments in its implementation.

Into this paper the author would like to present the case study of the successful implementation of AI-supportive tools into a mobile application development project outstanding both the most important benefits and challenges of the process. In order to keep the business secret the application name is anonymized and some particular implementation details will not be described. However, still, the most important characteristics of the app and the AI integration will be described.

2 MOBILE APPLICATIONS DEVELOPMENT

2.1 Mobile applications

Two of the most popular types of modern software are web and mobile applications. The second are specific kind of software intended to run on mobile devices usually with a specific platform. The most popular among those platforms are Android and IOS sharing more than ninety-nine percent of the market. At the moment, the market of mobile applications is already enormously big and still continues to grow.

Those apps have a lot of specifics, advantages, and disadvantages. Among the first are nativeness in general, the device features integrations, and platform specificness. Advantages coming from such specifics are faster and more stable operation, possibilities of device and platform features integration, and better security compared to web applications. Disadvantages are primarily related to the limitation to the one platform and consequently much more difficult de-

velopment and different operation systems adoption. At the moment, there exist some solutions trying to overcome those disadvantages – cross-platform frameworks, but most of them still have limitations with the device's native features integration and operating stability.

Most modern mobile applications have similar architecture of mobile user interface (UI), application programming interface (API), and local or remote server-based database. Optionally, there could be some local storage present (Figure 1).

User interfaces of Android applications can be programmed with Java or Kotlin language and XML and of IOS ones with Objective C or SWIFT. Typically, UI consists of some interactive views and some kind of navigation between them [5].

For API communication with the back-end and database, obviously, some of the standard communication protocols are utilized in order to ensure the highest effectiveness, stability, and security. Nowadays the most popular among them is REST [6]. Databases are warehouses for data of modern mobile applications. They can be either relational or non-relational also obviously called SQL and NoSQL accordingly [5].

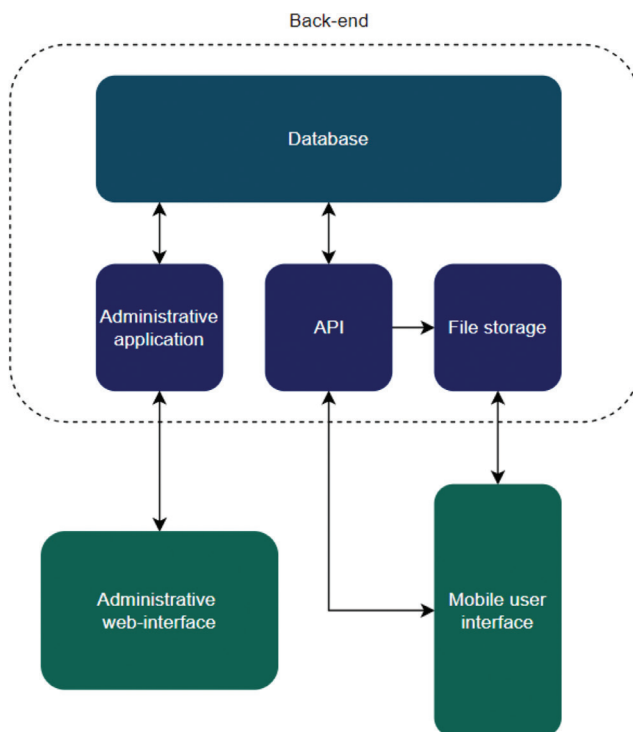


Figure 1: Typical modern mobile application architecture

2.2 Development process

Regardless of a model or a methodology selected for the development, its process usually includes the following steps:

1. Requirements definition,
2. Application design,
3. User interface prototyping and usability testing,
4. Code development,
5. Testing,
6. Publishing
7. Maintenance [5].

The most important in the context of this paper are 3., 4. and 5.

User interface design and user experience play very important roles in software development. It influences how users perceive an application and can significantly improve satisfaction with the app and conversion rate in the case of openly distributed ones [3]. The process usually includes first some prototype creation, based on the defined requirements. Then follows evaluation and testing. Later based on the previous step results corrections are made and the evaluation is repeated. The last step repeats until the appropriate version is not prepared [4]. Today there are a number of tools like Figma or Adobe XD intended to support the entire process and make it much more effective and easier.

Nowadays code development process commonly includes the utilization of integrated development environments (IDE) including for example code support and project management tools helping to easier and faster the process. Examples are Visual Studio Code or Android Studio [5]. Alongside, as code storage spaces repositories with integrated continuous integration, delivery, and deployment (CI/CD) pipelines are widely used. Such tools let save and share the code as far as to automatically build and test a new application version before deployment. Thus, they help to improve collaboration and ensure the maintenance of code quality and stability. Popular examples among the repositories are GitHub and GitLab and among the pipelines are Jenkins and GitHub Actions [7].

Testing is the last stage of the development before the deployment. It is intended for quality assurance and identification of potential problems before publishing. Usually, it is split into functional and non-functional. The first one is the most important in the

case of mobile applications. Standardly it includes UI, API, and service functionalities testing [8]. Nowadays testing can be performed manually by human testers or with the help of automated testing tools. The second utilizes predefined automated tests to perform some actions and thus analyze if the application behaves as required and expected. Examples of such tools are UI Automator, Robotium, and Espresso [9].

3 AI SUPPORTIVE TOOLS

AI-supportive tools have found a wide range of applications both in the software development sphere in general and in mobile development in particular. In terms of UI/UX development, the most popular and well-known kinds of them are those for designing and monitoring, code support, and testing.

3.1 Design tools

AI tools have already and still continue to change the UI/UX design process. They are capable of analyzing such amounts of data, that were impossible for humans before, enabling the creation of personalized user experiences and reducing errors and overlooked opportunities amount.

Those changes influence the nature of the work of UI/UX designers. From primarily creative roles they are now playing more business-related strategic roles of interpreters of AI insights and decision-makers. Researches show that the easiness and speed of ideation and prototype creation have now increased significantly.

The widely utilized AI tools range in UI/UX design include for example sentimental analytical, predictive, and generative ones [10].

Among previously mentioned tools the most popular are for example Framer, Uizard, ChatGPT, and in the case of some other popular platforms custom AI-supportive plugins. Most of them are pretty easy and fast to integrate even into an existing project for example using import of projects [11].

3.2 Coding tools

Coding is one of the parts of the UI/UX development process highly influenced by AI support integration. In an international environment, the most popular ones are the well-known ChatGPT and GitHub Copilot.

StackOverflow research shows that currently, sixty-two percent of developers are using AI su-

port, and even more plan to start utilization in the near future. The most popular applications are documenting, coding by itself, and code testing. At the same time, most developers are not worrying about possible job displacements but about possible disinformation and misinformation [12].

Similar research by GitHub has shown an even higher AI utilization rate (82%) among all IT professionals. The most important benefits are improved code quality as far as easier new languages and codebases understanding and enhanced testing. At the same time, some developers face issues and limitations from the side of the companies they are employed [13].

The research conducted among Slovenian software development professionals has shown pretty much similar results. The most popular tools there are ChatGPT and GitHub Copilot and each two of the three specialists are utilizing those tools. However, those professionals are primarily technicians while business and management-related people are avoiding AI utilization and show much more concern about it [2].

While ChatGPT is an external tool GitHub Copilot can be integrated with all the most popular IDEs like Visual Studio code or any of JetBrains's ones in a few clicks. It can support development into all the most popular languages like Java and JavaScript/TypeScript [14].

3.3 Testing tools

Testing is the most influenced by AI integration part of software development. AI tools are nowadays utilized in different aspects of the process. The most important are security and integration parts. The last one is commonly more addressed in UI/UX design nevertheless both of them play an important role there.

In terms of security testing AI is utilized in order to improve its efficiency and efficacy. For instance, those tools can analyze code to identify potential vulnerabilities, find anomalies, analyze user behavior, or perform penetration tests [15]. One of the research projects conducted in this sphere shows some minor improvements in efficiency and huge reductions in time consumption [15]. Another research among IT companies has identified that most of them (70%) nowadays use AI support in DevOps and security testing in some form and most of them have reported improvements in security and operational efficiency after the start of utilization [16].

In terms of integration testing AI is utilized for already mentioned user behavior analysis as far as for class analysis and test case generation and development [15] [17]. There for instance resources of testing teams optimizations and reduced error risks are mentioned as potential benefits in one of the research projects [17].

The most popular tools utilized in UI testing nowadays are for example Katalon and BrowserStack [18] [19]. Most of them are pretty much easy to integrate and utilize into the development process. For example, both of the mentioned before has possibilities for instant pipeline integration with the most popular automating tools like Jenkins and GitHub Actions [20] [21] [22] [23].

4 THE APPLICATION X

Application X is a student routine supportive app. It is developed as a native Android application with an XML/Java user interface and a REST API with a relational database in its back end.

According to the customer's requirements, it was made relatively simple in order to be able to support most of the active Android versions and to ensure as stable operating as possible.

The app consists of a few relatively simple views including some standard built-in Android inputs, navigation buttons, and data holders. Also, it has some custom commonly repetitive components.

Originally UI part of the application had integrated automated end-to-end testing with the Espresso and a Jenkins build pipeline.

A few years after the application was launched the company decided to improve and modernize it. The UI part was not an exception. Visual design and functionalities needed to be significantly changed and optimized. Some views were told to be totally reworked. In order to enhance the development process the team responsible for the application has decided to integrate and start utilizing some AI-supportive tools.

5 ARTIFICIAL INTELLIGENCE INTEGRATION

First of all, the team has decided to improve the design process, in particular visual prototyping. Also, the development process itself was selected for modernization. Here code optimization and testing were identified as possible application points. Because the application is relatively simple the team did

not include any user behavior analytical tools in the list. Calculations of the time spent were prepared based on the project management and monitoring tool utilized by the team.

As mentioned before, in order to keep business secret some particular details of management of the project or the implementation can not be described in this part precisely. However, all of the data, the development company has allowed to publish are presented.

5.1 Prototyping

The development team has been utilizing the Figma prototyping tool during the development process, so first of all Autoflow plugin was added in order to improve existing prototype corrections and adoption. Then the integrated UX Pilot AI tool has been selected for prototyping enhancement with design generation and evaluation. These tools helped to speed up design corrections and the prototyping process. Development and agreement of the design of the first version of the application has taken a little bit more than two months. Designing the updated version with the same customer, the same team, and approximately the same amount of workload was completed in a little bit less than one month (Figure 2). Of course, there could have been other factors influencing the process present. However, the impact of AI tools is definitely visible too.

5.2 Implementation

After the completion of designing the new app version, the team started corrections implementation. In comparison with the first version development UI part of the development team has started to utilize GitHub Copilot in work. The efficiency increase was significant. During the original version implementation, it took almost one working week to completely implement, test, and optimize one screen view. After the start of AI support utilization, the same process in case of complete rework and reimplementation of a view required nearly three working days (Figure 2).

Additionally, the development part of the team has reported improvements in code readability, due to built-in grammar checks. Also, the utilization of AI support has helped in refactoring and optimization both of some of the existing and newly written code. This was achieved with built-in Copilot’s coding support and analysis tools.

5.3 Testing

The last development stage where the improvements were the most significant was automated testing. Here the BrowserStack was integrated into the pipeline. It has taken 3 days and went through very smoothly. Here efficiency measurement could be made by the number of identified problems which has increased more than two times. At the same time, the number of crashes after the deployment of the

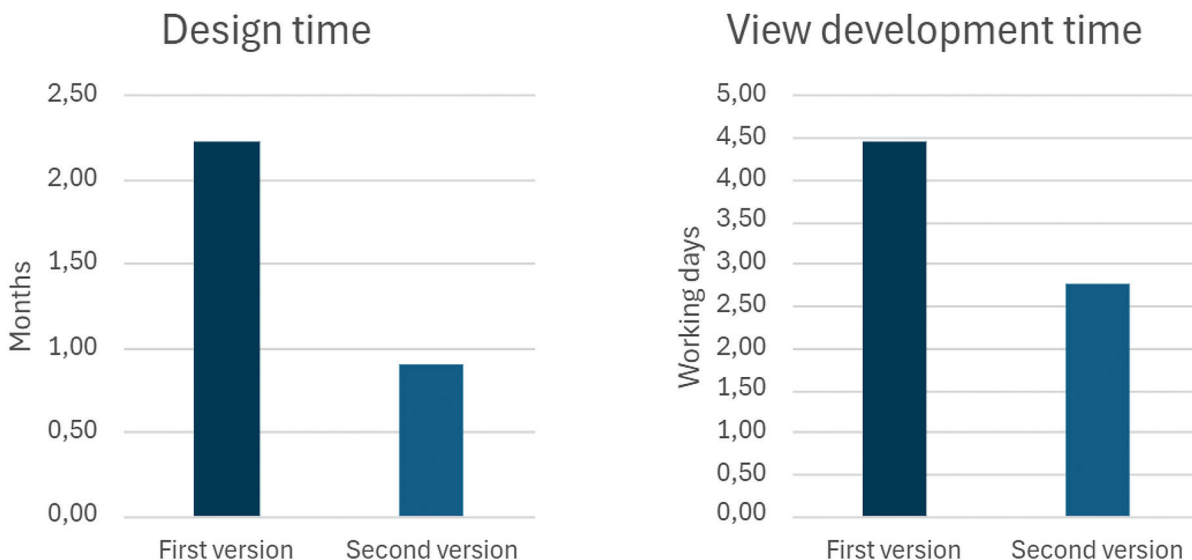


Figure 2: Design and development time comparison

second version decreased to zero. Of course, the period after deployment of the second version is less than the time of activity of the first version, but by the same period after publishing the original version already had a few crashes. An especially important benefit of the integration was that AI assistance multiple times helped to identify some very specific but realistic use cases and related issues that were not found by the development and testing responsible parts of the team nevertheless these problems were already present in the original version of the application and caused application crashes.

By the moment of writing, the time passed after the second version launch is still relatively short in order to compare the satisfaction and feedback of the users with the first version. For this reason, this factor cannot be analyzed and presented here.

6 CONCLUSION

By the end development and deployment of the second version has been finished and it has taken less than 2 months in comparison with a little bit more than 5 months in the case of the first version. The time spent on each of the steps described has decreased significantly. Of course, there are some limitations related to different amounts of workload in cases of development from the very beginning and of a huge rework. However, again the difference is very significant and the role of AI tools integration in such improvement can be assumed with very high probability. Also, a very important moment here is the improvement of the quality of the application.

In conclusion, it can be stated that AI integration into the software development process can significantly improve it both in terms of efficiency and quality. This case study can be used as a good example, for instance, to justify investments into such integration before business and management people or to motivate development teams to decide upon it.

LITERATURE

- [1] S. Singh, »15 Global Trends for 2024,« *Forbes*, Dec. 12, 2023. [Online]. Available: <https://www.forbes.com/sites/sarwantsingh/2023/12/11/15-global-trends-for-2024/>
- [2] M. Nikitashin and B. Werber, »Artificial Intelligence and Software Development in Slovenia: Adoption, Challenges, and Opportunities», submitted for publication.
- [3] C. Okonkwo, »Assessment of User Experience (UX) design trends in mobile applications,« *Journal of Technology and Systems*, vol. 6, no. 5, pp. 29–41, Aug. 2024, doi: 10.47941/jts.2147.
- [4] J. Nielsen, *Usability Engineering*. 1993. [Online]. Available: <https://dl.acm.org/citation.cfm?id=2821575>
- [5] T. C. G and A. J. Devi, »A study and overview of the mobile app development industry,« *International Journal of Applied Engineering and Management Letters*, pp. 115–130, Jun. 2021, doi: 10.47992/ijaeml.2581.7000.0097.
- [6] A. Neumann, N. Laranjeiro, and J. Bernardino, »An analysis of Public REST Web Service APIs,« *IEEE Transactions on Services Computing*, vol. 14, no. 4, pp. 957–970, Jun. 2018, doi: 10.1109/tsc.2018.2847344.
- [7] L. Četina and L. Pavlič, Življenjski cikel cevovodov neprekinjene namestitve informacijskih rešitev. 2024, pp. 47–56. doi: 10.18690/um.feri.4.2024.4.
- [8] A. Samir, H. Amin, and N. Badr, »A SURVEY ON AUTOMATED USER INTERFACE TESTING FOR MOBILE APPLICATIONS,« *International Journal of Intelligent Computing and Information Sciences/International Journal of Intelligent Computing and Information Sciences*, vol. 0, no. 0, pp. 1–11, May 2022, doi: 10.21608/ijicis.2022.98138.1124.
- [9] A. M. Sinaga, Y. Pratama, F. O. Siburian, and K. J. F. P. S, »Comparison of graphical user interface testing tools,« *Journal of Computer Networks Architecture and High Performance Computing*, vol. 3, no. 2, pp. 123–134, Jul. 2021, doi: 10.47709/cnahpc.v3i2.951.
- [10] Y. Xu, Y. Liu, H. Xu, and H. Tan, »AI-Driven UX/UI Design: Empirical research and Applications in FinTech,« *International Journal of Innovative Research in Computer Science & Technology*, vol. 12, no. 4, pp. 99–109, Jul. 2024, doi: 10.55524/ijrcst.2024.12.4.16.
- [11] GeeksforGeeks, »10 Best AI Tools for UX Designers in 2024,« *GeeksforGeeks*, May 24, 2024. <https://www.geeksforgeeks.org/best-ai-tools-for-ux-designers/>
- [12] »2024 Stack Overflow Developer Survey.« <https://survey.stackoverflow.co/2024/>
- [13] K. Daigle, »Survey: The AI wave continues to grow on software development teams – The GitHub Blog,« *The GitHub Blog*, Sep. 03, 2024. <https://github.blog/news-insights/research/survey-ai-wave-grows/>
- [14] »GitHub Copilot · Your AI pair programmer,« *GitHub*, 2024. <https://github.com/features/copilot>
- [15] B. R. Bhimanapati, S. Jain, and P. K. G. Pandian, »Security Testing for Mobile Applications Using AI and ML Algorithms,« *Journal of Quantum Science and Technology*, vol. 1, no. 2, pp. 44–58, Jun. 2024, doi: 10.36676/jqst.v1.i2.15.
- [16] N. B. Gajbhiye, N. A. Aggarwal, and N. S. Jain, »Automated Security Testing in DevOps Environments Using AI and ML,« *International Journal for Research Publication and Seminars*, vol. 15, no. 2, pp. 259–271, Jun. 2024, doi: 10.36676/jrps.v15.i2.1472.
- [17] I. Alazzam, A. M. R. AlSobeh, and B. B. Melhem, »Enhancing integration testing efficiency through AI-driven combined structural and textual class coupling metric,« *Online Journal of Communication and Media Technologies*, vol. 14, no. 4, Oct. 2024, doi: 10.30935/ojcm/15524.
- [18] GeeksforGeeks, »Top 10 AI testing tools for test Automation in 2024,« *GeeksforGeeks*, Oct. 15, 2024. <https://www.geeksforgeeks.org/top-ai-testing-tools-for-test-automation/>
- [19] »AI-Augmented Software-Testing Tools Reviews and Ratings,« *Gartner*. <https://www.gartner.com/reviews/market/ai-augmented-software-testing-tools> (accessed Nov. 05, 2024).
- [20] »Jenkins integration overview | Katalon Docs.« <https://docs.katalon.com/katalon-platform/integrations/cicd-integrations/jenkins-integration/jenkins-integration-overview>

- [21] »Katalon Studio GitHub Action | Katalon Docs.« <https://docs.katalon.com/katalon-platform/integrations/cicd-integrations/katalon-studio-github-action>
- [22] S. A. Gurudatt and B. Shrikanth, »Jenkins for Test Automation : Tutorial | BrowserStack,« BrowserStack, Sep. 01, 2022. <https://www.browserstack.com/guide/jenkins-for-test-automation>
- [23] »Integrate your Selenium test suite with GitHub Actions | BrowserStack Docs.« <https://www.browserstack.com/docs/automate/selenium/github-actions>

■

Maksim Nikitashin is a PhD student and an assistant in the field of Business Informatics at the Faculty of Management of the University of Primorska. He holds a master's degree in Organization and Management of Information Systems from the University of Maribor. Also, he has a few years of experience developing different kinds of web and mobile information solutions. The fields of his professional interests and competencies include, but are not limited to: software engineering, business informatics, and artificial intelligence.

The logo for MODRA consists of three white circles of varying sizes stacked vertically above the word "MODRA" in a bold, white, sans-serif font.

Zavarovalnica za dodatno
pokojsinsko zavarovanje



MANJ DOHODNINE. VEČ POKOJSINE.

ZAKORAKAJ Z MODRO V PRIHODNOST.

Z varčevanjem v dodatnem pokojninskem zavarovanju ste upravičeni do posebne davčne olajšave. Vplačila v posameznem letu vam znižajo osnovo za odmero dohodnine in država vam del dohodniče vrne ali pa se vam zniža morebitno doplačilo dohodnine.

IZRAČUNAJTE
DAVČNO OLAJŠAVO



Spodbude in ovire uporabe odprtih podatkov v podjetjih – pregled literature

Staša Blatnik, Andreja Pucihar, Mirjana Kljajić Borštnar
Fakulteta za organizacijske vede, Univerza v Mariboru, Kidričeva cesta 55a, 4000 Kranj
stasa.blatnik1@um.si, andreja.pucihar@um.si, mirjana.kljajic@um.si

Izveček

Koncept odprtih podatkov (OP), se je v zadnjem desetletju zelo razvil, kar se odraža v vladnih politikah in iniciativah ter kakovosti, obsegu in dostopnosti javnih podatkov. Pričakovanja, da bodo odprti podatki prispevali h kreiranju dodane vrednosti, inovacijam ter vrsti ekonomskih, družbenih in okoljskih preobrazb, so velika. Kljub pričakovanjem in potencialu, ki ga OP imajo, se kaže, da podjetja malo izkoriščajo potencial OP, saj nimajo ustreznih tehničnih in organizacijskih zmogljivosti. V raziskavi naslavljamo problem zrelosti podjetij za uporabo odprtih podatkov. Cilj je razvoj več-kriterijskega modela, s katerim bo moč oceniti organizacijske in tehnološke zmogljivosti podjetij za uporabo OP ter na podlagi ocene predlagati ustrezne aktivnosti za vzpostavitev teh zmogljivosti.

V ta namen smo opravili sistematičen pregled literature. Identificirali smo modele za merjenje zrelosti različnih vidikov OP ter spodbujevalne in zaviralne dejavnike uporabe OP. Razumevanje tega bo podlaga za identifikacijo kriterijev, s katerimi bo moč oceniti zrelost podjetij za uporabo OP.

Ključne besede: Modeli zrelosti, Ocena zrelosti, Odprti podatki, Uporaba odprtih podatkov, Več-parametrski odločitveni model, spodbujevalni in zaviralni dejavniki

Drivers and barriers of open data use in enterprises – a literature review

Abstract

Over the last decade, the concept of open data (OD) has evolved significantly, as reflected in government policies and initiatives as well as the quality, scope, and accessibility of public data. Expectations are high that OD will contribute to value creation, innovation, and a wide range of economic, social, and environmental transformations. Despite these expectations and the potential of OD, little is known about its actual use and impact on enterprises. In this study, we address the problem of assessing the maturity of enterprises in order to use OD. The objective is to develop a multi-criteria model for assessing enterprises regarding their own organizational and technological capabilities for the use of OD and propose activities to build these capabilities.

For this purpose, we conducted a systematic literature review. We identified models to measure the maturity of different aspects of OD and the drivers and barriers of OD use. Understanding this will provide a basis for the identification of criteria that can be used to assess the maturity level of enterprises to use OD.

Keywords: Maturity models, Maturity assessment, Open data, Open data use, multi-criteria decision model, drivers and barriers

1 UVOD

Koncept odprtih podatkov (OP) se je razvil pred več kot dvema desetletjema in se v Evropi začel uveljavljati po sprejetju Direktive EU o ponovni uporabi informacij javnega sektorja leta 2003 [1]. Slovenija je

tem prizadevanjem sledila z uveljavitvijo Zakona o dostopu do informacij javnega značaja [2]. Prelomnica v razvoju OP je bila uveljavitev Memoranduma o odprti državni upravi [3] v ZDA, kar je spodbudilo intenzivnejše raziskave na tem področju. Odprti po-

datki predstavljajo informatizirane zbirke podatkov, ki so dostopne z odprto licenco in v formatih, primernih za uporabo, ter ponovno uporabo [4]. Izraz »ponovna uporaba podatkov« se nanaša na uporabo podatkov za namene, za katere niso bili prvotno ustvarjeni [5], [6]. V tem prispevku izraza »uporaba« in »ponovna uporaba« uporabljamo izmenično. Večina OP izvira iz javnega sektorja, pri čemer direktive in zakonske zahteve narekujejo, da morajo biti ti podatki dostopni preko specializiranih spletnih portalov, kot je OPSI v Sloveniji ali Evropski portal podatkov. Za zagotavljanje uporabnosti OP se upoštevajo uveljavljene smernice za objavo, kot so na primer 8 principov odprte državne uprave [7] in petstopenjski model povezanih podatkov [8].

Odprti podatki poleg preglednosti delovanja vlad predstavljajo dragocen vir za ekonomski in družbeni razvoj. Podjetjem, zaposlenim in posameznikom omogočajo boljše odločanje ter razvoj novih produktov in poslovnih modelov [6], [9]. Kljub temu pa uporaba OP zahteva specifična znanja, večšine in vire, kar lahko predstavlja izziv, zlasti za majhna in srednje velika podjetja [10], [11]. Ti izzivi lahko omejujejo njihovo inovativnost in uporabo OP za povečanje dodane vrednosti [9]. Pričakuje se, da bo uporaba odprtih podatkov s strani podjetij prispevala k večji produktivnosti, konkurenčnosti, inovativnosti in prehodu v digitalno in zeleno gospodarstvo. Velikost trga odprtih podatkov je ocenjena na 184 milijard evrov in naj bi leta 2025 dosegla med 199,51 in 334,21 milijarde evrov [12]. Zato ne čudi, da vlade po vsem svetu za odpiranje podatkov namenjajo precejšnja sredstva.

Kljub napredku pri zagotavljanju kakovosti in dostopnosti OP pa je malo znanega o njihovi dejanski uporabi in učinkih, ki jih ustvarjajo [13], [14]. Kot poudarjajo Janssen in sodelavci [15], odprti podatki sami po sebi nimajo vrednosti; vrednost ustvarimo z njihovo uporabo. Različne institucije po vsem svetu merijo zrelost, uporabo in učinke odprtih podatkov, vendar večinoma z vidika institucij, ki podatke objavljajo (na primer, poročilo o zrelosti odprtih podatkov Evropske komisije (ang. Open Data Maturity report ODM) in poročilo OECD »Odprti, uporabni in ponovno uporabni podatki« (ang. »Open, Useful and Reusable data« – OURData index). Slovenija se po teh merilih uvršča med najboljše države, v letu 2023 je na OURData Indeks zasedla 7. mesto, na ODM pa 14. mesto [16], [17]. Kljub temu je uporaba OP med

podjetji še vedno nizka, saj manj kot 4 % podjetij z več kot 10 zaposlenimi ali samozaposlenimi poroča o uporabi OP [18].

Za učinkovito načrtovanje politik in strategij je ključno razumeti, kako podjetja uporabljajo OP, kakšne učinke dosegajo ter kakšne ovire se pojavljajo pri njihovi uporabi. Izhajamo iz predpostavke, da je možno razviti model, s katerim bo moč oceniti stopnjo zrelosti posameznega podjetja za uporabo odprtih podatkov glede na različne dejavnike. Ocena zrelosti bo podlaga za pripravo načrta aktivnosti, ki jih mora podjetje izvesti, da bo lahko učinkovito uporabilo odprte podatke. Množica ocen posameznih podjetij pa je lahko dragocena informacija za odločevalce za sprejemanje nadaljnjih spodbujevalnih politik. V preteklih raziskavah takšnega modela nismo zasledili, zato se izkaže potreba po razvoju več-kriterijskega modela za ocenjevanje zrelosti podjetij za uporabo odprtih podatkov. Za razvoj modela pa je treba najprej prepoznati dejavnike, ki spodbujajo oz. zavirajo uporabo odprtih podatkov v podjetjih in s tem opredeljujejo zrelost podjetja za uporabo odprtih podatkov, kar bomo naslovili v tej raziskavi. Na podlagi tega smo oblikovali raziskovalno vprašanje, ki se glasi: *Kateri dejavniki spodbujajo ali ovirajo uporabo odprtih podatkov v podjetjih?* Raziskovalno vprašanje bomo naslovili s sistematičnim pregledom literature, s poudarkom na obstoječih modelih zrelosti za merjenje različnih vidikov OP ter ovirah, ki preprečujejo uporabo OP v podjetjih. Na ta način bomo identificirali dejavnike, ki vplivajo na zrelost podjetja za uporabo OP.

V nadaljevanju prispevka bomo v drugem poglavju opisali metodologijo raziskave, v tretjem poglavju sledi pregled rezultatov sistematičnega pregleda literature, to je obstoječih modelov ter spodbujevalnih in zaviralnih dejavnikov. Nazadnje so v četrtem poglavju predstavljene ugotovitve raziskave, ki mu sledi še zaključek.

2 METODOLOGIJA

V raziskavi bomo s sistematičnim pregledom literature identificirali obstoječe modele za merjenje zrelosti podjetij ter spodbujevalne in zaviralne dejavnike, ki se pojavljajo pri uporabi OP v podjetjih. Z analizo obstoječih modelov in izpostavljenih ovir za uporabo OP bomo poskusili odgovoriti na raziskovalno vprašanje, ki se glasi:

RV: Kateri dejavniki spodbujajo ali ovirajo uporabo odprtih podatkov v podjetjih?

Sistematični pregled literature smo izvedli po digitalnih bibliografskih bazah Scopus, Web of Science, ProQuest in Google Scholar, kjer smo izvedli iskanje po naslednjih ključnih besedah in njihovih smiselnih kombinacijah.

- Odprti podatki (Open data)
- Odprti vladni podatki (Open Government Data)
- Uporaba odprtih podatkov (Open Data Use)
- Modeli zrelosti (Maturity model)
- Ocena zrelosti (Maturity assessment)
- Več-parametrski odločitveni model (Multi-criteria Decision Model)
- Spodbujevalni in zaviralni dejavniki (Drivers and barriers)

Pridobili smo 214 publikacij in po uvodni izločitvi neustreznih rezultatov, sledili še principu »snežne kepe« in identificirali dodatnih 19 publikacij. Ustreznost publikacij smo ocenili glede na izključitvene

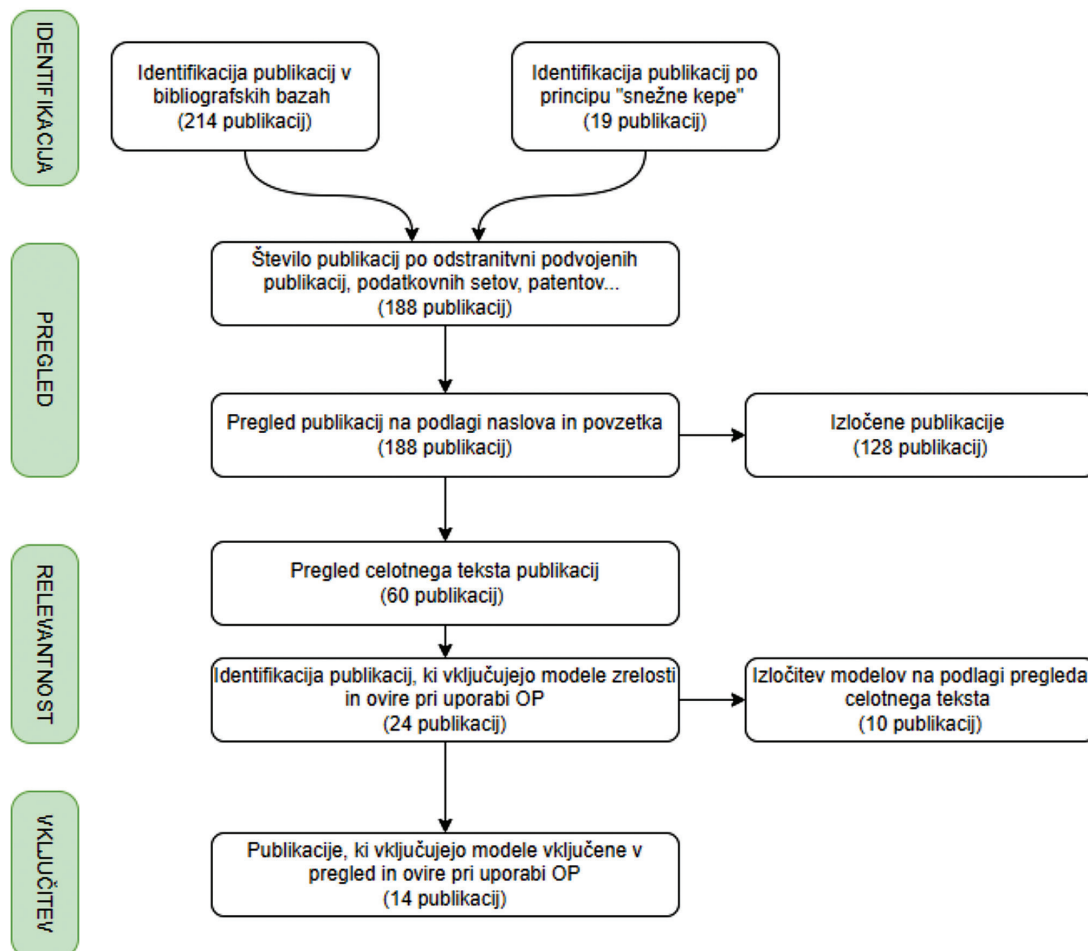
kriterije, ki so bili:

- Jezik publikacije ni angleščina ali slovenščina.
- Raziskava se ne osredotoča na OP, OP so le omejeni ali uporabljeni kot primer,
- Raziskava se osredotoča na druge ključne besede, vendar ne na OP,
- Model ne ocenjuje zrelosti OP,
- Model zrelosti na noben način ne vključuje uvedbe in uporabe OP.

Pri izboru smo se osredotočili na znanstvene publikacije, pri čemer moramo upoštevati, da nekatera ocenjevalna orodja niso vključena v znanstvene raziskave, zaradi česar niso vključena v našo analizo.

Proces izbora je prikazan na sliki 1.

Skupno smo identificirali 8 publikacij, ki vsebujejo modele in 6 publikacij, ki preučujejo spodbujevalne in zaviralne dejavnike uporabe OP, ki smo jih vključili analizo.



Slika 1: Proces izbora modelov

3 REZULTATI

Rezultati raziskave so dvojni. V prvem delu smo pregledali obstoječe modele za merjenje različnih vidikov OP, analizirali dejavnike, ki so jih avtorji izpostavili, cilje in osredotočenost predlaganih modelov. V drugem delu smo identificirali zaviralne in spodbujevalne dejavnike objave in uporabe OP ter izpostavili tiste, ki so relevantni za razvoj modela za merjenje zrelosti podjetij za uporabo OP.

3.1 Modeli za merjenje zrelosti OP

Pri pregledu literature smo identificirali 8 modelov, ki merijo različne vidike zrelosti organizacij za objavljanje in uporabo OP.

3.1.1 Indeks odprtosti e-državne uprave (ang. E-government Openness index – EGovOI)

Veljkovič in sodelavci [19] so razvili model, ki ocenjuje odprtost e-državne uprave z uporabo indeksa eGovOI. Kazalniki, ki jih ocenjuje, so prisotnost podatkovnih zbirk velike vrednosti, odprtost podatkov, preglednost (državne uprave in podatkov), vključevanje državljanov v demokratične procese in sodelovanje državne uprave z državljanji.

Indeks poda vpogled v stanje odprtosti državne uprave preko perspektive dostopa, kakovosti, uporabnosti, verodostojnosti in preglednosti objavljenih OP. Poleg tega omogoča analizo stanja odprtosti preko analize posameznih kazalnikov. Za model za ocenjevanje zrelosti podjetij za uporabo OP sta v tem modelu kazalnika »Vključevanje državljanov« in »Sodelovanje«. Ta kazalnika nakazujeta, do katere mere so ponudniki podatkov pripravljene sodelovati z uporabniki pri oblikovanju in objavljanju podatkovnih zbirk. Z vidika podjetja, ki bi uporabilo OP, ta dva kazalnika indeksa predstavljata dostopnost ponudnika za sodelovanje, prilagajanje in izboljševanje objavljenih podatkovnih zbirk.

3.1.2 Model zrelosti odprte državne uprave (ang. Open government maturity model – OGMM)

Za ocenjevanje pobud odprte državne uprave sta [20] zasnovala model OGMM. Model je zasnovan kot pomoč agencijam državne uprave za učinkovito izvajanje pobud za odpiranje svojega delovanja javnosti. Model razlikuje med procesi, povezanimi s podatki, in procesi sodelovanja in vključevanja državljanov. Za vsako stopnjo zrelosti model opredeli

specifične mejnike in poda smernice za napredovanje na naslednjo stopnjo. Najvišje stopnje institucije državne uprave dosegajo z integracijo načinov delovanja družbenih medijev na spletne strani in s tem omogočijo sodelovanje javnosti.

3.1.3 Model zrelosti OP (ang. Open Data Maturity Model – OD-MM)

Avtorji Solar in sodelavci [21] so predlagali model zrelosti za oceno predanosti in sposobnosti agencij javnega sektorja za implementacijo načel in praks OP. Skupina dejavnikov, ki z vidika državljanov in podjetnikov meri uporabo podatkov, vključuje metrike, ki posredno merijo uporabo OP. Take metrike so na primer: število objavljenih podatkovnih zbirk; obstoj enotne točke za dostop do podatkov; obstoj metrik za merjenje dostopanja do podatkovnih zbirk; število projektov, ki potekajo, ali so zaključeni; vključevanje strokovnjakov za povečevanje uporabnosti OP; in druge. Model vključuje dve skupini dejavnikov, ki lahko predstavljata vhodne informacije za model, ki meri zrelost organizacije za uporabo OP. Ti skupini sta:

- pravni in organizacijski dejavniki, kamor se uvrščajo vprašanja strateške usmerjenosti podjetja; stil vodenja, za katerega smo iz literature ugotovili, da razloži velik del variance v naklonjenosti zaposlenih za uporabo OP [22] in upravljanje podjetja, ter
- tehnološki dejavniki, ki opredeljujejo zmožnosti organizacije za neprekinjeno prevzemanje OP in razpoložljivost kapacitet, za avtomatiziran prevzem OP in njihovo vključitev v poslovne procese.

3.1.4 Model za oceno uporabnosti odprtih podatkov (ang. Metric for the evaluation of Open Data 5 – MELODA 5)

MELODA 5 [5] je model za ocenjevanje ponovne uporabnosti OP. Model predpostavi osem uteženih dejavnikov (licenciranje podatkov, dostop do informacij, tehnični standardi, stopnja standardizacije, geolociranje, frekvenca posodabljanja podatkov, razširjanje informacij in ugled vira podatkov) in ima uporabno vrednost predvsem za ocenjevanje ponovne uporabnosti podatkovnih zbirk. Mnogi modeli obravnavajo uporabnost podatkov in njihovo povezanost na podlagi uveljavljenih okvirjev, kot je na primer Berners-Leejev pet-zvezdni model povezanih podatkov. MELODA 5 poleg teh, vključuje še nekatere druge dejavnike, ki so pomembni, da so podatki

primerni za ponovno uporabo. Taki dejavniki so na primer licenciranje podatkov, tehnični standardi podatkovnih zbirk, obstoj geografske vsebine in pogostost posodabljanja podatkov.

Dejavniki, vključeni v model MELODA 5, ki so ob ustrezni prilagoditvi, primerni tudi za uporabo v modelu, ki meri zrelost podjetja za uporabo OP so: zakonski vidik; tehnični standardi, ki jih mora organizacija zagotavljati za implementacijo OP; in standardizacija z vidika poznavanja standardov OP vključno z geolokacijski podatki.

3.1.5 Stopenjski model OP (ang. OGD – A stage model)

Model, ki so ga razvili Kalampokis in sodelavci [23], je namenjen odpiranju znanstvene razprave o stopenjskih modelih OP. Avtorji zasledujejo dva cilja, prvi je predstaviti načrt za omogočanje ponovne uporabe OP državne uprave, drugi cilj je omogočiti oceno prefinjenosti pobud za objavljane podatkov državne uprave.

Model se osredotoča na objavo podatkov in njihovo združevanje na način, da so na vsaki naslednji višji stopnji podatki bolj kompleksni in pri uporabi lahko z njimi podjetja kreirajo višjo dodano vrednost. Prvo stopnjo predstavlja preprosto združevanje lastnih podatkov, ki jih med opravljanjem svojih vsakodnevnih dejavnosti proizvaja državna uprava. Naslednjo stopnjo predstavlja postopno dodajanje in združevanje podatkov drugih agencij javnega sektorja. V tretji stopnji se vključi uradne podatke nevladnih organizacij. Najvišjo stopnjo predstavlja vključitev kompleksnih podatkov, ki jih ustvarjajo in prostovoljno delijo državljani preko družbenih omrežij. Model ocenjuje dva glavna dejavnika:

- Organizacijsko in tehnološko kompleksnost
- Dodano vrednost.

Skozi proces višanja organizacijske in tehnološke kompleksnosti vključevanja podatkov, se viša tudi možnost za povečevanje dodane vrednosti produktov. Na ta način ta model predstavlja pomemben uvid za razvoj modela, ki bi ocenjeval zrelost podjetij za uporabo OP.

3.1.6 Model za oceno odprtih podatkov državne uprave Brazilije (orig. Datos de Gobierno Abierto Brazil – DGABr)

Silva in Pinheiro [24] sta predlagala model, osredotočen na vrednotenje OP državne uprave Brazi-

lije DGABr. Model ocenjuje zrelost objavljenih podatkovnih zbirk javnega sektorja in je sestavljen iz kvalitativnih in kvantitativnih skupin dejavnikov, ki ocenjujejo različne elemente OP. Rezultat predstavlja stopnjo zrelosti, ki pove potencialno uporabnost ocenjevane podatkovne zbirke. Model je prilagojen brazilski zakonodaji, vendar bi ga bilo možno ob ustrezni prilagoditvi uporabiti tudi drugje.

Poleg dejavnikov, ki se osredotočajo na merjenje objavljane OP sta v model vključeni skupini dejavnikov, uporabni tudi pri merjenju zrelosti podjetij za uporabo OP. Ti dve skupini sta:

- Tehnološki dejavniki, ki predstavljajo omejitve za organizacije z vidika razpolaganja z IT infrastrukturo, potrebno za prepoznavo, prevzem ter implementacijo velikih količin podatkov in podatkov v različnih strukturah in formatih.
- Dejavniki upravljanja, ki predstavljajo usmerjenost strategije podjetja v izboljšave poslovanja z uporabo OP in namero vodstva za izvajanje te strategije in zagotavljanje kadrovske virov za implementacijo poslovnega modela, ki temelji na (odprtih) podatkih.

3.1.7 Model zrelosti odprtih podatkov (ang. Open Data Maturity Model – ODMM)

Model avtorjev Dodds in Newman [25] predlaga način ocenjevanja, kako učinkovito organizacije objavljajo in tudi uporabljajo OP. Z opredelitvijo posamezne stopnje za vsako izmed merjenih skupin dejavnikov, pomaga organizacijam identificirati področja in ukrepe za izboljšanje svoje stopnje zrelosti za objavo in uporabo OP.

Model ocenjuje operativne in strateške aktivnosti za objavljane in uporabo OP. Za vsako skupino dejavnikov so definirane aktivnosti za izvedbo, za doseganje posamezne stopnje zrelosti. Dosežena stopnja predstavlja orientacijsko točko glede stanja na različnih področjih in možnih izboljšav. Za vsak dejavnik je opisan cilj, kar omogoča jasen vpogled v potencialne koristi doseganja višjih stopenj zrelosti.

Skupine dejavnikov, ki merijo zrelost za uporabo OP so:

- Zrelost procesov upravljanja podatkov, kot je sposobnost zagotavljanja kakovosti ter upoštevanja standardov za uporabo OP,
- Sposobnosti podjetja za upravljanje znanj in veščin v podjetju, kot so pridobivanje, ohranjanje ali izobraževanje strokovnjakov na področju OP.

Poleg tega pa tudi upravljanje s pridobljenim znanjem znotraj organizacije.

- Investicije in finančni pregled, ki predvideva aktivnosti za vrednotenje lastnih in pridobljenih podatkovnih zbirk in omogoča pregled nad investicijami v OP.
- Strateški pregled, ki pove, ali ima organizacija strategijo upravljanja OP, ali uporablja katalog sredstev OP in ali ima finančna sredstva, ki to omogočajo.

3.1.8 Model digitalne zrelosti

Predlagani model avtoric Kljajić Borštnar in Pucihar [26] obravnava ocenjevanje digitalne zrelosti MSP. Zrelosti za uporabo OP model posebej ne obravnava, vendar sposobnost podjetja za uporabo OP lahko obravnavamo med zmogljivostmi za uporabo naprednih tehnologij. Zaradi tega je ta model vključen v naš pregled. Model vsebuje 32 osnovnih kriterijev, ki se združujejo v hierarhični drevesni strukturi v dve glavni skupini:

- Tehnološke zmogljivosti, ki vključujejo zrelost za integracijo novih digitalnih tehnologij, vloge informatike v strategiji podjetja, digitalnega poslovnega modela in strategije izrabe digitalnih tehnologij, ki so jim na voljo.
- Organizacijske zmogljivosti, ki vključujejo kadrovske vire, kulturo organizacije in upravljanje.

Skupine kriterijev, ki so zastopane v tem modelu in jih lahko smiselno prilagodimo za merjenje zrelosti za uporabo OP v podjetjih so: *digitalne tehnologije, vloga informatike, poslovni model, strategija podjetja, zagotovitev človeških virov, organizacijska kultura in način vodenja podjetja.*

3.2 Pregled obravnavanih modelov

Različni modeli za ocenjevanje zrelosti državne uprave in podjetij za objavlanje in uporabo OP se osredotočajo na različne vidike, kot so tehnološke in organizacijske zmogljivosti, politični in pravni dejavniki, ter odprtost za sodelovanje z državljani in njihovo vključevanje v odločitvene procese. V analiziranih obstoječih modelih so nekatere skupine dejavnikov pojavljajo v več modelih, kar nakazuje, da obstajajo skupne točke, ki so pomembne za merjenje različnih vidikov OP. Dejavnike po njihovi vsebini lahko združimo v sedem enakovrstnih skupin dejavnikov. To so:

- skupina, ki opredeljuje podatke, predvsem z vidika njihove celovitosti in kakovosti za objavo [5], [19], [20], [21], [24], [25]
- skupina, ki opredeljuje strateško upravljanje OP [5], [24], [25], [26]
- skupina, ki opredeljuje pripravljenost za sodelovanje z javnostmi oz. deležniki OP [19], [20], [21], [24], [25]
- skupina, ki opredeljuje tehnološke kapacitete [5], [21], [24], [25], [26]
- skupina, ki opredeljuje vidike zahtev strategije podjetja [5], [21], [23], [24], [25], [26]
- skupina, ki opredeljuje zakonske vidike objavlanja OP [5], [21], [24] in
- skupina, ki opredeljuje potrebe po kadrovskih virih [25], [26].

Celoten pregled izbranih modelov, njihovi cilji, skupine dejavnikov in dejavnike prikazujemo v tabeli 1.

Tabela 1: Pregled modelov OP

Naziv in avtorji	Cilj	Skupine dejavnikov	Dejavniki
Indeks odprtosti e-uprave eGovOI Veljković et al. (2014)	Ocena e-uprave skozi perspektivo odprtih podatkov	Osnovni podatkovni nizi	
		Odprtost podatkov	Popolni
			Osnovni
			Pravočasno
			Dostopni
			Možnost strojne obdelave
			Ne diskriminatorni
			Možnost uporabe z odprtokodnimi programi
			Brezplačna licenca
			Preglednost
	Indeks sodelovanja		
	Indeks vključevanja		
Model zrelosti odprte javne uprave OGMM Lee & Kwak (2012)	Ocena iniciativ za odpiranje javne uprave z namenom omogočanja vključevanja javnosti	Podatki	
		Vključevanje in sodelovanje	
Zrelostni model OP OD-MM Solar et al. (2012)	Ponuja poglobljen vpogled v zrelost in zmožnosti javnih institucij za uresničevanje pobud OP	Organizacijska kultura in zakonodaja	Strategija, način vodenja in organizacijska kultura Zakoni in predpisi Upravljanje
		Tehnologija	Varnost in razpoložljivost Dostopnost Kakovost podatkov
		Omogočanje vključevanja in sodelovanja z državljani in podjetniki	Ponovna uporaba podatkov Razvijalci Vključevanje in sodelovanje
MELODA 5 Abella et al. (2019)	Ocena stopnje ponovne uporabnosti OP	Licence	
		Dostop do informacij	
		Tehnični standardi ponovne uporabe	
		Standardizacija	
		Geolokacijska vsebina	
		Frekvenca posodabljanja podatkov	
		Razširjanje	
		Ugled	
Model uporabe OP javne uprave Kalampokis et al. (2011)	Ponuditi načrt za ponovno uporabo odprtih podatkov javnega sektorja in omogočiti oceno pobud za odpiranje podatkov javnega sektorja	Dodana vrednost	
		Organizacijska in tehnološka kompleksnost	

Naziv in avtorji	Cilj	Skupine dejavnikov	Dejavniki
Metrika za oceno odprtih vladnih podatkov Brazilije DGABr Silva & Pinheiro (2018)	Za ocenjevanje odprtosti javne uprave v Braziliji na podlagi metrik in mednarodnih kazalnikov	Odprti podatki	Popolni
			Osnovni
			Pravočasni
			Dostopni
			Možnost strojne obdelave
			Ne diskriminatorni
			Možnost uporabe z odprtokodnimi programi
			Brezplačna licenca
			Obstoj URI za povezovanje
			Povezani podatki
		Zakonodaja	Vrsta informacij
			Upravljanje občutljivih podatkov
			Načrt upravljanja OP
		Tehnološke zahteve	Vir informacij
			Format podatkov
			Kakovost
			Meta-podatki
			Opredeljeno besedišče
			Geografska dimenzija
	Katalogiranje podatkov		
Upravljalne zahteve	Prioritete in strategije odpiranja		
	Strategija vlaganj in časovnice		
	Vzdrževanje in naslavljanje težav		
Ponovna uporaba	Število objavljenih podatkovnih nizov		
	Število razvitih aplikacij na podlagi OP		
	Stopnja prenosov aplikacij		
	Vključevanje družbe		
	Informacije o ponovni uporabi OP javne uprave		
Model zrelosti odprtih podatkov ODMM Doods & Newman (2015)	Omogoča oceno, kako dobro organizacija objavlja in uporablja odprte podatke, ter identificira aktivnosti za izboljšave	Proces upravljanja podatkov	Objava podatkov
			Razvoj standardov in njihova uporaba
			Upravljanje podatkov
			Upravljanje občutljivih podatkov
		Znanje in veščine	Strokovna znanja o OP
			Upravljanje znanja
		Podpora in vključevanje strank	Procesi vključevanja
			Podpora ponovni uporabi
			Skupnostne norme
		Naložbe in finančni kazalniki	Pregled nad financami
			Vrednotenje podatkovnih setov
			Pridobivanje odprtih podatkov
		Strateški pregled	Strategija odprtih podatkov
	Katalog sredstev		
Digitalna zrelost malih in srednje-velikih podjetij Kljajić Borštnar & Pucihar (2021)	Razvoj modela za oceno digitalne zrelosti malih in srednje-velikih podjetij	Digitalne zmožnosti	Digitalne tehnologije
			Vloga informatike
			Digitalni poslovni model
			Strategija
		Organizacijske zmožnosti	Človeški viri
			Organizacijska kultura
			Upravljanje

3.3 Spodbude in ovire uporabe OP

Spodbude za uporabo OP so v začetku predstavlja le predvsem globalne direktive in zakonodaja, ki so predpisale odpiranje podatkov javnega sektorja. Za namen povečanja razumevanja uporabnosti OP ob hitrem razvoju tehnologij je Evropska komisija objavila poročilo [27] o vrsti poskusov, ki identificirajo možno uporabo prebojnih tehnologij in orodij za inovacije, temelječe na uporabi OP. Med objavo OP in povečevanjem inovacij v podjetjih obstaja močna povezava, zato je za spodbujanje uporabe in inoviranja na podlagi OP pomembna osredotočenost državnih pobud na promocijo dobrega upravljanja OP. To vključuje: zagotavljanje periodičnosti objavljanja OP, zanesljivost virov OP in zagotavljanje objavljanja OP v prihodnosti [9].

V večji meri kot na spodbude se avtorji v obstoječi literaturi osredotočajo na ovire, ki se pojavljajo pri objavi in uporabi OP. Ovire se v podjetjih pojavljajo v treh procesih: 1. pridobivanje in vključitev OP v procese, 2. preoblikovanje OP in njihova uporaba skupaj z obstoječimi podatki na način, da so kasneje lahko uporabljani na nov način, in 3. uporaba OP za inovacije. Za uspešno implementacijo OP v svoje procese, premostitev ovir in uspešno uporabo OP za inoviranje, podjetja potrebujejo specifične zmogljivosti, od sposobnosti za sodelovanje s ponudniki OP, sposobnosti vzpostavitve organizacijske kulture za inoviranje na podlagi OP, veččin upravljanja s kadri, sposobnosti neprestanega razvoja idr. [9].

[15] ugotavljajo, da se ovire lahko pojavljajo pri objavljanju in pri uporabi OP. Ovire, ki se pojavljajo pri ponudnikih OP, vodijo v odpor do objavljanja podatkov. Na strani uporabnikov pa ovire povzročajo težave pri enostavni uporabi OP. V raziskavi so identificirali pet kategorij ovir: *Institucionalne*, ki se nanašajo predvsem na objavljanje OP, *Kompleksnost opravil in Uporaba in sodelovanje*, ki se pojavljajo pri uporabi OP, in *Zakonodaja in Kakovost informacij in Tehnologija*, ki pa se pojavljajo pri uporabi in pri objavljanju OP.

Med življenjskim ciklom OP se pojavljajo različne ovire oz. tveganja [28], [29]. Ta cikel zajema procese od identifikacije, zbiranja in organizacije podatkov za objavo, preko procesov objavljanja podatkov, uporabe podatkov do procesov zbiranja povratnih informacij, ocenjevanja učinka in vzdrževanja podatkovnih setov. Ovire povezane s podatki in njihovo uporabo, upravljanjem, uporabljeno tehnologijo za objavo in uporabo, tgom in uporabniki, organizacijo

in človeškimi viri, se lahko pojavijo v različnih oblikah na več ali vseh stopnjah življenjskega cikla OP.

V raziskavi podatkov, objavljenih v centralnih repozitorijih OP Poljske, Združenega kraljestva, Nemčije in Združenih držav Amerike, je Wiczorkowski (2019) identificiral ovire pri implementaciji rešitev ustvarjenih z uporabo OP. Raziskava se je osredotočila na zrelost podatkovnih formatov, možnosti avtomatiziranega procesiranja z uporabo programskih vmesnikov (ang. Application Programming Interface – API) in uporabo konceptov povezanih OP [8]. Izpostavljene so splošne ovire, ki lahko izhajajo iz tehničnih, ekonomskih, organizacijskih ali človeških dejavnikov. Poleg teh pa še dve dodatni, ki sta specifični pri implementaciji projektov OP, to so: ovire, ki izhajajo iz posebnosti podatkov, kot so periodičnost objavljanja, kakovost in struktura podatkov, ter zakonske ovire, ki omejujejo dostopnost in uporabo teh podatkov.

V preglednem članku sta [10] identificirala niz organizacijskih ovir, s katerimi se podjetja soočajo pri implementaciji OP. Poleg tega sta prepoznala tudi nekatere ovire, ki so v literaturi manj zastopane, kot so politična zavezanost, zaupanje znotraj organizacije, upravljanje IT in druge. Identificirane ovire sta razdelila v tri skupine dejavnikov glede na tehnološki – organizacijski – okoljski okvir (ang. Technological – Organizational – Environmental framework – TOE). V okviru posamezne skupine dejavnikov sta identificirala skupno 22 dejavnikov, ki negativno vplivajo na uporabo OP v podjetjih.

Ugotavljamo, da se ovire, identificirane v pregledani literaturi, ujemajo z ogrodjem TOE, ki ponuja celovito perspektivo za analizo notranje dinamike in dinamike podjetja z okoljem. Poleg tega je TOE dobro sprejet in potrjen kontekstualni okvir, ki pojasnjuje sprejemanje novih tehnologij v podjetjih [10]. Zaradi tega smo celostni okvir, ki ga ponuja ogrodje TOE, uporabili tudi za umestitev ovir, ki se pojavljajo pri objavljanju in uporabi OP.

Iz pregledanih virov smo identificirali skupno 118 ovir, ki smo jih glede na njihov pomen združili v skupine s podobnimi lastnostmi. Na ta način smo identificirali 15 skupin, ki smo jih umestili v ogrodje TOE. V skupino tehnoloških dejavnikov ogrodja TOE smo umestili ovire, ki se nanašajo na standardizacijo; kakovost in uporabnost; infrastrukturo, potrebno za objavljanje in implementacijo OP; upravljanje IT; varnost; dostopnost in obstoj podpore za objavo in upo-

Tabela 2: Skupine ovir OP, vključene v ogrodje TOE

Vrsta dejavnika	Skupina	Število ovir iz virov, vključenih v skupino	Viri
Tehnološke	Standardizacija	8	Çaldağ & Gökalp (2023), Janssen et al. (2012)
	Kakovost in uporabnost	16	Çaldağ & Gökalp (2023), Wiczorkowski (2019), Huber et al. (2020), Janssen et al. (2012), Crusoe & Melin (2018), Wang et al. (2019)
	Infrastruktura	4	Çaldağ & Gökalp (2023), Wiczorkowski (2019), Janssen et al. (2012)
	Upravljanje IT	2	Çaldağ & Gökalp (2023), Wang et al. (2019)
	Varnost	3	Çaldağ & Gökalp (2023), Janssen et al. (2012), Crusoe & Melin (2018), Wang et al. (2019)
	Dostopnost	4	Janssen et al. (2012), Crusoe & Melin (2018)
	Podpora	4	Janssen et al. (2012), Crusoe & Melin (2018), Wang et al. (2019)
Organizacijske	Gospodarske	9	Çaldağ & Gökalp (2023), Wiczorkowski (2019), Janssen et al. (2012), Wang et al. (2019)
	Človeški viri	12	Çaldağ & Gökalp (2023), Wiczorkowski (2019), Huber et al. (2020), Janssen et al. (2012), Crusoe & Melin (2018), Wang et al. (2019)
	Upravljanje	15	Çaldağ & Gökalp (2023), Janssen et al. (2012), Crusoe & Melin (2018), Wang et al. (2019)
	Organizacijska struktura in kultura	8	Çaldağ & Gökalp (2023), Wiczorkowski (2019), Janssen et al. (2012), Crusoe & Melin (2018)
Okoljske	Konkurenca	7	Çaldağ & Gökalp (2023), Huber et al. (2020), Janssen et al. (2012), Crusoe & Melin (2018), Wang et al. (2019)
	Politike	5	Çaldağ & Gökalp (2023), Wang et al. (2019)
	Sodelovanje z deležniki	9	Çaldağ & Gökalp (2023), Huber et al. (2020), Janssen et al. (2012), Crusoe & Melin (2018), Wang et al. (2019)
	Zakonske	12	Wiczorkowski (2019), Huber et al. (2020), Janssen et al. (2012), Wang et al. (2019)

rabo OP. Med organizacijske dejavnike smo umestili dejavnike gospodarske vrednosti OP; človeške vire njihova znanja in veščine za delo z OP; način upravljanja podjetja in organizacijska struktura in kultura podjetja. Okoljske dejavnike predstavljajo konkurenca; politike; sodelovanje z deležniki; zakonski dejavniki in uporabniška podpora. Dejavnike po skupinah ovir glede na ogrodje TOE in njihovo pojavnost v literaturi predstavlja tabela 2.

Potencial OP je bil prepoznan, vendar je za njihovo implementacijo in spodbujanje inovacij ključno zagotavljanje ustreznih tehnoloških in organizacijskih virov v podjetjih. Poleg teh je pomembno tudi upravljanje zunanjih dejavnikov, ki vplivajo na de-

lovanje organizacij in sprejemanje novih tehnologij. Prepoznani dejavniki ovir predstavljajo osnovo za razumevanje (ne)uporabe OP v podjetjih. Z odpravo ovir, ki v podjetjih onemogočajo, ali povečujejo odpor do uporabe OP, lahko povečamo pripravljenost podjetij za uporabo in inoviranje na podlagi odprtih podatkov.

Poglobljeno razumevanje obstoječih modelov in ovir predstavlja osnovo za razvoj modela za merjenje zrelosti podjetij za uporabo OP. Na podlagi ocene zrelosti podjetja za uporabo OP pa lahko podamo individualizirane smernice za izvedbo aktivnosti za povečevanje zrelosti in s tem implementacijo kulture, ki bo OP vključevala v svoje procese.

4 UGOTOVITVE

V raziskavi smo pregledali modele, katerih glavna skupna točka je osredotočenost na merjenje različnih vidikov OP in ovire, ki se pojavljajo pri njihovem objavljanju in uporabi. Identificirali smo osem modelov, ki merijo različne vidike koncepta OP in šest publikacij, ki celostno obravnavajo ovire objave in implementacije OP v podjetjih.

S pregledom obstoječih modelov smo želeli prepoznati dejavnike, ki jih avtorji uporabljajo za ocenjevanje različnih vidikov zrelosti konceptov OP. Ugotavljamo, da se obstoječe raziskave večinoma osredotočajo na vidike zrelosti za objavljanje OP, medtem ko so vidiki zrelosti za uporabo OP le redko obravnavani. Izmed identificiranih modelov le eden obravnava tudi nekatere vidike, pomembne za merjenje zrelosti podjetij za uporabo OP. V identificiranih modelih je potreba po merjenju dejanske uporabe OP večkrat prepoznana, vendar kljub temu modela, ki bi na celovit način obravnaval merjenje zrelosti podjetij za uporabo OP nismo zasledili.

Obstoječi modeli izpostavljajo dejavnike, kot so: kakovost podatkov; možnosti vključevanja državljanov v državotvorne procese; zakonodajni vidiki; organizacijski in tehnološki dejavniki; dejavniki upravljanja organizacij in drugi dejavniki. Ugotavljamo, da te dejavnike lahko združimo v enakovrstne skupine dejavnikov. Te skupine so:

- skupina, ki opredeljuje podatke, predvsem z vidika njihove celovitosti in kakovosti za objavo,
- skupina, ki opredeljuje strateško upravljanje OP v organizaciji,
- skupina, ki opredeljuje pripravljenost za sodelovanje z javnostmi,
- skupina, ki opredeljuje potrebne tehnološke kapacitete,
- skupina, ki opredeljuje vidike zahtev strategije podjetja,
- skupina, ki opredeljuje zakonske vidike objavljanja OP in
- skupina, ki opredeljuje potrebe po kadrovskih virih.

Ob ustrezni prilagoditvi te skupine lahko predstavljajo tudi skupine dejavnikov za merjenje zrelosti podjetij za uporabo OP.

OP so na voljo vsem zainteresiranim javnostim pod licencami, ki omogočajo njihovo uporabo za vsakršne namene, vključno s komercialnimi. Kljub

dostopnosti OP in prizadevanjem državnih uprav za povečanje uporabe OP pa opazamo, da je stopnja dejanske uporabe v podjetjih nizka. V okviru tega pregleda literature smo zato identificirali tudi dejavnike ovir, ki odvrčajo podjetja od integracije OP v poslovne procese ter od izkoriščanja teh podatkov za ustvarjanje večje dodane vrednosti.

Ugotavljamo, da je identificirane ovire za uporabo OP mogoče združiti v skupne dejavnike, ki so tehnološke ali organizacijske narave, oziroma izhajajo iz zahtev okolja podjetja. Identificirane ovire sovpadajo s teoretičnim ogrodjem TOE, ki predstavlja konceptualno dovršeno osnovo, za sistematično razvrstitev teh ovir. V tehnološko dimenzijo smo umestili dejavnike, povezane s standardizacijo, kakovostjo in uporabnostjo podatkov, infrastrukturo, upravljanjem IT, varnostjo, dostopnostjo ter podporo pri uporabi OP. Organizacijska dimenzija zajema ovire, povezane z gospodarskimi vidiki, človeškimi viri, upravljanjem ter organizacijsko strukturo in kulturo. Okoljski dejavniki pa vključujejo ovire, ki izhajajo iz konkurenčnega pritiska, državnih in lokalnih politik, sodelovanja z deležniki OP in zakonodajne omejitve.

Skupine dejavnikov, ki so bili prepoznani v obstoječih modelih za merjenje različnih vidikov OP, se v veliki meri ujemajo z dejavniki, ki predstavljajo ovire za implementacijo OP v poslovne procese podjetij. Ta skladnost nakazuje povezavo med prepoznanimi vidiki objavljanja in uporabe OP in ovirami, ki preprečujejo njihovo učinkovito integracijo. Prepoznavanje in analiza obeh vidikov omogočata identifikacijo kriterijev za razvoj celovitega modela, s katerimi bi lahko merili zrelost podjetij za uporabo OP.

5 ZAKLJUČEK

V raziskavi smo se osredotočili na pregled obstoječih modelov za merjenje različnih vidikov odprtih podatkov ter spodbujevalnih in zaviralnih delavnikov uporabe OP. Pregled relevantne literature je pokazal, da trenutno ne obstaja celovit model, ki bi omogočal merjenje zrelosti podjetij za uporabo OP. Poleg tega smo izpostavili dejavnike, ki ovirajo vključitev odprtih podatkov v poslovne procese podjetij. Cilj raziskave je bil prepoznati dejavnike, ki vplivajo na zrelost podjetij za uporabo OP, s katerimi bo mogoče razviti model za merjenje zrelosti podjetij za uporabo OP.

V ta namen smo izvedli sistematičen pregled literature, osredotočen na obstoječe modele in ovire, ki se pojavljajo pri objavi in uporabi OP. Identificirali

smo 8 publikacij, ki so vsebovale modele merjenja različnih vidikov OP in 6 publikacij, ki so celostno obravnavale ovire objavljajanja in uporabe OP. Pri pregledu literature smo se osredotočili na publikacije, objavljene v digitalnih bibliografskih bazah. Zato moramo, kot omejitve raziskave upoštevati, da niso vsi razviti modeli objavljeni v znanstvenih publikacijah in zato teh modelov pri pregledu literature nismo zajeli.

Ugotovili smo, da so obstoječi modeli večinoma usmerjeni v merjenje zrelosti za objavo kakovostnih podatkov, zbirk podatkov in odprtosti državne uprave. Obstoječi modeli merijo kvalitativne in kvantitativne dejavnike zrelosti za objavo OP, ki jih lahko združimo v skupine dejavnikov in prilagodimo tudi za merjenje zrelosti za uporabo OP. Do podobnih ugotovitev smo prišli tudi pri analizi ovir, ki preprečujejo uporabo OP v podjetjih. Prepoznane ovire lahko združimo v skupine, ki sovpadajo z ogrođjem TOE. Podobnost identificiranih skupin dejavnikov ovir z identificiranimi skupinami dejavnikov v obstoječih modelih nakazuje njihovo povezanost. Na podlagi tega lahko zaključimo, da ob upoštevanju obstoječih modelov za merjenje konceptov OP in prepoznanih ovir lahko identificiramo dejavnike za merjenje zrelosti podjetij za uporabo OP.

Ugotovitve tega pregleda bodo služile kot osnova za oblikovanje kriterijev, ki jih bomo uporabili za razvoj modela za merjenje zrelosti podjetij za uporabo OP. Predlagani model bo podjetjem omogočil, da ocenijo svojo trenutno stopnjo zrelosti za učinkovito vključitev OP v svoje poslovne procese, poleg tega pa podal tudi smernice za izvedbo aktivnosti, potrebnih za povečanje zrelosti za uporabo OP.

S pregledom obstoječe literature smo izvedli uvodni korak v metodologiji načrtovanja in razvoja. Prepoznani dejavniki omogočajo razvoj modela za celovito merjenje zrelosti podjetij za uporabo odprtih podatkov. V prihodnjih raziskavah predlagamo izvedbo nadaljnjih korakov v okviru zastavljene metodologije, in sicer strukturiranje prepoznanih dejavnikov v ustrezno hierarhijo in razvoj ter vrednotenje modela za ocenjevanje zrelosti podjetja za uporabo odprtih podatkov. Na podlagi ocene tako izpopolnjenega modela pa bo možno podati konkretne predloge za aktivnosti, s katerimi bodo podjetja lahko povečala svoj potencial za inoviranje na podlagi OP in si s tem odprla nove konkurenčne priložnosti.

ZAHVALA

Raziskavo financirata Javna agencija za znanstvenoraziskovalno in inovacijsko dejavnost Republike Slovenije ter Ministrstvo za digitalno preobrazbo Republike Slovenije v okviru ciljnega raziskovalnega projekta V5-2356 in raziskovalnega programa P5-0018.

VIRI IN LITERATURE

- [1] European Commission, »Directive 2003/98/EC of the European Parliament and of the Council of 17 November 2003 on the re-use of public sector information«. Pridobljeno: 4. marec 2024. [Na spletu]. Dostopno na: <https://eur-lex.europa.eu/eli/dir/2003/98/oj>
- [2] Republika Slovenija, »Zakon o dostopu do informacij javnega značaja«, Uradni list RS, Slovenija, 3. maj 2026. Pridobljeno: 7. marec 2024. [Na spletu]. Dostopno na: <https://pisrs.si/pregledPredpisa?id=ZAKO4868>
- [3] B. Obama, »Transparency and Open Government«. The White House, 21. januar 2009. Pridobljeno: 8. marec 2024. [Na spletu]. Dostopno na: <https://obamawhitehouse.archives.gov/the-press-office/transparency-and-open-government>
- [4] OPSI, »OPSI«, Kaj so odprti podatki? [Na spletu]. Dostopno na: <https://podatki.gov.si/posredovanje-podatkov/kaj-so-odprti-podatki>
- [5] A. Abella, M. Ortiz-de-Urbina-Criado, in C. De-Pablos-Hereadero, »Meloda 5: A metric to assess open data reusability«, *EI Prof. Inf.*, let. 28, št. 6, jan. 2020, doi: 10.3145/epi.2019.nov.20.
- [6] J. Wiczorkowski, »Open Data as a Source of Product and Organizational Innovations«, v *Proceedings of the 14th European Conference on Innovation and Entrepreneurship ECIE 2019*, Kalamata, Greece: Academic Conferences and Publishing International Limited, sep. 2019, str. 1118–1127. doi: 10.34190/ECIE.19.190.
- [7] Open Government Working Group, »Open Government Data Principles«, Open Government Data Principles. Pridobljeno: 5. marec 2024. [Na spletu]. Dostopno na: https://public.resource.org/8_principles.html
- [8] T. Berners-Lee, »5 star open data«, 5 star open data. Pridobljeno: 22. februar 2024. [Na spletu]. Dostopno na: <https://5stardata.info/en/>
- [9] F. Huber, T. Wainwright, in F. Rentocchini, »Open data for open innovation: managing absorptive capacity in SMEs«, *RD Manag.*, let. 50, št. 1, str. 31–46, jan. 2020, doi: 10.1111/radm.12347.
- [10] M. T. Çaldağ in E. Gökalp, »Understanding barriers affecting the adoption and usage of open access data in the context of organizations«, *Data Inf. Manag.*, str. 100049, sep. 2023, doi: 10.1016/j.dim.2023.100049.
- [11] OECD, *The Digital Transformation of SMEs*. v OECD Studies on SMEs and Entrepreneurship. OECD, 2021. doi: 10.1787/dbd9256a-en.
- [12] Publications Office of the EU, *The economic impact of open data: opportunities for value creation in Europe*. LU: Publications Office, 2020. Pridobljeno: 28. avgust 2024. [Na spletu]. Dostopno na: <https://data.europa.eu/doi/10.2830/63132>
- [13] Publications Office of the EU, *Rethinking the impact of open data: a first step towards a European impact assessment for open data.*, let. 2023. LU: Publications Office, 2023. Pridobljeno: 22. avgust 2024. [Na spletu]. Dostopno na: <https://data.europa.eu/doi/10.2830/911822>

- [14] A. Zuiderwijk, M. Janssen, in Y. K. Dwivedi, »Acceptance and use predictors of open data technologies: Drawing upon the unified theory of acceptance and use of technology«, *Gov. Inf. Q.*, let. 32, št. 4, str. 429–440, okt. 2015, doi: 10.1016/j.giq.2015.09.005.
- [15] M. Janssen, Y. Charalabidis, in A. Zuiderwijk, »Benefits, Adoption Barriers and Myths of Open Data and Open Government«, *Inf. Syst. Manag.*, let. 2012, št. 29, str. 258–268, 2012, doi: 10.1080/10580530.2012.716740.
- [16] OECD, 2023 *OECD Open, Useful and Re-usable data (OUR-data) Index: Results and key findings*, let. 2023. v OECD Public Governance Policy Papers, no. 43, vol. 2023. Paris, 2023. [Na spletu]. Dostopno na: <https://doi.org/10.1787/a37f51c3-en>
- [17] Publications Office of the EU, *Open data maturity report 2023*. LU: Publications Office, 2023. Pridobljeno: 28. avgust 2024. [Na spletu]. Dostopno na: <https://data.europa.eu/doi/10.2830/384422>
- [18] SURS, »Izvajanje podatkovne analitike, velikostni razred podjetja«, Izvajanje podatkovne politike. Pridobljeno: 23. avgust 2024. [Na spletu]. Dostopno na: <https://pxweb.stat.si/SiStat-Data/pxweb/sl/Data/-/2984101S.px/table/tableViewLayout2/>
- [19] N. Veljković, S. Bogdanović-Dinić, in L. Stoimenov, »Benchmarking open government: An open data perspective«, *Gov. Inf. Q.*, let. 31, št. 2, str. 278–290, apr. 2014, doi: 10.1016/j.giq.2013.10.011.
- [20] G. Lee in Y. H. Kwak, »An Open Government Maturity Model for social media-based public engagement«, *Gov. Inf. Q.*, let. 29, št. 4, str. 492–503, okt. 2012, doi: 10.1016/j.giq.2012.06.001.
- [21] M. Solar, G. Concha, in L. Meijueiro, »A Model to Assess Open Government Data in Public Agencies«, v *Electronic Government*, let. 7443, H. J. Scholl, M. Janssen, M. A. Wimmer, C. E. Moe, in L. S. Flak, Ur., v Lecture Notes in Computer Science, vol. 7443. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, str. 210–221. doi: 10.1007/978-3-642-33489-4_18.
- [22] H. J. Wang in J. Lo, »Factors Influencing the Adoption of Open Government Data at the Firm Level«, *IEEE Trans. Eng. Manag.*, let. 67, št. 3, str. 670–682, avg. 2020, doi: 10.1109/TEM.2019.2898107.
- [23] E. Kalampokis, E. Tambouris, in K. Tarabanis, »Open Government Data: A Stage Model«, v *Electronic Government*, M. Janssen, H. J. Scholl, M. A. Wimmer, in Y. Tan, Ur., v Lecture Notes in Computer Science, vol. 6846. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, str. 235–246. doi: 10.1007/978-3-642-22878-0_20.
- [24] P. N. Silva in M. M. K. Pinheiro, »DGABr: Metric for evaluating Brazilian open government data«, *Informação Soc. Estud.*, let. 28, št. 3, dec. 2018, doi: 10.22478/ufpb.1809-4783.2018v28n3.42183.
- [25] L. Dodds in A. Newman, »Open Data Maturity Model«, Open Data Institute, Technical Report, mar. 2015. [Na spletu]. Dostopno na: <https://theodi.org/insights/reports/open-data-maturity-model/>
- [26] M. Kljajić Borštnar in A. Pucihar, »Multi-Attribute Assessment of Digital Maturity of SMEs«, *Electronics*, let. 10, št. 8, str. 885, apr. 2021, doi: 10.3390/electronics10080885.
- [27] C. Granell idr., *Emerging approaches for data-driven innovation in Europe: sandbox experiments on the governance of data and technology*. Luxembourg: Publications Office of the European Union, 2022.
- [28] J. Crusoe in U. Melin, »Investigating Open Government Data Barriers: A Literature Review and Conceptualization«, v *Electronic Government*, let. 11020, P. Parycek, O. Glassey, M. Janssen, H. J. Scholl, E. Tambouris, E. Kalampokis, in S. Virkar, Ur., v Lecture Notes in Computer Science, vol. 11020. , Cham: Springer International Publishing, 2018, str. 169–183. doi: 10.1007/978-3-319-98690-6_15.
- [29] Wang, A. Zhao, H. Zhao, in J. Chu, »Building a Holistic Taxonomy Model for OGD-Related Risks: Based on a Lifecycle Analysis«, *Data Intell.*, let. 1, št. 4, str. 309–332, nov. 2019, doi: 10.1162/dint_a_00018.
- [30] J. Wiczorkowski, »Barriers to Using Open Government Data«, v *Proceedings of the 2019 3rd International Conference on E-commerce, E-Business and E-Government – ICE-EG 2019*, Lyon, France: ACM Press, 2019, str. 15–20. doi: 10.1145/3340017.3340022.

■

Staša Blatnik je po zaključenem magistrskem študiju informacijskih sistemov na Fakulteti za organizacijske vede Univerze v Mariboru začela kariero na področju implementacije in upravljanja informacijskih sistemov in drugih področjih. Po desetletju pridobivanja poklicnih izkušenj v zasebnem in javnem sektorju je nadaljevala izobraževanje na doktorski ravni. Kot doktorska kandidatka se v svojih raziskavah posveča konceptu odprtih podatkov, zlasti njihovi uporabi v podjetjih.

■

Andreja Pucihar je redna profesorica na področju informacijskih sistemov na Fakulteti za organizacijske vede Univerze v Mariboru in vodja diplomskega in magistrskega študijskega programa organizacija in management informacijskih sistemov. Je tudi vodja laboratorija za digitalno poslovanje - eCenter. Raziskovanje usmerja v inovacije na področju informacijskih sistemov, predvsem v digitalno preobrazbo in digitalne poslovne modele. Raziskovalne dosežke redno objavlja v domačih in mednarodnih revijah in na konferencah. Ima preko 20 let izkušenj z industrijskimi in mednarodnimi evropskimi projekti, ki so bili in so prevladujoče usmerjeni v digitalizacijo, digitalno preobrazbo in podporo malim in srednje velikim podjetjem. Od leta 2009 je vodja uveljavljene mednarodne konference o digitalnem poslovanju – Bled eConference, ki na tem področju deluje že vse od leta 1988. Je tudi so-urednica revij s faktorjem vpliva »Electronic Markets - The International Journal on Networked Business« in »Journal of Theoretical and Applied Electronic Commerce Research«.

■

Mirjana Kljajić Borštnar je redna profesorica za področje informacijskih sistemov na Fakulteti za organizacijske vede Univerze v Mariboru. Njeno raziskovalno delo je usmerjeno v sisteme za podporo odločanju, odkrivanju znanja v podatkih in organizacijsko učenje. Je glavna urednica revije *Uporabna informatika*, podpredsednica Slovenskega društva INFORMATIKA, sovodja programskih odborov mednarodnega simpozija operacijskih raziskav in Blejske e-konference ter članica izvršnega odbora AI4Slovenia.

Poenostavite upravljanje vašega IT-okolja z rešitvijo NIL Cloud Management Platform

Preoblikujte vaš podatkovni center v sodobno storitveno platformo. Zagotovite si preglednost stroškov in učinkovito dostavo storitev IT.

Prednosti NIL Cloud Management Platform



Ena platforma za celovito upravljanje okolja skozi storitveno tržnico



Izboljšanje odzivnosti in učinkovitosti IT-službe skozi avtomatizacijo in orkestracijo



Procesna in stroškovna preglednost vedno bolj kompleksnih IT-okolij z možnostjo integracije z zunanjimi sistemi (SIEM, XDR, EDR, ITSM...)

Kontaktirajte nas za demo:

consulting@conscia.com

www.nil.com



SOPHOS

Cybersecurity delivered.



Sophos Managed Detections and Response

Sophos MDR je najbolj razširjena MDR storitev na svetu. Zaupa nam že več kot **18.000** podjetij!



Distributer: Sophos d.o.o., www.sophos.si, slovenija@sophos.si, T: 07/39 35 600

Iz Islovarja

Islovar je spletni terminološki slovar informatike in računalništva, ki ga objavlja jezikovna sekcija Slovenskega društva INFORMATIKA. Navajamo nekaj izpisov na temo varovanja:

informacijsko bojevanje -ega -a s (*angl. information warfare*) onemogočanje dostopa do informacij z namenom doseči prednost ali premagati nasprotnika; prim. kibernetško bojevanje

izkoriščevalec ranljivosti -lca -- m (*angl. exploit*) škodljiva koda, ki izkorišča ranljivost računalniškega sistema

kibernetsko bojevanje -ega -a s (*angl. cyber warfare*) dejanja držav ali organizacij, ki poskušajo s kibernetškimi napadi poškodovati računalnike ali omrežja nasprotnika; sin. kibernetško vojskovanje; prim. informacijsko bojevanje

komplét za izkoriščanje -a -- -- m (*angl. exploit toolkit, exploit kit*) programski paket z različnimi orodji za izkoriščanje ranljivosti računalniških sistemov, npr. brskalnikov, operacijskih sistemov

kripto ugrabítev -- -tve ž (*angl. cryptojacking*) kibernetški kriminal, pri katerem kriminallec prikrito izrablja procesorsko moč žrtve za rudarjenje kriptovalute; sin. rudarjenje v mimohodu

módra skupína -e -e ž (*angl. blue team*) varnostni strokovnjaki, ki branijo informacijski sistem pred napadi; prim. rdeča skupina

rdéča skupína -e -e ž (*angl. red team*) varnostni strokovnjaki, ki s testnimi napadi preizkušajo odpornost informacijskega sistema; prim. modra skupina

volumétrični napád -ega -a m (*angl. volumetric attack*) vrsta napada porazdeljene ohromitve storitve, kjer se žrtev poplavi z veliko količino podatkov

Znanstveni prispevki

Yauhen Unuchak, Mirjana Kljajić Borštnar, Tatyana Unuchak
UPORABA DNEVNIŠKIH ZAPISOV V NAMEN PROFILIRANJA
OBREMENITEV SPLETNE APLIKACIJE

Strokovni prispevki

Gal Gantar, Matevž Pesek
ANALIZA NAPADOV NA PAMETNE POGODBE OSNOVANE
NA OKOLJIH ETHEREUM VIRTUALNEGA STROJA

Maksim Nikitashin
UMETNA INTELIGENCA IN RAZVOJ UPORABNIŠKIH VMESNIKOV:
ŠTUDIJA PRIMERA MOBILNE APLIKACIJE X

Pregledni znanstveni prispevki

Staša Blatnik, Andreja Pucihar, Mirjana Kljajić Borštnar
SPODBUDE IN OVIRE UPORABE ODPRTIH PODATKOV
V PODJETJIH – PREGLED LITERATURE

Informacije

IZ ISLOVARJA

