U P O R A B N A
I N F O R M A T I K A
2025 ŠTEVILKA 1 JAN/FEB/MAR LETNIK XXXIII ISSN 1318-1882

Znanstveni prispevki

Strokovni prispevki

Pregledni znanstveni prispevki

Informacije

2025 ŠTEVILKA 1 JAN/FEB/MAR LETNIK XXXIII ISSN 1318-1882

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

Predstavnik
Slavko Žitnik

Odgovorni urednik
Mirjana Kljajić Borštnar

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.

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@drustvo-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 ustreznic 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 metod strojnega učenja za klasifikacijo nalog po prioritetah v IT projektih

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

Izvleček

Določanje prioritet in razvrščanje nalog še vedno predstavlja izziv pri učinkovitem vodenju projektov. Obstaja veliko klasičnih pristopov za določanje prioritet. Vendar so te tehnike delovno intenzivne, subjektivne in neprilagodljive. V prispevku obravnavamo pristope za samodejno določanje prioritet nalog v IT projektih, ki temeljijo na strojnem učenju. Raziskujemo, kako lahko z uporabo metod strojnega učenja pomagamo projektnim vodjem pri učinkovitejšem razvrščanju nalog v IT projektih. V ta namen smo na množici več kot 1000000 zapisov projektnih nalog razvili klasifikacijski model za samodejno določanje prioritet. Problem, ki smo ga obravnavali, je večrazredni, pri tem je večina primerov, označenih z najvišjo prioriteto, kar predstavlja izziv pri modeliranju kot tudi pri učinkovitosti upravljanja IT projektov. Preskusili smo različne algoritme ter različne pristope, s ciljem izboljšanja rezultatov klasifikacije. Pokazali smo, da je naloge smiselno razvrstiti v manjše skupine prioritet, kar prispeva k večji natančnosti klasifikacijskega modela in preglednosti prioritet nalog, slednje pa lahko olajša upravljanje IT projektov.

Ključne besede: vodenje IT projektov, strojno učenje, določanje prioritet nalog, večrazredna klasifikacija, neuravnoteženost podatkov.

Using machine learning methods to classify tasks by priority in IT projects

Abstract

Prioritizing and classifying tasks remain a challenge in effective project management. There are many classic approaches to prioritization. However, all these techniques are labour intensive, subjective and lack flexibility. In this paper we investigate machine learning based approaches for automatic prioritization of tasks in IT projects. We explore how machine learning methods can be used to help project managers prioritize tasks in IT projects more efficiently. We developed a classification model for automatically determining priorities based on a dataset of over 1,000,000 project task records. The problem we addressed is multi-class, with the majority of cases labelled with the highest priority, which presents a challenge both in modelling and in the efficiency of IT project management. To address these challenges, we explored strategies to enhance classification performance, including reducing the number of priority classes. Our findings demonstrate that simplifying the priority structure improves the model's accuracy and contributes to more efficient task management in IT projects.

Keywords: IT project management, machine learning, task prioritization, multiclass classification, data imbalance.

1 UVOD

Napredki na področju umetne inteligence in strojnega učenja hitro spreminjajo številne panoge po vsem svetu [1]. Njihov največji vpliv na procese upravljanja je opazen v IT sektorju [2]. V zadnjih letih se je močno povečalo zanimanje za uporabo umetne inteligence pri optimizaciji različnih vidikov vodenja projektov. Nedavna raziskava [2] je pokazala, da se različne metode umetne inteligence, kot sta razvrščanje in napovedovanje, dosledno vključujejo v splo-

šne procese vodenja projektov. Algoritmi strojnega učenja lahko samodejno kategorizirajo elektronska sporočila, posodabljajo statuse projektov in ustvarjajo poročila – naloge, ki bi sicer zahtevale dragoceni čas in vire [3], [4]. Ta avtomatizacija omogoča vodjem projektov, da se osredotočijo na strateške vidike svojega dela.

Posebej zanimiv vidik projektnega vodenja je povezan z določanjem prioritet in razvrščanjem nalog. Sodobni projekti namreč ustvarjajo ogromno količino podatkov, vključno z besedilnimi opisi nalog, časovnimi oznakami, metrikami uspešnosti itd. Ročna obdelava in analiza teh podatkov postaja vse bolj zamudna in neučinkovita. Z uporabo strojnega učenja pri razvrščanju nalog lahko bistveno izboljšamo vodenje projektov, saj to pomaga prepoznati kritične naloge, napovedati tveganja in optimizirati dodeljevanje virov. To je še posebej pomembno pri velikih in zapletenih projektih, kjer lahko napaka pri določanju prioritet nalog povzroči velike zamude in prekoračitve virov (denarnih, časovnih, človeških, tehničnih). Človeške napake in subjektivnost v procesu odločanja lahko negativno vplivajo na vodenje projektov. Uporaba strojnega učenja pomaga zmanjšati vpliv človeških napak ter zagotavlja bolj objektiven in temelječ na podatkih pristop k določanju prioritet nalog. Algoritmi strojnega učenja lahko samodejno obdelajo in analizirajo velike količine podatkov, kar znatno pospeši postopek odločanja. Algoritmi lahko upoštevajo različne dejavnike, kot so besedilni opis naloge v IT projektu, njena kompleksnost, časovni okvir, poslovna pomembnost itd. Avtomatizacija tega procesa omogoča tudi hitro prilagajanje spremembam, saj je mogoče modele sproti prilagajati posodobljenim podatkom.

Določanje prioritet nalog vključuje ocenjevanje in razvrščanje nalog glede na njihovo pomembnost, nujnost in vpliv. Pri vodenju projektov določanje prioritet nalog zagotavlja, da so prizadevanja usmerjena v naloge, ki so najpomembnejše za doseganje ciljev projekta, kar preprečuje, da bi se čas in viri zapravljali za manj pomembne naloge. Učinkovit organizator dela se zaveda, kako pomembno je določanje prioritet nalog za uspešno upravljanje časa.

Številni dokumenti, ki urejajo projektne dejavnosti (npr. PMBOK, BABOK, PRINCE2), vsebujejo opise nekaterih tehnik določanja prioritet [5], [6], [7]. Vendar so vse te tehnike pri izvajanju delovno intenzivne (zahtevajo posebne napore razvojne ekipe), subjektivne (ker prioritete določajo ljudje) in same po sebi niso prilagodljive (v primeru sprememb zahtev naročnika, pojave novih tehnologij itd.).

Cilj raziskave je razviti modele strojnega učenja za klasifikacijo nalog v IT projektu po prioritetah, kar bo omogočilo boljšo organizacijo delovnih nalog v procesu razvoja programske opreme. Klasifikacijski model bo pomagal samodejno določiti prioritete nalog, natančneje opredeliti najpomembnejše naloge in izboljšati dodeljevanje virov v IT projektih. Osnovno raziskovalno vprašanje je: »Ali lahko z uporabo metod strojnega učenja pomagamo projektnim vodjem pri bolj učinkovitem razvrščanju prioritet nalog v IT projektih?«.

V nadaljevanju članka najprej predstavimo pregled sorodnih del, povezanih z razvrščanjem prioritet, opišemo metodologijo raziskave, faze predhodne obdelave podatkov, uporabljene algoritme razvrščanja, metrike vrednotenja modelov razvrščanja in opisujemo postopek modeliranja. Nadaljujemo z rezultati modeliranja, vključno z analizo kakovosti dobljene razvrstitve. Na koncu zaključimo z oceno pridobljenih rezultatov, omejitvami in implikacijami za prakso.

2 SORODNA DELA

Mednarodni inštitut za poslovno analizo (angl. International Institute of Business Analysis) je koncept določanja prioritet oblikoval kot »določitev relativne pomembnosti niza elementov za določitev vrstnega reda, v katerem jih bomo obravnavali«[6]. Posebno pozornost je treba nameniti temu, kaj je predmet prioritizacije (ali čemu dajemo prioriteto). Številni avtorji obravnavajo problem »določanja prioritet zahtev« (angl. »requirements prioritization«) [8], [9], [10]. Vendar se današnji razvijalci ne soočajo le s prioritetnim razvrščanjem zahtev, temveč tudi s prioritetnim razvrščanjem epov (angl. Epic), uporabniških zgodb (angl. User Story), nalog (angl. Task), hroščev (angl. Bug) [11]. Raznolikost vrst projektnih nalog omogoča prilagodljivo vodenje razvojnega procesa, vendar hkrati prinaša izzive pri določanju prioritet in razvrščanju nalog. Ob upoštevanju teh vidikov lahko rečemo, da splošne pristope za določanje prioritet nalog lahko uporabimo tudi v IT projektih. Obstoječe pristope za določanje prioritet razvrstimo na dve skupini: klasični pristopi in pristopi, ki temeljijo na strojnem učenju.

2.1 Klasični pristopi

Obstaja več pristopov k določanju prioritet nalog, ki vključujejo naslednje metode: MoScoW (angl. Must have, Should have, Could have, Won't have this time), preprosto rangiranje, razvrščanje z mehurčki, drevo binarnega iskanja (angl. Binary Search Tree), analitični hierarhični proces (angl. Analytic Hierarchy Process, AHP), pristop stroškov in vrednosti (angl. Cost--Value Approach, CVA), metoda $100 (angl. Hundred Dollar Method), kumulativno glasovanje, igre načrtovanja (angl. Planning Game, PG), numerično razvrščanje (angl. Numerical Assignment, NA), teorija-W (Win-Win Theory), Wiegerjev pristop [8], [9], [10], [12]. Glede na izbrano metodo je treba uporabiti ordinalno, nominalno ali intervalno lestvico [12].

Med njimi najpogosteje uporabljene metode so AHP, igre načrtovanja, CVA, razvrščanje z mehurčki, MoSCoW [8]. Te metode omogočajo vodjem projektov in razvijalcem sprejemanje utemeljenih odločitev o tem, katere naloge ali zahteve imajo prednost pri izvedbi. Vendar pa imajo te metode resne omejitve, zlasti ko gre za stopnjevanost pri velikem številu zahtev. V pogojih sodobnih projektov programske opreme, kjer se število nalog in sprememb lahko meri v stotinah in celo tisočih, postanejo ročne metode določanja prioritet izjemno zamudne in neučinkovite [8], [12]. Postopek določanja prioritet mora biti enostaven za osvojitev, preprost za uporabo, neviden za zainteresirane strani, da si pridobi njihovo zanimanje in zaupanje, razumljiv za nestrokovnjake, zanesljiv in učinkovit [12].

Metode AHP, BST, numerično razvrščanje in MoSCoW, še vedno priljubljene za določanje prioritet nalog, vendar jih je treba prilagoditi bolj zapletenim in dinamičnim projektnim okoljem v realnem svetu [10]. Raziskovalci se zato obračajo k bolj prilagodljivim in razširljivim metodam, ki upoštevajo parametre, kot so odvisnost nalog, kritičnost in časovni razpored. Pri tem AHP je ena od najbolj natančnih metod, ki se uporabljajo v IT industriji, vendar obstajajo tudi njene omejitve pri razširljivosti in potrebe po iskanju novih pristopov [9].

Sodobni trendi določanja prioritet nalog se vse bolj obračajo k metodam strojnega učenja, ki lahko avtomatizirajo postopek analize nalog, ugotavljanja povezav in optimizacije vrstnega reda njihovega izvajanja [9].

2.2 Pristopi strojnega učenja

Sodobni pristopi k določanju prioritet nalog uporabljajo možnosti strojnega učenja za natančnejše in učinkovitejše upravljanje nalog, zlasti pri kompleksnih projektih, ki zahtevajo veliko podatkov. Metode strojnega učenja avtomatizirajo postopek določanja prioritet na podlagi podatkov o nalogi, kot so opis naloge, trajanje izvedbe, kritičnost in drugi dejavniki. Algoritmi, kot so odločitvena drevesa, naključni gozdovi, gradientno povečanje z drevesi odločanja in nevronske mreže, se lahko učijo iz preteklih podatkov in to znanje uporabijo za določanje prioritet nalog v prihodnosti. Leta 2004 sta Cubranic in Murphy [13] izvedla eno od prvih raziskav, katerih cilj je bil določiti prioritete napak z uporabo metod strojnega učenja. Pokazali so, da lahko uporaba kategorizacije besedila z nadzorovanim Bayesovim učenjem izboljša natančnost dodeljevanja nalog razvijalcem. V študiji so uporabili velike nabore podatkov iz odprtokodnih projektov, ki so postali dejanski standardi za testiranje modelov strojnega učenja.

Možnosti uporabe strojnega učenja za izboljšanje natančnosti in hitrosti določanja prednostnih nalog v IT projektih so bile potrjene v številnih študijah. Kombinacija naivnega Bayesovega klasifikatorja z optimizacijo pričakovanj izboljša natančnost klasifikacije na 48 % [14].

Klasifikacija kot raziskovalno področje strojnega učenja danes zajema različne vidike in vrste nalog. Opredeljene so štiri glavne kategorije klasifikacijskih nalog v strojnem učenju: binarna, večrazredna, klasifikacija z več oznakami in neuravnotežena klasifikacija [15]. Najbolj priljubljeni algoritmi, ki se uporabljajo za reševanje takih nalog, so logistična regresija, k-najbližjih sosedov, odločitvena drevesa, metoda podpornih vektorjev, naivni Bayesov klasifikator, naključni gozdovi in gradientno povečanje z drevesi odločanja. Primeri različnih klasifikacij vključujejo odkrivanje neželene elektronske pošte, odkrivanje napak v proizvodnih procesih in napovedovanje pretvorbe, prepoznavanje obrazov, razvrščanje rastlinskih vrst in kategoriziranje novic itd. [15].

Pri delu z velikimi količinami podatkov pri razvoju programske opreme se pogosto pojavi problem neuravnoteženosti podatkov, ki otežuje postopek razvrščanja [15], [16], [17]. Uspešnost klasifikacije je zelo odvisna od stopnje neuravnoteženosti razredov [17]. Obstoječe rešitve za odpravo neuravnoteženosti razvite tako na ravni podatkov kot na ravni algorit-

mov. Na ravni podatkov je rešitev v uravnoteženju porazdelitve razredov s ponovnim vzorčenjem podatkovnega prostora, medtem ko se na ravni algoritmov rešitev osredotoča na prilagoditev obstoječih algoritmov za usposabljanje klasifikatorjev, s čimer se okrepi učenje manjšinskih razredov [16].

Tako pregled literature kaže, da se poleg tradicionalnih metod določanja prednosti nalog pogosto uporabljajo tudi sodobnejši pristopi, ki temeljijo na strojnem učenju in analizi podatkov. To je utemeljeno s potrebo po izboljšanju učinkovitosti in natančnosti projektnega vodenja v kontekstu nenehno naraščajoče količine podatkov in spremenljivosti zahtev.

3 METODOLOGIJA

V raziskavi smo sledili procesu CRISP-DM (angl. Cross-Industry Standard Process for Data Mining) [18]. Metodologija zagotavlja strukturiran pristop k izvajanju projektov podatkovne analize in strojnega učenja ter je sestavljena iz naslednjih šestih korakov: razumevanje (poslovnega) problema (angl. Business Understanding), razumevanje podatkov (angl. Data Understanding), priprava podatkov (angl. Data Preparation), modeliranje (angl. Modeling), vrednotenje modela (angl. Evaluation), izvajanje in spremljanje (angl. Deployment and Monitoring). Za podatkovno rudarjenje in obdelavo podatkov smo uporabili programski jezik Python in knjižnice Ijson, pandas, seaborn, imblearn, matplotlib, sklearn.

3.1 Razumevanje (poslovnega) problema

V tej raziskavi je glavni poslovni problem potreba po avtomatizaciji postopka določanja prioritet nalog v IT projektih na podlagi njihovega besedilnega opisa. V sodobnih sistemih za upravljanje vodenje projektov, kot je Jira, je veliko nalog z različnimi prednostnimi nalogami. Vendar pa zaradi velike količine podatkov in zapletenosti ročnega razvrščanja obstaja tveganje, da naloge z visoko prioriteto ne bodo pravočasno obdelane, kar bo negativno vplivalo na učinkovitost ekipe. Ker je ocena prioritete naloge subjektivna, se pogosto zgodi, da je večina nalog razvrščena v najvišjo prioriteto, kar ne prispeva k izboljšanju učinkovitosti izvajanja projektov.

Glavna ideja je raziskati smeri za izboljšanje rezultatov razvrščanja nalog v IT projektih na podlagi algoritmov strojnega učenja. Dobljeni rezultati naj bi postali osnova za izboljšanje procesa upravljanja nalog.

3.2 Razumevanje podatkov

Podatke smo pridobili iz javno dostopnega vira Zenodo https://zenodo.org/records/5901804) [19]. Iz arhiva mongodump-JiraRepos.archive smo povzeli podatke in jih pretvorili v obliko JSON-datoteke, ki je vsebovala naslednje atribute (značilke) nalog v angleščini: prioriteta (angl. Priority), opis stanja (angl. Status Description), naziv stanja (angl. Status Name), kategorija stanja (angl. Status Category), ime ustvarjalca (angl. Creator Name), časovni pas ustvarjalca (angl. Creator Time Zone), ime poročevalca (angl. Reporter Name), skupni napredek (angl. Aggregate Progress), skupni seštevek (Aggregate Total), skupni odstotek (angl. Aggregate Percent), opis vrste zadeve (angl. Issue Type Description), ime vrste zadeve (angl. Issue Type Name), porabljeni čas (angl. Time Spent), ključ projekta (angl. Project Key), vrsta projekta (angl. Project Type), datum ustvarjanja (angl. Created Date), datum posodobitve (angl. Updated Date), opis naloge (angl. Task Description), povzetek (angl. Summary). Iz datoteke JSON smo izbrali polja »Priority«, »Status Name«, »Creator Name«, »Time Spent«, »Task Description« in pretvorili v obliko CSV. Izbrana polja omogočajo ključne informacije za razumevanje prioritet nalog, stanja njihove izvedbe, vpletenih oseb, porabljenega časa ter vsebine nalog. Za modeliranje smo izbrali značilke: prioriteta in opis naloge. Omejitev na dve ključni značilke zmanjša kompleksnost modela, kar je koristno pri začetnih analizah in primerno za končne uporabnike, saj omogoča jasne vpoglede v nujnost nalog in njihov kontekst. Obe značilki predstavljata besedilni podatkovni tip. Prioriteta je imela naslednje vrednosti: »Major«, »Minor«, »Critical«, »Blocker«, »Trivial«, »Unknown«, »Normal«, »Low«. Izvirna podatkovna datoteka je vsebovala 1014926 zapisov (primerov) o nalogah iz 645. IT projektov. Uporabili smo različne IT projekte, da bi lahko klasifikator učili na različnih naborih podatkov. Po odstranitvi zapisov z manjkajočimi vrednostmi nam je ostalo 916030 zapisov, enega od katerih prikazuje slika 1.

Prioritete za vsako nalogo so določili avtorji, ki so te naloge ustvarili v Jiri. Za nadaljnjo obdelavo besedila je bila potrebna predhodna priprava podatkov, vključno s tokenizacijo, odstranjevanjem stop besed in lematizacijo.

Klasifikacijo smo izvedli na projektnih nalogah, ki so bile razvrščene v osmih razredih prioritet:

```
priority,status,creator_name,timespent,description
Major,Open,RyanO751,,"In a production environment with some connectivity problems it was found the ZooKeeper
server was using over 1000 threads with name ""SyncThread"" (that were never being freed).
Looking through the server logs indicates that these nodes were experiencing connection timeouts to the leader.
A test environment (described below in the ""environment"" field of this ticket) showed that these connection
timeouts are what seem to be leaking these threads."
```

Slika 1: **Primer opisa naloge iz IT projekta s prioriteto »Major«**

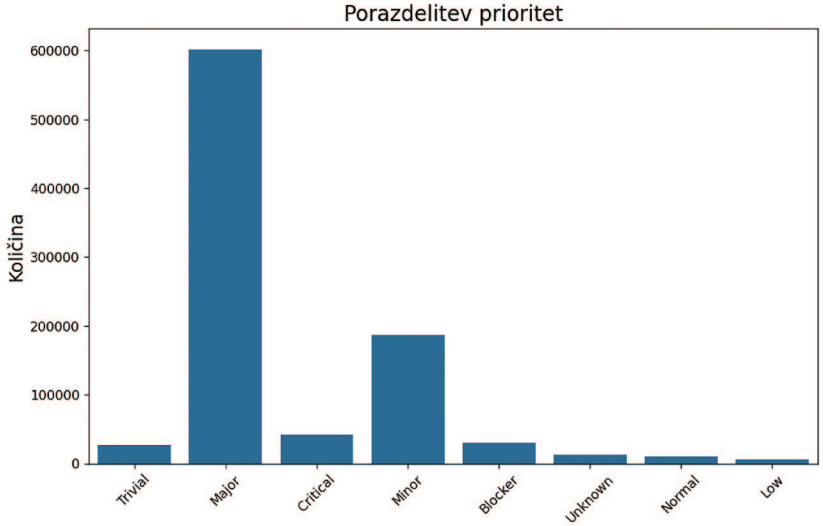
»Trivial«, »Major«, »Critical«, »Minor«, »Blocker«, »Unknown«, »Normal«, »Low« (slika 2).

Prioriteta »Major« je večinska (približno 80% vseh primerov sodi v ta razred), kar kaže na neuravnoteženost podatkov in jo je potrebno upoštevati pri nadaljnjem modeliranju.

3.3 Priprava podatkov

Priprava podatkov je pomemben korak v procesu razvoja modela strojnega učenja. Da bi se model učinkovito naučil in podal pravilne rezultate, je treba podatke očistiti in pretvoriti v ustrezno obliko. Sledili smo naslednjim korakom:

1. Nalaganje in začetna obdelava podatkov.
   Glavna naloga v tem koraku je bila naložiti podatke in se prepričati o njihovi pravilnosti.
2. Odstranitev zapisov z manjkajočimi podatki.
3. Odstranitev zapisov, ki vsebujejo dele programske kode, opredeljene s kombinacijami ključnih besed »if«, »for«, »class« ali dele SQL poizvedb, opredeljenih s kombinacijami ključnih besed »select«, »from«, »where« ali kombinacijami simbolov »:«, » : «, značilnih za podatke v obliki JSON.
4. Odstranitev kratkih zapisov, ki ne vsebujejo več kot ene besede, ob predpostavki, da kratki opisi nalog ne vsebujejo dovolj informacij za analizo.
5. Oblikovanje podatkov.
   Zaradi lažjega nadaljnjega dela smo podatke uredili v enotno obliko. Besedilo smo pretvorili v male črke, kar je omogočilo odpravo razlik med besedami, zapisanimi z različnimi črkami.
6. Tokenizacija in odstranitev stop besed.
   Tokenizacija je postopek razdelitve besedila na posamezne besede (»žetone«) [20]. Po tokenizaciji smo iz besedila odstranili stop besede (npr. prislovi in vezniki), ki za model niso nosile pomembnih informacij. V ta namen smo uporabili Pythonovo knjižnico »scikit-learn«. Vgrajeni seznam stop besed te knjižnice vključuje 318 besed, kot so »the«, »and«, »is«, »in«, »on«, »next«, »go«, »where«, »being« in druge.
7. Lematizacija.



Slika 2: **Porazdelitev podatkov po prioritetah.**

Lematizacija je postopek vračanja besed v njihovo osnovno obliko. Pri tem se odstranijo le pregibne končnice in vrne slovarsko obliko besede, ki je znana kot lema [20]. Za predobdelavo besedila smo uporabili lematizator WordNetLemmatizer iz knjižnice nltk (Pythonova knjižnica), ki je bese pretvoril v njihovo osnovno obliko. Na primer, besedi »following« in »thrown« sta bili pretvorjeni v obliko »follow« in »throw«. S tem se zmanjša število edinstvenih besed v besedilu in izboljša kakovost modela.

8. Vektorizacija besedila.
   Vektorizacija besedila je postopek pretvorbe besedila v številčno obliko. V našem primeru smo uporabili tehniko TF-IDF (angl. Term Frequency--Inverse Document Frequency), ki nam omogoča, da besede predstavimo kot številčne vektorje glede na njihovo pomembnost v dokumentu [21]. V raziskavi smo za vektorizacijo besedila uporabili razred TfidfVectorizer iz scikit-learn (knjižnica Python).
9. Uravnoteženje podatkov.
   V prvotnih podatkih smo opazili precejšnje neravnovesje razredov – naloge s prioriteto »Major« so predstavljale približno 80% vseh nalog. Neuravnoteženi podatki predstavljajo izziv pri natančnosti razločevanja med posameznimi razredi, saj se model nauči prepoznati večinski razred, ne pa tudi manjšinskih razredov. Za rešitev te težave smo uporabili postopek uravnoteženja razredov z enakomernim vzorčenjem (angl. undersampling). Pod uravnoteženjem razumemo spreminjanje sestave vzorcev v učnem naboru podatkov z zmanjšanjem števila primerov v večinski skupini, tako da ustrezajo velikosti manjšinske skupine. Pri tem smo iz večinske skupine naključno izbrali primere, dokler nismo dosegli enake velikosti kot manjšinska skupina, in nato združili podatke v enoten učni nabor.

Po čiščenju in pripravi podatkov nam je ostalo 51536 zapisov o nalogah v IT projektih. Po odstranitvi stop besed in lematizaciji je bila velikost slovarja 84328 unikatnih besed. Med najpogostejšimi besedami so bile: »I« (frekvenca 40711-krat), »code« (pojavnost 28132-krat), »use« (pojavnosti 27479-krat), »error« (pojavnost 24775-krat), »info« (pojavnost 23067-krat).

3.4 Modeliranje

Modeliranje je faza procesa CRISP-DM, v kateri z izbranimi metodami strojnega učenja razvijamo, ocenjujemo in optimiziramo klasifikacijski model. V naši raziskavi smo uporabili različne algoritme strojnega učenja za razvoj klasifikacijskih modelov za določanje prioritet nalog v IT projektih. Model strojnega učenja je funkcija, ki preslika nize vhodnih podatkov (neodvisne spremenljivke, ki jih predstavljajo opisi nalog) v izhodne podatke (odvisna spremenljivka Y oziroma klasifikacijski razred). Cilj modela je pravilno uvrstiti vrednost odvisne spremenljivke na podlagi vrednosti neodvisnih spremenljivk.

V našem modelu je spremenljivka x_i opis naloge v IT projektu, potem je X množica, ki vsebuje opise vseh nalog v IT projektih. Spremenljivka y_k je prioriteta naloge, ki lahko pripada enemu od razredov množice Y = {»Trivial«, »Major«, »Critical«, »Minor«, »Blocker«, »Unknown«, »Normal«, »Low«}.

Podatke smo razdelili na učno množico U , ki je definirana kot množica, na kateri se algoritem uči in je definirana, kot prikazuje enačba (1), in testno množico T, ki je definirana kot množica, na kateri se algoritem preizkusi. Učna množica je definirana z uporabo desetkratnega prečnega preverjanja (angl. cross-validation), kjer model ob vsakem koraku uporablja 9/10 podatkov za učenje, preostalo 1/10 pa za preizkušanje [22].

$$U = \{(x_1, y_k), (x_2, y_k), ..., (x_l, y_k)\}, \quad (1)$$

kjer je $y_k \in Y$, $k \in [1, m]$, m je število razredov prioritet, $x_i \in X$, $i \in [1, n]$, n je število vseh opisov nalog v IT projektih, $l < n$, l je velikost učne množice.

Cilj algoritmov klasifikacije je, da ob dani učni množici U najdejo naslednjo funkcijo (2):

$$h: x_i \longrightarrow y_k. \quad (2)$$

Zgradili smo več modelov, pri čemer smo spreminjali število razredov izhodne spremenljivke: osnovno z 8. razredi prioritet in različice, kjer smo obstoječih osem razredov združili v dva razreda (Y = {»High Priority«, »Low Priority«}.), tri razrede (Y = {»High Priority«, »Medium Priority«, »Low Priority«}) in štiri razrede (Y = {»High Priority«, »Critical Priority«,»Medium Priority«, »Low Priority«}).

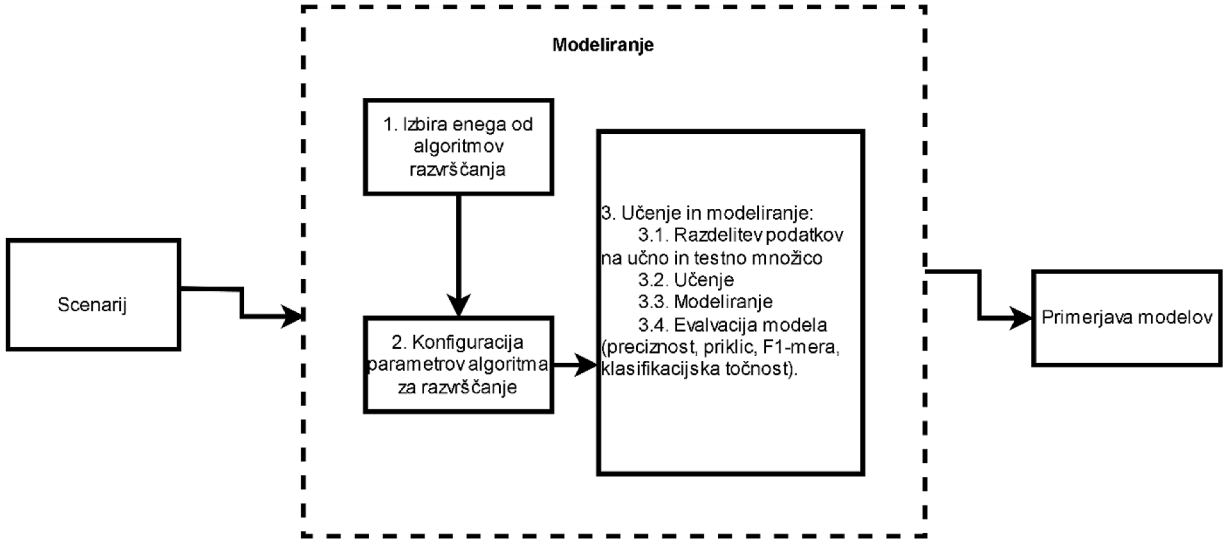
Vse postopke za modeliranje smo opisali z naslednjimi scenariji, ki so prikazani v tabeli 1.

Tabela 1: **Scenariji modeliranja.**

Naziv scenarija	Količina razredov	Opis	Število primerov
Scenarij 1	8	Vključuje prioritete »Trivial«, »Major«, »Critical«, »Minor«, »Blocker«, »Unknown«, »Normal«, »Low«. Podatki niso uravnoteženi	26508/601876 /42136/186604 /30258/12103 /10103/6442
Scenarij 2	8	Vključuje iste prioritete, kot pri scenariju 1, vendar podatki so uravnoteženi (»Trivial«, »Major«, »Critical«, »Minor«, »Blocker«, »Unknown«, »Normal«, »Low«)	6442 v vsakem razredu
Scenarij 3	8	Vključuje iste prioritete, kot pri scenariju 2, vendar podatki so uravnoteženi in programska koda je odstranjena(»Trivial«, »Major«, »Critical«, »Minor«, »Blocker«, »Unknown«, »Normal«, »Low«)	1430 v vsakem razredu
Scenarij 4	2	Dva razreda prioritet: 1) »High Priority«, v katerega smo vključili naloge s prioritetami »Blocker«, »Critical« in »Major«, 2) »Low Priority«, v katerega smo vključili naloge s prioritetami »Minor«, »Trivial«, »Normal«, »Unknown«, »Low«.	241760 v vsakem razredu
Scenarij 5	3	Tri razreda: naloge s prioriteto »Major«, »Blocker«, »Critical« smo uvrstili v razred 1)»High Priority«. V razred 2)»Low Priority« smo uvrstili naloge s prioriteto »Unknown«, »Low« in »Minor«. Naloge s »Trivial« in »Normal« prioritete smo uvrstili v razred 3)»Medium Priority«.	23771 v vsakem razredu
Scenarij 6	4	Štiri razreda. Skupina 1)»High Priority« vključuje naloge s prioriteto »Major«. Skupina 2)»Critical Priority« vključuje naloge s prioriteto »Blocker« in »Critical«. Skupina 3)»Medium Priority« vključuje naloge s prioriteto »Minor« in »Trivial«. Skupina 4)»Low Priority« združuje naloge označene z »Normal«, »Unknown« in »Low«.	28648 v vsakem razredu

Iz tabeli 1 je razvidno, da je bilo pri scenarijih od 1 do 3 število razredov izvirno (8 razredov). Pri scenariju 2 smo preverili, kako vpliva na rezultate klasifikacije uravnoteženost podatkov. Za uravnoteženost smo uporabili postopek uravnoteženja razredov z enakomernim vzorčenjem. Pri scenariju 3 smo preverili, kako vplivata na rezultate klasifikacije uravnoteženost podatkov in odstranitev programske kode. Pri scenarijih od 4 do 6 smo preverili, ali se rezultati klasifikacije izboljšajo, če podatke razvrstimo v manjše število razredov.

Shematsko je proces modeliranja prikazan na sliki 3.



Slika 3: **Postopek modeliranja**

Slika 3 prikazuje postopek modeliranja na podlagi različnih scenarijev. Najprej izberemo enega izmed izbranih algoritmov klasifikacije, nato prilagodimo njegove parametre. Podatke razdelimo na učne in testne množice, kar omogoča začetek postopka učenja modela. Sledi modeliranje, v katerem izračunamo metrike modela (preciznost, priklic in F1-mera). Postopek modeliranja izvedemo za vse scenarije z izbranimi algoritmi.

Pri modeliranju smo uporabili naslednje klasifikatorje: naključni gozd (angl. Random forest), metoda podpornih vektorjev (angl. support vector machine, SVM), logistična regresija (angl. Logistic regression), gradientno povečanje z drevesi odločanja (angl. Gradient Boosting with Decision Trees) in k-najbližjih sosedov (angl. k-Nearest Neighbors – kNN).

Razlogi za izbiro teh algoritmov izhajajo iz izkušenj preteklih raziskav. Algoritem naključni gozd dosledno zagotavlja visoko natančnost in robustnost, zlasti pri obravnavi velikih in zapletenih podatkovnih nizov [23]. Metoda podpornih vektorjev se je izkazala za učinkovito pri nalogah, ki zahtevajo visoko natančnost klasifikacije, zlasti, kadar je podatke težko ločiti [24]. Gradientno povečanje z drevesi odločanja sodi med ansambelske metode, ki s postopnim zmanjševanjem napake oziroma optimizacijo iz šibkih klasifikatorjev gradi močne klasifikatorje. V praksi kaže odlične rezultate pri nalogah, ki zahtevajo natančno klasifikacijo, in lahko znatno izboljša učinkovitost v primerjavi z modeli z eno samo optimizacijo, kot so odločitvena drevesa [25]. Logistična regresija se zaradi svoje preprostosti in učinkovitosti pogosto uporablja v različnih aplikacijah strojnega učenja, kjer je potrebno zanesljivo napovedovanje kategorij [26]. Algoritem k-najbližjih sosedov smo izbrali zaradi preprostosti izvajanja [27].

3.5 Vrednotenje modelov

Za vrednotenje pridobljenih rezultatov modeliranja smo uporabili naslednje metrike: klasifikacijska točnost (angl. Classification Accuracy), preciznost (angl. Precision), priklic (angl. Recall), F1-mera (angl. F1-score). Metrike, ki jih upoštevamo, temeljijo na naslednjih rezultatih: pravilno klasificirane kot pozitivne (angl. true positive – TP), pravilno klasificirane kot negativne (angl. true negative – TN), napačno klasificirane kot pozitivne (angl. false positive – FP) in napačno klasificirane kot negativne (angl. false negative – FN). V tabeli 2 prikazujemo osnovne enačbe za izračun metrik za vrednotenje modelov za binarno in večrazredno klasifikacijo. Uporabili smo enačbe za večrazredno klasifikacijo [16], [28], [29] (stolpec 3 tabele 2).

Za ocenjevanje kakovosti modelov smo uporabili tudi krivuljo delovne karakteristike sprejemnika (angl. Receiver Operating Characteristic –ROC krivulja), ploščino pod ROC krivuljo (angl. Area Under ROC Curve – AUC ROC), krivuljo preciznost-priklic (angl. Precision-Recall curve) in ploščino pod krivuljo preciznost-priklic (angl. Area Under the Precision-Recall Curve – AUC PR) [30], [31].

Krivulja ROC prikazuje razmerje med pogostostjo pravilno klasificiranih pozitivnih rezultatov (TP, na osi Y) in pogostostjo napačno klasificiranih pozitivnih rezultatov (FP, na osi X) ob spreminjanju praga razlikovanja [28], [32]. To nam pomaga razumeti, kako dobro naš model ločuje med pozitivnimi in negativnimi primeri, saj lahko spremljamo razmerje med pravilno prepoznanimi pozitivnimi primeri (TP) in napačno prepoznanimi negativnimi primeri (FP) pri različnih nastavitvah pragov. V osnovi je namenjena analizi binarne klasifikacije. Posebej je treba omeniti pristope za izračun ROC krivulj za večrazre-

Tabela 2: **Enačbe za izračun mer točnosti klasifikacije.**

Metrika	Izračun binarne klasifikacije	Izračun večrazredne klasifikacije, k-število razredov
1	2	3
Klasifikacijska točnost (ACCUR)	ACCUR = $(TP + TN)/(TP + TN + FN + FP)$	Povprečno ACCUR = $\sum_{i=1}^{K} \text{ACCUR}_i/K$
Preciznost (PREC)	PREC = $TP/(TP + FP)$	Povprečno PREC = $\sum_{i=1}^{K} \text{PREC}_i/K$
Priklic (REC)	REC = $TP/(TP + FN)$	Povprečno REC = $\sum_{i=1}^{K} \text{REC}_i/K$
F1-mera (F1)	F1 = 2 * PREC * REC/(PREC + REC)	Povprečno F1 = $\sum_{i=1}^{K} \text{F1}_i/K$

Tabela 3: **Glavni dobljeni rezultati modeliranja (algoritem naključni gozd).**

Metrike	Scenarij 1	Scenarij 2	Scenarij 3	Scenarij 4	Scenarij 5	Scenarij 6
1	2	3	4	5	6	7
Preciznost	0,20	0,41	0,36	0,74	0,57	0,53
Priklic	0,15	0,41	0,37	0,75	0,57	0,54
F1-mera	0,13	0,41	0,36	0,74	0,57	0,53
Klasifikacijska točnost	0,64	0,41	0,37	0,75	0,69	0,61
AUC ROC	–	0,81	0,75	0,82	0,77	0,80
AUC PR	0,32	0,43	0,37	0,80	0,64	0,56

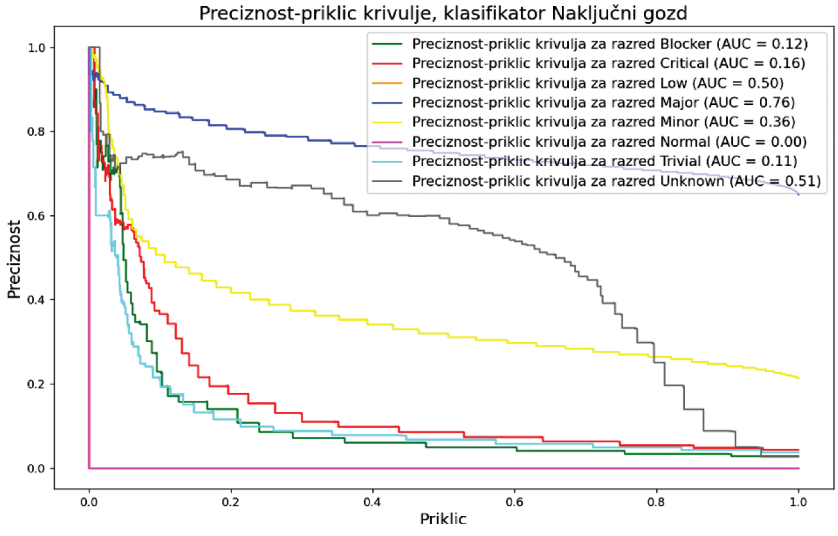
dno klasifikacijo in neuravnotežene podatke. Obstajata dva pristopa za izračun ROC krivulje za klasifikacijo več razredov: pristop »eden proti vsem« (angl. one versus all – OvA) in pristop »eden proti enemu« (angl. one versus one – OvO) [30], [33]. Pristop OvA temelji na učenju niza neodvisnih binarnih klasifikatorjev, v katerih je en razred pozitiven, vsi drugi pa negativni. Po izračunu ROC krivulje za vsak binarni klasifikator se krivulje povprečijo, da dobimo končno ROC krivuljo [30]. Pri pristopu OvO se za vsako možno kombinacijo kategorij nauči binarni klasifikator. Če je na primer treba razlikovati tri kategorije, se razvijejo trije ločeni binarni klasifikatorji (razred 1 proti razredu 2, razred 1 proti razredu 3 in razred 2 proti razredu 3). Ko je izračunana ROC krivulja za vsak binarni klasifikator, se krivulje združijo in dobimo končno krivuljo ROC [34]. V našem primeru so enačbe za izračun metrik binarnih klasifikatorjev in metrik več razredov podane v tabeli 2.

Na neuravnoteženih podatkih pa se še bolje, kot ROC krivulja izkaže krivulja preciznost-priklic, ki prikazuje razmerje med deležem pravilno klasificiranih pozitivnih primerov med vsemi pozitivnimi primeri, torej pravilno pozitivno klasificiranih in nepravilno negativno klasificiranih) [28] saj se stopnja lažno pozitivnih rezultatov pri zelo neuravnoteženih podatkih zmanjša zaradi velikega števila resničnih negativnih rezultatov [35], [36]. Za oceno kakovosti naučenih modelov smo uporabili tako krivuljo preciznost-priklic (za modele naučene na neuravnoteženih podatkih), medtem ko smo za modele, naučene na uravnoteženih podatkih uporabili ROC krivuljo.

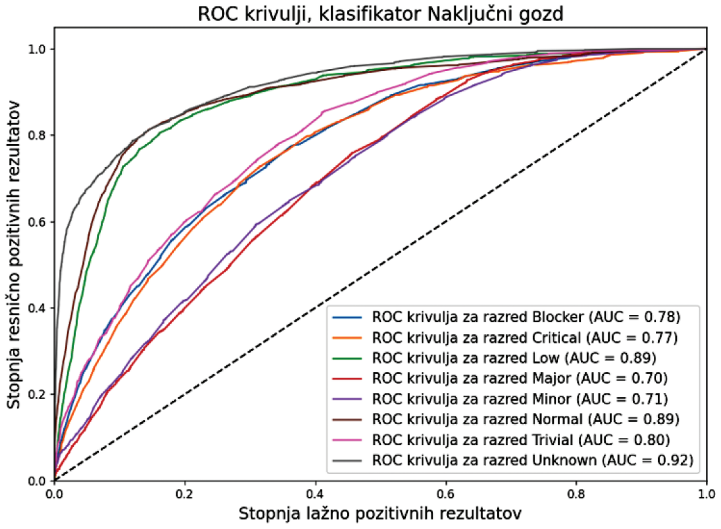
4 REZULTATI

V tem poglavju predstavljamo rezultate modeliranja in oceno njihove uspešnosti z uporabo ustreznih metrik.

Modeliranje smo izvedli na predhodno pripravljenih podatkih o nalogah iz različnih IT projektov.



Slika 4: **Krivulje Preciznost-Priklic za scenarij 1 (algoritem naključni gozd).**



Slika 5: **Krivulje ROC za scenarij 2 (algoritem naključni gozd).**

Primerjali smo rezultate modeliranja za scenariji 1 – 6. Uporabili smo algoritme naključni gozd, metoda podpornih vektorjev, logistična regresija, gradientno povečanje z drevesi odločanja, k-najbližjih sosedov. Kriterije za preverjanje kakovosti naučenih modelov za različne scenarije prikazujemo v tabeli 3.

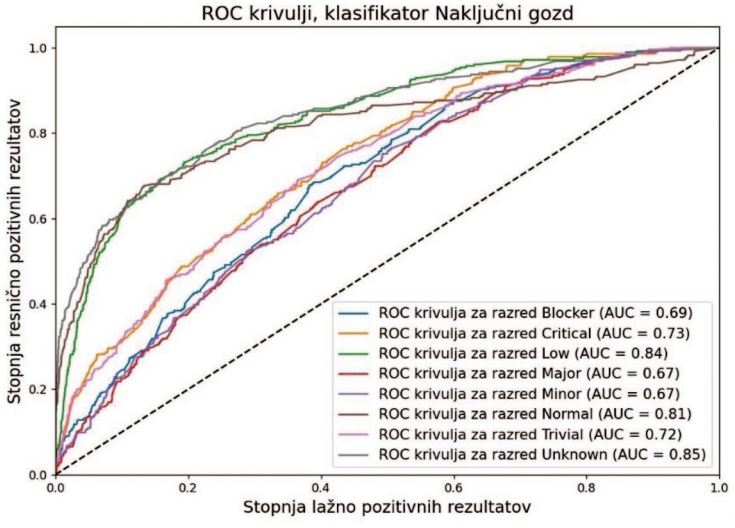
V nadaljevanju rezultate za posamezne scenarije podrobneje opišemo.

Prvi scenarij (scenarij 1) je predvideval modeliranje brez predhodnega postopka uravnoteženja podatkov. Rezultate testiranja modela prikazujemo s krivuljo preciznost-priklic (slika 4).

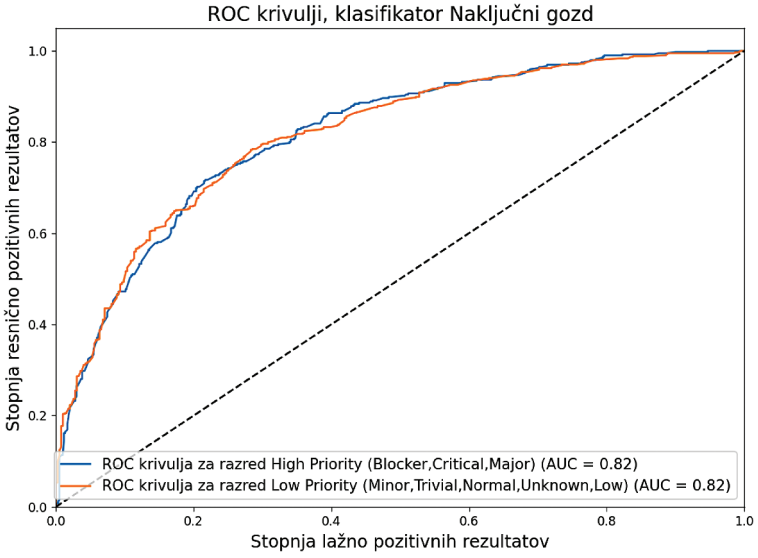
Iz slike 4 je razvidno, da ima razred »Major« (0,76) najvišjo vrednost AUC PR. Razreda »Unknown« (0,51) in »Low« (0,50) imata zmerni vrednosti AUC PR. Preostali razredi imajo nizke vrednosti AUC PR. Iz tabele 3 (stolpec 2) je razvidno, da so metrike priklic, preciznost in F1-mera zelo nizke.

V scenariju 2 smo zaradi uravnoteženja podatkov za oceno rezultatov uporabili ROC-krivulje. Rezultati za scenarij 2 so prikazani na sliki 5.

S slike 5 je razvidno, da model najbolje od ostalih razredov loči razrede »Unknown«, »Low« in »Normal« z vrednostmi AUC ROC 0,92 in 0,89. Za razred »Trivial« model kaže rezultate z AUC ROC okoli 0,80. Vrednosti AUC ROC za razrede »Blocker«, »Critical«, »Major« se gibljejo med 0,70 in 0,78.. Povprečne vrednosti metrik preciznosti, priklica in F1-mere za vse razrede so približno 0,41 (tabela 3, stolpec 3).



Slika 6: **Krivulje ROC za scenarij 3 (algoritem naključni gozd).**



Slika 7: **Krivulji ROC za scenarij 4 (algoritem naključni gozd).**

Pri scenariju 3 (odstranitev programske kode) so bili rezultati klasifikacije slabši (tabela 3 stolpec 4). Preciznost je bila 0,36, priklic 0,37, F1-mera 0,36. ROC krivulji za scenarij 3 so prikazani na sliki 6.

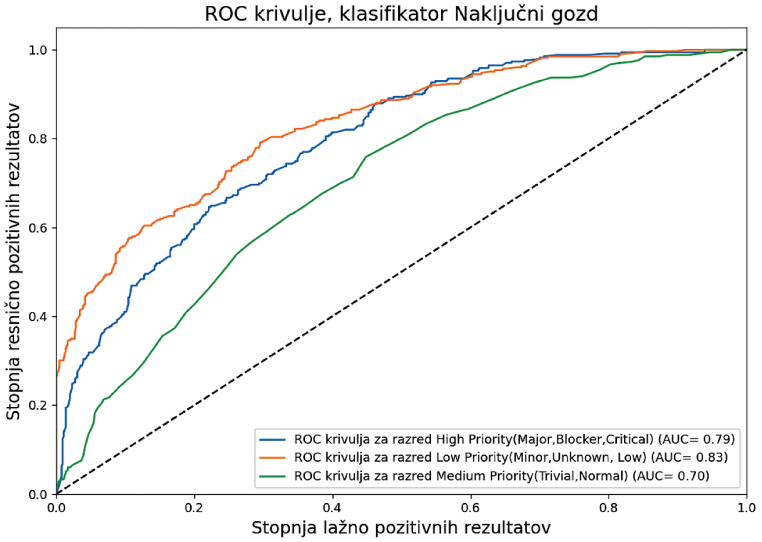
S slike 6 je razvidno, da razred »Unknown« dosega najvišjo vrednost AUC ROC 0,85. Sledijo razredi »Critical« (AUC ROC= 0,73) in »Low« (AUC ROC = 0,84). Razredi »Blocker«, »Major«, in »Normal« imajo srednje vrednosti AUC ROC, ki se gibljejo med 0,67 in 0,81. Najslabše rezultate kaže razred »Trivial«, kjer vrednost AUC ROC znaša približno 0,75.

Dobljene vrednosti za scenarij 4 so prikazane na sliki 7 ter v tabeli 3 (stolpec 5).

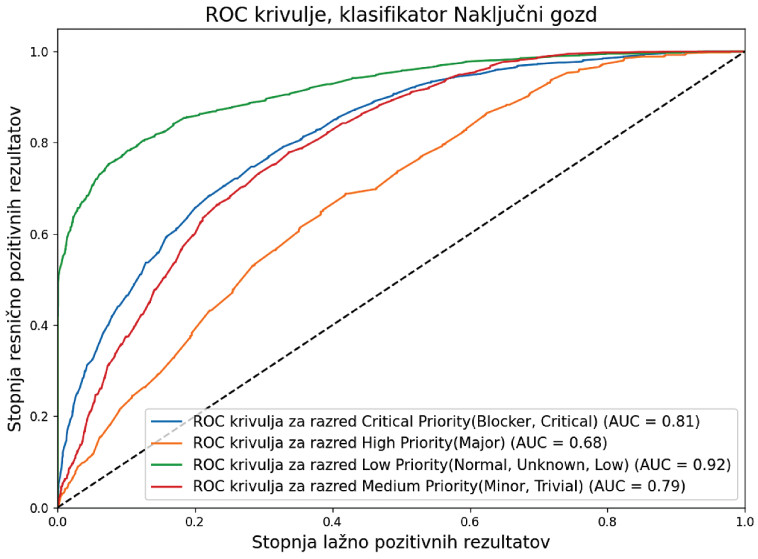
Na sliki 7 je za skupino razredov s prioriteto »Low Priority« in prioriteto »High Priority« AUC ROC 0,82. Tudi vrednosti metrik so precej visoke in znašajo 0,74 (0,75) (tabela 3 stolpec 5).

Na sliki 8 so prikazane vrednosti ploščine pod ROC krivuljo, pridobljene po razdelitvi podatkov v tri razrede (scenarij 5).

Slika 8 prikazuje, da imata dve skupini (»High Priority« in »Low Priority«) vrednosti AUC ROC 0,79 in 0,83. Skupina razredov s prioriteto »Medium Priority« ima vrednost AUC ROC 0,70. Ta rezultat je v primerjavi z drugima dvema skupinama razredov izrazito nižji. Model je dosegel klasifikacijsko točnost 0,69 (tabela 3 stolpec 6).



Slika 8: **Krivulje ROC za scenarij 5 (algoritem naključni gozd).**



Slika 9: **Krivulje ROC za scenarij 6 (algoritem naključni gozd).**

Slika 9 prikazuje krivulje ROC za scenarij 6 (štiri skupine razredov). S slike 9 je razvidno, da model dosega najboljše rezultate za razrede s prioriteto »Low Priority«, kjer AUC ROC doseže vrednost 0,92. Model prav tako uspešno klasificira razreda »Critical Priority« in »Medium Priority«, pri čemer vrednost AUC ROC znaša 0,81 oziroma 0,79. Vendar ima model pri razredu »High Priority« precejšnje težave, kar dokazuje nizka vrednost AUC ROC, ki znaša 0,68.

Povprečne vrednosti metrik so približno 0,53 (tabela 3 stolpec 7), skupna točnost modela pa je 0,61.

5 DISKUSIJA

Glavni cilj naše naloge je bil razviti modele strojnega učenja za klasifikacijo nalog v IT projektih po prioritetah, kar bi olajšalo načrtovanje in izvedbo razvojnih procesov programske opreme.

V kontekstu naše raziskave in rezultatov analize podatkov lahko oblikujemo naslednje ugotovitve, ki jih povzemamo v tabeli 4.

Rezultati modeliranja kažejo na pomembne ugotovitve glede uspešnosti modelov v različnih scenarijih. Pri scenariju 1 (neuravnoteženi podatki, osem razredov) smo pričakovano dosegli najslabše rezultate kakovosti klasifikacijskega modela pri vseh uporabljenih algoritmih. Dobili smo nizke vrednosti metrik, kot so preciznost 0,20, priklic 0,15, F1-mera 0,13, pri tem je AUC PR 0,32. To poudarja potrebo po izboljšanju modela, zlasti z uporabo metod za uravnoteženje podatkov, da bi izboljšali uspešnost za razrede z nizko površino pod krivuljo preciznost-priklic.

Pri scenariju 2 (uravnoteženi podatki, osem razredov) smo z uporabo metode uravnoteženja podatkov (SMOTE) dosegli opazno izboljšanje kako-

Tabela 4: **Pridobljeni rezultati.**

Pristopi modeliranja	**Rezultat**
Uravnoteženje podatkov (scenarij 2)	izboljšalo
Odstranitev zapisov s programsko kodo (scenarij 3)	poslabšalo
Uporaba različnih algoritmov za razvrščanje (vsi scenariji)	brez vpliva, razen k-najbližjih sosedov (dobljeni rezultati za ta algoritem so bili vedno slabši)
Razdelitev v dva razreda (scenarij 4)	izboljšalo
Razdelitev v tri razrede (scenarij 5)	izboljšalo
Razdelitev v štiri razrede (scenarij 6)	izboljšalo

vosti klasifikacijskega modela. Vrednosti metrik so se znatno povečale: preciznost se je zvišala z 0,20 na 0,41, priklic prav tako na 0,41, F1-mera na 0,41, medtem ko je AUC ROC dosegel 0,81. Povečanje teh metrik je rezultat boljše obravnave manj zastopanih razredov, kar omogoča bolj uravnoteženo delovanje modela. Ti rezultati potrjujejo, da ima uravnoteženje podatkov ključno vlogo pri izboljšanju učinkovitosti modela, še posebej pri klasifikaciji razredov z manjšo zastopanostjo.

Pri scenariju 3 (odstranitev zapisov s programsko kodo, osem razredov) smo opazili poslabšanje rezultatov modela. Preciznost se je znižala z 0,41 na 0,36, priklic je padel z 0,41 na 0,37, F1-mera se je znižala z 0,41 na 0,36, klasifikacijska točnost je zmanjšala z 0,41 na 0,37, je AUC ROC padla z 0,81 na 0,75. Odstranitev zapisov, ki so vsebovali programsko kodo, je negativno vplivala na zmogljivost modela, saj ti podatki pogosto vsebujejo ključne informacije, ki so pomembne za razvrščanje nalog. Ta scenarij poudarja potrebo po skrbni analizi podatkov pred njihovo odstranitvijo, da bi se izognili izgubi pomembnih informacij. Dobljeni rezultati so boljši v primerjavi z rezultati pri scenariju 1.

Pri scenariju 4 smo osem razredov preslikali v nova dva razreda (»High Priority« in »Low Priority«), kar je prineslo najstabilnejše in najnatančnejše rezultate, kar je tudi znano iz teorije in preteklih raziskav [37]. V primerjavi s scenarijem 3 se je preciznost povečala z 0,36 na 0,74, priklic je narasel z 0,37 na 0,75, F1-mera se je izboljšala z 0,36 na 0,74, klasifikacijska točnost pa se je prav tako dvignila z 0,37 na 0,75. AUC ROC je pokazal rahlo izboljšanje z 0,75 na 0,82. Vrednost AUC ROC je bila v tem scenariju najvišja, klasifikacija pa zanesljivejša. Poenostavljena klasifikacijska shema je omogočila bolj jasno razlikovanje med nalogami in izboljšala predvidljivost modela, kar je še posebej pomembno pri nalogah, kjer lahko klasifikacijske napake povzročijo resne posledice.

Pri scenariju 5 (oblikovanje treh razredov: »High Priority«, »Medium Priority« in »Low Priority«) smo opazili izboljšanje v primerjavi s scenariji 1–3. Ta razdelitev omogoča bolj natančno razlikovanje nalog glede na prioritete. Kljub izboljšavam so rezultati v primerjavi s scenarijem 4 nekoliko slabši. Preciznost se je zmanjšala iz 0,74 na 0,57, priklic se je znižal iz 0,75 na 0,57, F1-mera se je poslabšala iz 0,74 na 0,57, medtem ko je klasifikacijska točnost padla iz 0,75 na 0,69. AUC ROC je upadel iz 0,82 na 0,77. Vrednosti metrik in klasifikacijska točnost kažejo, da model deluje zadovoljivo.

Pri scenariju 6 (razdelitev v štiri razrede) so rezultati presegli rezultate iz scenarijev 1–3. Model je bolje razvrstil različne nivoje prioritete, pri čemer so vrednosti preciznosti, priklica in F1-mere ostale na zadovoljivi ravni. Kljub temu je bila uspešnost nekoliko slabša v primerjavi s scenarijem 5 (z razdelitvijo v tri razrede). Preciznost se je zmanjšala iz 0,57 na 0,53, priklic je padel iz 0,57 na 0,54, prav tako se je F1-mera znižala iz 0,57 na 0,53. Klasifikacijska točnost se je zmanjšala iz 0,69 na 0,61. Pri AUC ROC opazimo rahlo izboljšanje z 0,77 na 0,80, medtem ko je AUC PR padla z 0,64 na 0,56. To je še posebej pomembno pri nalogah, kjer so napake pri razvrščanju lahko kritične. V podobnih primerih je lahko koristno poenostaviti razvrščanje in se osredotočiti na ključne kategorije, da bi dosegli optimalno točnost.

Naša raziskava je pokazala, da je zmanjšanje števila razredov prioritet nalog v IT projektih dobra rešitev za izboljšanje rezultatov razvrščanja nalog po prioritetah. Uravnoteženje podatkov in zmanjšanje števila prednostnih razredov nalog sta prispevala k boljšim rezultatom. Ta pristop resnično poenostavi postopek in daje dobre rezultate. Hkrati se zavedamo, da je njegova uporaba v resničnih projektih lahko omejena zaradi raznolikosti in zapletenosti nalog. Uporaba različnih algoritmov za razvrščanje ni bistveno vplivala na rezultate. Algoritem k-najbližjih sosedov je v primerjavi z drugimi algoritmi razvrščanja pokazal bistveno slabše rezultate. Razlago za to vidimo v kompleksnosti analiziranih podatkov. Izbira algoritma za razvrščanje igra vlogo, vendar je ta vpliv v primerjavi z drugimi dejavniki manj izrazit.

Pomemben dejavnik, ki vpliva na kakovost razvrščanja, ostaja način opisovanja nalog. V večini primerov sam opis naloge IT projekta ni popoln, ni jasen in celo ni zaporeden, v samih opisih je težko zaslediti kakršno koli strukturiranost. Vse to lahko prispeva k slabši kakovosti naučenih klasifikatorjev. Posledično obstaja potreba po standardizaciji opisov, saj lahko jasno in strukturirano besedilo bistveno izboljša natančnost odkrivanja vzorcev in razvrščanja.

6 ZAKLJUČEK

V prispevku smo obravnavali uporabo metod strojnega učenja za samodejno določanje prioritet nalog v IT projektih. V ta namen smo po metodologiji CRISP-DM razvili klasifikacijske modele z uporabo

algoritmov naključni gozd, metoda podpornih vektorjev, logistična regresija, gradientno povečanje z drevesi odločanja in k-najbližjih sosedov. Raziskavo smo izvedli na podatkovni množici, ki vsebuje številne zapise projektnih nalog za različne IT projekte iz javno dostopnega vira Zenodo v obliki JSON-datoteke. Pri tem smo naslovili dva problema: neuravnoteženost podatkov in večrazredno klasifikacijo. Oblikovali smo osem scenarijev ter preučili, kako le-ti vplivajo na kakovost klasifikacijskih modelov. Pokazali smo, da uporaba algoritmov strojnega učenja omogoča razvoj klasifikacijskega modela, kar lahko prispeva k učinkovitejšemu vodenju projektov, še posebej v kontekstu velikih količin podatkov in zapletenih struktur nalog.

Raziskava je pokazala, da uravnoteženje podatkov pri učenju modelov strojnega učenja vpliva na doseganje večje natančnosti razvrščanja, kar je še posebej opazno pri velikem številu razredov. Prav tako smo pokazali vpliv števila razredov na uspešnost klasifikacijskega modela, kar tudi sicer predstavlja velik izziv pri transparentnem določanju prioritet nalog in izvajanju projektov. V praksi je tako potrebno poiskati ravnovesje med številom razredov prioritet, ki projektnemu timu še zagotavljajo učinkovito vodenje in izvajanje projektov ter natančnostjo klasifikacijskega modela, s katerim lahko podpremo projektne vodje pri določanju prioritet nalog.

Uporaba strojnega učenja za določanje prioritet nalog predstavlja obetavno področje, ki lahko bistveno izboljša učinkovitost vodenja IT projektov. Rezultati raziskave so pomemben prispevek k praksi avtomatiziranega določanja prioritet in omogočajo boljše upravljanje z nalogami v IT projektih. Prihodnje raziskave se lahko usmerijo v nadaljnji razvoj algoritmov in njihovo integracijo v obstoječe sisteme za vodenje projektov, kar bo omogočilo široko uporabo v poslovnem okolju ter izboljšanje učinkovitosti in uspešnosti projektnega vodenja.

ZAHVALA

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

LITERATURA

[1] W. Wang in K. Siau, »Artificial Intelligence, Machine Learning, Automation, Robotics, Future of Work and Future of Humanity: A Review and Research Agenda«, *J. Database Manag.*, let. 30, str. 61–79, maj 2022, doi: 10.4018/978-1-6684-6291-1.ch076.

[2] M. Nenni, F. De Felice, C. De Luca, in A. Forcina, »How artificial intelligence will transform project management in the age of digitization: a systematic literature review«, *Manag. Rev. Q.*, apr. 2024, doi: 10.1007/s11301-024-00418-z.

[3] H. Dinendra, C. Rajapakse, in P. Asanka, »Personalized Classification of Non-Spam Emails Using Machine Learning Techniques«, sep. 2022, str. 171–177. doi: 10.1109/SCSE56529.2022.9905110.

[4] R. K. Karn, V. E. Jesi, in S. M. Aslam, »Spam Email Detection Using Machine Learning Integrated In Cloud«, *2023 Int. Conf. Netw. Commun. ICNWC*, Pridobljeno: 28. september 2024. [Na spletu]. Dostopno na: https://www.academia.edu/102418331/SPAM_EMAIL_DETECTION_USING_MACHINE_LEARNING_INTEGRATED_IN_CLOUD

[5] AXELOS Limited, Ur., *Managing successful projects with PRINCE2*, 6th edition. London Norwich: TSO, 2017.

[6] IIBA, »BABOK® Guide Glossary | IIBA®«. Pridobljeno: 12. julij 2024. [Na spletu]. Dostopno na: https://www.iiba.org/knowledgehub/glossary/

[7] PMI, *The standard for project management and a guide to the project management body of knowledge (PMBOK guide).*, Newtown Square, Pennsylvania 19073-3299 USA., julij 2021. [Na spletu]. Dostopno na: https://ibimone.com/PMBOK%207th%20Edition%20(iBIMOne.com).pdf

[8] P. Achimugu, A. Selamat, R. Ibrahim, in M. N. Mahrin, »A systematic literature review of software requirements prioritization research«, *Inf. Softw. Technol.*, let. 56, št. 6, str. 568–585, jun. 2014, doi: 10.1016/j.infsof.2014.02.001.

[9] F. A. Bukhsh, Z. A. Bukhsh, in M. Daneva, »A systematic literature review on requirement prioritization techniques and their empirical evaluation«, *Comput. Stand. Interfaces*, let. 69, str. 103389, mar. 2020, doi: 10.1016/j.csi.2019.103389.

[10] R. Qaddoura, A. Abu srhan, M. Haj Qasem, in A. Hudaib, »Requirements Prioritization Techniques Review and Analysis«, okt. 2017, str. 258–263. doi: 10.1109/ICTCS.2017.55.

[11] Y. Bugayenko *idr.*, »Prioritizing tasks in software development: A systematic literature review«, *PLOS ONE*, let. 18, št. 4, str. e0283838, 2023, doi: 10.1371/journal.pone.0283838.

[12] M. Narendhar in D. K. Anuradha, »Different Approaches of Software Requirement Prioritization«.

[13] D. Cubranic in G. C. Murphy, »Automatic bug triage using text categorization«, SEKE 2004: Proceedings of the Sixteenth International Conference on Software Engineering & Knowledge Engineering., 2004, str. 92-97. Pridobljeno: 23. september 2024. [Na spletu]. Dostopno na: https://www.cs.ubc.ca/labs/spl/papers/2004/seke04-bugzilla.pdf

[14] J. Xuan, J. He, Z. Ren, J. Yan, in Z. Luo, »Automatic Bug Triage using Semi-Supervised Text Classification.«, jan. 2010, str. 209–214.

[15] J. Brownlee, *Imbalanced Classification with Python: Better Metrics, Balance Skewed Classes, Cost-Sensitive Learning*. Independently published, 2021.

[16] Yanmin Sun, A. Wong, in M. S. KAMEL, »Classification of imbalanced data: a review«, *Int. J. Pattern Recognit. Artif. Intell.*, let. 23, nov. 2011, doi: 10.1142/S0218001409007326.

[17] F. Thabtah, S. Hammoud, F. Kamalov, in A. Gonsalves, »Data imbalance in classification: Experimental evaluation«, *Inf. Sci.*, let. 513, str. 429–441, mar. 2020, doi: 10.1016/j.ins.2019.11.004.

[18] C. Shearer, »The CRISP-DM Model: The New Blueprint for Data Mining«, let. 5, št. 4, str. 13–22, 2000.

[19] L. Montgomery, C. Lüders, in Prof. Dr. W. Maalej, »The Public Jira Dataset«. Zenodo, 25. januar 2022. doi: 10.5281/zenodo.5901804.
[20] C. Manning, P. Raghavan, in H. Schuetze, *Introduction to Information Retrieval*. Cambridge, England: Cambridge University Press, 2009.
[21] M. Arora, V. Mittal, in P. Aggarwal, »Enactment of tf-idf and word2vec on Text Categorization«, v *Proceedings of 3rd International Conference on Computing Informatics and Networks*, A. Abraham, O. Castillo, in D. Virmani, Ur., Singapore: Springer, 2021, str. 199–209. doi: 10.1007/978-981-15-9712-1_17.
[22] D. Berrar, »Cross-Validation«, 2018. doi: 10.1016/B978-0-12-809633-8.20349-X.
[23] L. Breiman, »Random Forests«, *Mach. Learn.*, let. 45, št. 1, str. 5–32, okt. 2001, doi: 10.1023/A:1010933404324.
[24] C. Cortes in V. Vapnik, »Support-vector networks«, *Mach. Learn.*, let. 20, št. 3, str. 273–297, sep. 1995, doi: 10.1007/BF00994018.
[25] J. Friedman, »Greedy Function Approximation: A Gradient Boosting Machine«, *Ann. Stat.*, let. 29, nov. 2000, doi: 10.1214/aos/1013203451.
[26] M. Collins, R. E. Schapire, in Y. Singer, »Logistic Regression, AdaBoost and Bregman Distances«, *Mach. Learn.*, št. 48(1/2/3), jan. 2002.
[27] T. Cover in P. Hart, »Nearest neighbor pattern classification«, *IEEE Trans. Inf. Theory*, let. 13, št. 1, str. 21–27, jan. 1967, doi: 10.1109/TIT.1967.1053964.
[28] C. C. Aggarwal, *Data Mining: The Textbook*. New York, USA: Springer, 2015.
[29] A. Hevapathige, »Binary and Multi-Class Classification Using Supervised Machine Learning Algorithms and Ensemble Model«, sep. 2021.
[30] M. Majnik in Z. Bosnic, »ROC analysis of classifiers in machine learning: A survey«, *Intell. Data Anal.*, let. 17, str. 531–558, maj 2013, doi: 10.3233/IDA-130592.
[31] J. Czakon, »F1 Score vs ROC AUC vs Accuracy vs PR AUC: Which Evaluation Metric Should You Choose?«, neptune.ai. Pridobljeno: 14. julij 2024. [Na spletu]. Dostopno na: https://neptune.ai/blog/f1-score-accuracy-roc-auc-pr-auc
[32] A. P. Jawalkar *idr.*, »Early prediction of heart disease with data analysis using supervised learning with stochastic gradient boosting«, *J. Eng. Appl. Sci.*, let. 70, št. 1, str. 122, okt. 2023, doi: 10.1186/s44147-023-00280-y.
[33] A. Alnuaimi in T. Albaldawi, »An overview of machine learning classification techniques«, *BIO Web Conf.*, let. 97, str. 00133, apr. 2024, doi: 10.1051/bioconf/20249700133.
[34] K. Marshall, »How to Use the AUC ROC Curve for the Multi-class Model?«, Deepchecks. Pridobljeno: 14. julij 2024. [Na spletu]. Dostopno na: https://deepchecks.com/question/how-to-use-the-auc-roc-curve-for-the-multi-class-model/
[35] T. Saito in M. Rehmsmeier, »The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets«, *PLoS ONE*, let. 10, št. 3, str. e0118432, mar. 2015, doi: 10.1371/journal.pone.0118432.
[36] J. E. de la Calle, »How and Why I Switched from the ROC Curve to the Precision-Recall Curve to Analyze My Imbalanced…«, Medium. Pridobljeno: 13. julij 2024. [Na spletu]. Dostopno na: https://juandelacalle.medium.com/how-and-why-i-switched-from-the-roc-curve-to-the-precision-recall-curve-to-analyze-my-imbalanced-6171da91c6b8
[37] C. Thrampoulidis, S. Oymak, in M. Soltanolkotabi, »Theoretical Insights Into Multiclass Classification: A High-dimensional Asymptotic View«, v *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2020, str. 8907–8920. Pridobljeno: 5. december 2024. [Na spletu]. Dostopno na: https://proceedings.nips.cc/paper_files/paper/2020/hash/6547884cea64550284728eb26b0947ef-Abstract.html

▪

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

▪

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.

▪

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«.

Artificial Intelligence and Management of Dualities in Software Development Companies

Maksim Nikitashin
University of Primorska, Faculty of Management, Izolska vrata 2, 6000 Koper, Slovenia
max.nikitashin@gmail.com

Abstract

Artificial intelligence (AI) has become omnipresent in the software development area and its management nowadays, including a very important part of the management of dualities. However, this theme still remains poorly studied and debatable, especially for leading personnel of software development companies. Effective management of dualities is very important for the success of any company and artificial intelligence can significantly influence it. For this reason, the first initial goal of the research was to study different aspects of the utilization of AI in the management of dualities in companies from the area of development of software solutions to systemize existing knowledge on the topic. Those aspects include possibilities of use, its benefits and drawbacks, and analysis of existing solutions and case studies. The second less important initial goal was to prepare recommendations for the implementation and utilization of AI in the business area. The study is qualitative and consists of two sequential parts: theoretical and practical. The first one is conducted with a scoping literature review. The second one is based on the first and represents research in the professional area in the form of a structured interview. The results show the benefits as well as the importance and necessity of the implementation of AI support for the management of dualities for the success of software development companies. At the same time, they identify some potential drawbacks and problems with technology acceptance. Additionally, they identify potential inconsistencies between theoretical investigations and the reality of the business area. Based on them the most important recommendations for the process are prepared and written down. The results are useful for the people employed in the area both in leading and non-leading roles. The inconsistencies identified need an additional investigation. This work could be used as a good starting point for continuing research works on the topic.

Keywords: agile methodologies, artificial intelligence, decision-support systems, management of dualities, software development, software engineering.

Umetna inteligenca in management dualnosti v podjetjih za razvoj programske opreme

Izvleček

Umetna inteligenca (UI) je danes postala vseprisotna na področju razvoja informacijskih rešitev in njihovega upravljanja, vključno z zelo pomembnim delom managementa dualnosti. Vendar ta tema še vedno ostaja slabo raziskana in sporna, zlasti za vodilne kadre podjetij za razvoj programske opreme. Učinkovito upravljanje dualnosti je zelo pomembno za uspeh vsakega podjetja in umetna inteligenca lahko nanj pomembno vpliva. Zato je bil prvi začetni cilj raziskave preučiti različne vidike uporabe umetne inteligence v managementu dualnosti v podjetjih s področja razvoja informacijskih rešitev za sistemizacijo obstoječega znanja o temi. Ti vidiki vključujejo možnosti uporabe, njene prednosti in slabosti ter analizo obstoječih rešitev in študije primerov. Drugi manj pomemben začetni cilj je bil pripraviti priporočila za implementacijo in uporabo umetne inteligence na področju. Študija je kvalitativne narave in je sestavljena iz dveh zaporednih delov: teoretičnega in praktičnega. Prvi je izveden s pregledom literature. Drugi temelji na prvem in predstavlja raziskavo na strokovnem področju v obliki strukturiranega intervjuja. Rezultati kažejo na koristi ter pomen in nujnost implementacije podpore UI za upravljanje dualnosti za uspeh podjetij za razvoj programske opreme. Hkrati pa identificirajo nekatere morebitne po-

manjkljivosti in težave pri sprejemanju tehnologije. Poleg tega ugotavljajo morebitna neskladja med teoretičnimi raziskavami in realnostjo poslovnega področja. Na njihovi podlagi so pripravljene in zapisane najpomembnejša priporočila za proces. Rezultati so koristni za ljudi, zaposlene na področju, tako na vodilnih kot strokovnih vlogah. Ugotovljene nedoslednosti pa zahtevajo dodatno preiskavo. Rezultate dela lahko služijo kot dobro izhodišče za nadaljevalne raziskave na temo.

Ključne besede: agilne metodologije, management dualnosti, programsko inženirstvo, razvoj programske opreme, sistemi za podporo odločanju, umetna inteligenca.

1 INTRODUCTION

Artificial intelligence (AI) has been developing rapidly in recent decades and has become the most significant trend in all professional areas in the last few years [1]. The sphere of development of software was not an exception. Artificial intelligence has found a wide range of applications there, from the development process by itself to the management of projects. Almost anywhere it brought a lot of improvements [2], [3], [4], [5].

Management of dualities is an important part of the entire management process dealing with paradoxes or dualities. The latter means a situation of dealing with some pole opinions about some decisions appearing in the organization. Management of dualities by itself is intended to study and provide a guideline about dealing with such situations. Its effectiveness plays a big role in all steps of the development process, since paradoxes and poles may appear in any of them, from design till maintenance. Particularly productive collaboration, delivery, and quality assurance can be ensured only this way. However, it may be hard to implement, especially in the case of big, complex projects [6].

Management of dualities is closely related to decision-making. Supportive systems intended to help with decision-making can be utilized in the management of dualities as well. These systems nowadays are deeply integrated with artificial intelligence and machine learning [7], [8].

Despite the benefits of utilization of artificial intelligence in decision-making and development of software as well as the improvements it brings, this theme still remains a controversial topic, especially for non-technical employees, including those who make important decisions and direct organizations [9]. In terms of the utilization of artificial intelligence in software development companies, previously described thematizes of the management of dualities and of use of AI in order to support it are very important. Management of dualities is only a small part of the entire process of development of software, however as it was described before, this part plays a big role. This part can be pretty much representative for the analysis of the entire management process impact [6] and as a good example of helping companies decide whether to implement the supportive systems or not.

For the reasons described above, the author has decided to conduct comprehensive research on the theme. There were two goals set. The first goal of the research was to study different aspects of the utilization of AI in the management of dualities in software development companies: possibilities of use, its benefits and drawbacks, and analysis of existing case studies. The second goal was to prepare recommendations for the implementation and utilization based on the study of the aspects.

The research was designed as a sequential qualitative one, consisting of two parts: theoretical literature analysis and research of the real professional sphere. During the first one, the sources related to these thematic were identified and analyzed, and by the end, the results were written down. The study included both studies of scientific sources related to AI and the management of dualities and analyses of the professional sources and case studies. In the second part, a set of structured interviews with professionals employed in the field were conducted. Then their conclusions were also written down and compared with the results of the literature review to cross-validate them based on complementary/non-complementary and identify similarities, differences, and possible deficiencies.

According to the sequential design principle, one of the goals of the entire research and the main purpose of the first part was to analyze the topic in order to prepare the ground for the second part. Another less important optional goal was to study the theme by itself and systemize all the found knowledge

about it in order to help employees of software development organizations in different steps of the integration of AI into the process of the management of dualities. Another goal of the entire research and the main goal of the second part was to analyze the situation in the real business area to provide a ground for comparing the results of the interviews with the findings of the first part, evaluate those findings, and identify their possible weak points as well as required additions and corrections.

2 METHODOLOGY

The entire research was designed as a sequential qualitative one. First, the theoretical study of the topic in the form of a literature review was planned to be conducted. Then, real professional area research, based on the study of theory, should have been done.

2.1 Literature Review

As a methodology for research of the theory, the scoping literature review was utilized because of the novelty of the topic and in order to ensure a complete and comprehensive overview of the theme. Its goal is to find, map, analyze, and summarize as many sources on the topic of study as possible to make sure that no existing knowledge is missed and to identify key concepts and gaps. The most important steps here are the following:

- Definition of the research question
- Definition of the query for search
- Search for sources
- Selection/filtering of the relevant sources
- Extraction and organization of data
- Analysis and summarization of data
- Synthesis and presentation of the results [10], [11]

The research questions (RQs) defined by the author were as follows:

- RQ1. »What is the management of dualities?«
- RQ2. »How does artificial intelligence influence the management of dualities in general?«
- RQ3. »What are specifics of the management of dualities in software development companies?«
- RQ4. »How does AI influence the management of dualities specifically in the companies from the sector of development of software?«
- RQ5. »How can AI be implemented and utilized in the management of dualities specifically in software development companies?«.

Considering these questions and the fact that nowadays a large number of software development companies are utilizing agile methodologies [9], the keywords selected for the search query were the following:

- Agile methodologies
- Artificial intelligence
- Decision-making support
- Management of dualities
- Software development
- Software engineering

The keywords were combined and entered in pairs. The sources were searched in the Scopus database as one of the most recognized cross-discipline ones. Additionally, some identified relevant sub-sources of the found literature were analyzed and included as well.

Based on the research questions, the keywords, and relevance requirements the four criteria for the inclusion/exclusion were defined (Table 1).

Table 1: **The Criteria for the Literature Review**

Criteria	Description	Objectives
Actuality	The work was published no more than 5 years before the moment of the search. More recent items are prioritized.	Ensure only valid knowledge is included in the study.
Reliability	The work is published in a peer-reviewed resource, according to the information from the official web page of the resource.	Ensure the results of the analyzed work are reliable and officially confirmed.
Formal relevance	Keywords of the work include a combination of the query ones. Found items with a bigger number of matching keywords are prioritized. At least one match must be present.	Ensure the work is formally relevant. Fasten and automatize filtering of non-relevant items without the necessity to study them in depth.
Actual relevance	The work studies one of the topics specified by one of the research questions or its relationship with agile methodologies.	Ensure the work is actually relevant to the intended analysis.

For data analysis and synthesis, qualitative and quantitative approaches are available. The author selected the first one because of the width of the topic and relatively general search criteria.

After conducting the search, the author has identified and filtered according to the prioritization principles mentioned, twenty-seven of the most relevant and suitable for the intended study research items. As planned, the search was made directly in the Scopus database and in sub-sources of the literature found. The number is relatively small due to the specificness and novelty of the theme as well as the fact that a lot of items repeat each other or reference the same research works. These were the reasons for the fifth additional uniqueness criteria application and for the mentioned filtering and inclusion of sub-sources.

Also, conducting a search for the works related to the available specific tools for decision-making support enhanced with AI, the exception in the selected methodology was made. The reason was that the author was unable to find sufficient scientific sources matching the criteria for this exact topic. Here the search was made on the web with evaluation based on the trustworthiness, wide public recognition, and reliability of the identified sources. In this part, three additional web sources were identified and included in the study.

2.2 Structured Interview

For research of the real business world, the structured interview methodology was selected. It is characterized by consistency, standardization, reliability, and overall efficiency due to minimized bias and much easier data analysis compared to other kinds of interviews. At the same time, it requires good planning and preparation. Using this approach the researcher prepares a set of predefined questions which are standardized and asked always in the same order. This set is based on research questions and objectives. The recommended number of interviews is approximately 15 [12].

The research questions for this part were the same as the third (RQ3), the fourth (RQ4), and the fifth (RQ5) ones for the study of theory. The questions for the interviews were planned to be defined after the completion of the scoping literature review in order to ensure their relevance, accuracy, and usefulness.

The interviews were conducted with the professionals employed in management roles in different companies in the sector of development of software from different Eastern European countries. The interviewees were selected and invited randomly. The companies were distributed equally by size, 5 small, 5 medium-sized, and 5 large ones. These companies utilize agile methodologies for the management of projects. All the responses from the respondents were collected anonymously which was explained before the start of each interview alongside its purpose and the goals of the research.

Analysis of the responses was conducted with the standard for structured interview coding process: initial then focused and by the end theoretical coding. For the initial step descriptive approach was selected since the responses sometimes differed a lot by length and depth. For the focused coding, two main criteria were selected: relevance and frequency. The purposes were to ensure firstly actuality and secondly objectivity through using a quantitative criterion. For the theoretical step, the relevance criterion was selected as the main, since other typical criteria are less suitable for comparison and evaluation of the results of the investigation of theory, which was the predominant goal of the second part of the entire research [12].

3 RESULTS

3.1 Literature Review

After conducting the literature review utilizing the scoping literature review methodology described in the Methodology section, the following results were obtained.

3.1.1 Decision-Support

Decision support and related branches of science play an important role in any sphere of modern economics. It can influence management on all levels and stages from procurement and supply [13], [14] to after-sales services [8] or what is much closer to the theme of this paper in security management of information systems [15]. Decision-making support studies different aspects of the process, for instance, decision-making styles (DMS) [16], [17] dealing with managers' ways of thinking during the process or building of decision models and trees intended to help with the process [15].

3.1.2 Artificial Intelligence

Artificial intelligence started to play an important role in nearly all spheres of modern economics in recent years as well. The range of areas varies from the most practically oriented ones like production [18] and logistics [19] to insurance [20], law [21], and sport [22]. Of course, management and decision-making support are among them too. Artificial intelligence and data mining have boosted the second sphere with the possibility to collect and analyze enormous amounts of data and thus have become a necessary part of any successful modern business [7].

3.1.3 Management of Dualities

Dualities or paradoxes in management mean situations when there are two opposite opinions or points of view about something. Management of dualities is a branch of management dealing with all kinds and stages of such situations. It plays an important role in the success of any company [6]. Management of dualities as a branch of management science provides guidelines on how to identify, understand, and exploit arising paradoxes in an organization [23].

All dualities can be split into two main categories: normative and strategic. Both are related to the corresponding kinds of management. The normative one deals with an organization's mission, values, and norms. Examples of paradoxes arising here are »profitability vs. responsibility« and »exploitation vs. exploration.« The strategic one deals with the organization of the company's resources in order to achieve its objectives and goals. Examples of dualities there are »cost vs. differentiation« and »insourcing vs. outsourcing« [6].

Management of paradoxes is closely related to decision-making support. The latter helps to improve dealing with dualities [7], [8]. Consequently, the integration of artificial intelligence also started there and influenced this sphere too.

3.1.4 Management in Software Development Companies

Management of modern software development companies has a number of specifics. The most important among them is the utilization of agile principles and methodologies. Those principles were established in the year 2001, and since then, the approaches for the management of companies and projects implementing them have become the most popular in the sphere of the development of software. In general, they are based on adaptive iterative development with continuous improvements [24]. One of the main specifics of agile management is the high level of decentralization. Development teams are allowed to make and implement some decisions at a low level without confirmation from the higher management [25], [26].

Dualities are present in such methodologies as well. Primarily they are of a strategic nature. The most popular among them is »trust vs. control.« Another widespread one is »insourcing vs. outsourcing« [27], [28].

3.1.5 Artificial Intelligence and Decision Support in Management of Dualities

As well as anywhere nowadays, artificial intelligence has found its utilization in the management of software development companies too. This brings both opportunities and challenges. The first includes routine task automation as well as improved planning, risk management, and decision-making. Examples of the second are staff training necessity, overreliance on AI support, and data security issues [8], [29].

Artificial intelligence nowadays is widely utilized in decision support. There are specific kinds of tools called artificial intelligence decision-support systems (AI-DSS). They are utilized in different areas from public management to healthcare, and of course, the business management sphere is among them too [30].

Some researchers have studied the application of AI-DSS particularly in relation to management of dualities and agile management. Unfortunately, there was no work found directly related to the most popular paradoxes present in agile management mentioned before. The closest to the theme of this paper is research from Manzur et al. [31]. These authors have identified that there are realistic possibilities for the application of artificial intelligence into agile management and management of dualities with opportunities and challenges similar to those described before. Those possibilities are design thinking and sprint preparation integration as well as violative conditions management. The latter includes dealing with arising dualities.

Decision-support systems have found their utilization in risk management of different technical spheres like power engineering [32], infrastructure [33], cybersecurity [34], [35], and, as mentioned before, in management [8], [36] including its kind of

development of software. The authors of the last two cited papers identify some improvements in relation to both IT and the company's general efficiency. For instance, in terms of speed of processing of data, quality of decisions, and reduction of primary risks. At the same time, some drawbacks and problems are pointed out too. Examples are issues with the quality and protection of data and the necessity of regular checks of the outputs. Another work by Khamitov [37] has identified potential problems in the utilization of agile management caused by non-rational human decision-making not supported by some technical systems.

It is important to mention that acceptance and utilization of AI-supported tools, including decision--support systems, is still problematic. Business and management-related employees still have a tendency to avoid the usage of such tools and show much more concern rather than will, even in software development companies [9]. This moment is especially important because management of dualities and decision-making primarily fail in their sphere of responsibilities.

3.1.6 Case Studies

An important aspect of analysis of the topic of the research is to study not only theory by the practical examples as well. They can provide enhanced and broader overview as well as help to evaluate the theoretical investigations. At the moment, there exist a number of specific solutions from different providers. Most of them are intended for some specific industry. Some well-known providers of such artificial intelligence-supported systems are SAP [38], Wolfram [39], and SalesForce [40].

In the real business world, artificial intelligence decision support systems are utilized in different ways. Nowadays they are able to collect data from different sources, from open kinds like social networks [41] to organization's internal ones, like human resource management systems [42].

A case study of one Italian company specializing in staff training has identified that artificial intelligence decision support systems by themselves are not comprehensive stand-alone solutions. At the same time, they can be effectively utilized to enhance certain tasks and activities conducted by employees [42].

Due to the small number of sufficient case studies found, the work conducted by the research group from the Slovenian universities was included in the analysis. The paper based on its results has already been submitted for publication in the Scopus-indexed journal but is still passing the review process. This research on the business world, much more directly related to the theme of this paper, was conducted specifically on software development companies in Slovenia. It has been identified that employees of these organizations do not utilize some special decision-support tools, but rather chatbots intended for information search like ChatGPT as a substitute for such kind of systems while making certain decisions [9].

3.2 Structured Interview

In this part preparation, conduction, and the results of the research of the real professional area conducted with the structured interviews are presented. The results of the scoping literature review are already described in the second section of the paper. In accordance with the sequential design of the entire research, the second part was prepared and started only after the completion of the first one.

According to the selected methodology and based on the results of the literature research the author has prepared the questions for the interviews. These questions are presented in Table 2 below, together with their objectives and sources they are based on.

Figure 1 shows the distribution of the interviewees by the levels of management. Distribution was not even. The top level of management was prevailing. However, other levels were represented with some significant part of the audience as well. Additionally, such distribution is adequate because according to the findings of the literature analysis, today, top--level management deals with paradoxes much more often than others.

Answers to the first interview question were different. Five of the most popular dualities mentioned included »stability vs change«, »scale vs scope«, »standardization vs mutual adjustments«, »centralization vs decentralization« and »functions vs. processes«

Responses to the second interview question were more unified. Almost all of the interviewees utilized some kind of SWOT analysis while resolving the dualities faced manually. Also, the use of some tools for data analytics was mentioned.

The breakdown of responses to the third interview question is presented in Figure 2.

The reasons mentioned in answering the fourth

Table 2: **The Questions for the Structured Interview**

Objectives	Question	Sources
Identify what kind of dualities the interviewee faces at work. Answer RQ3.	What kinds of dualism do you face in your work?	[6], [24], [25], [26], [28], [31]
Identify how the interviewee deals with the dualism faced. Answer RQ3 and RQ 4.	How do you deal with the dualism you face in your work?	[6], [27], [28], [31]
Identify if the interviewee utilizes AI support in the management of dualities at all. Answer RQ4 and RQ5.	Do you utilize any kind of AI support in the management of dualities?	[9], [31], [42]
Identify the reasons why the interviewee utilizes or does not utilize AI support in the management of dualities. Analyze the most popular among the reasons. Answer RQ4 and RQ5.	Why do you utilize/avoid utilizing AI support in the management of dualities?	[7], [9], [30], [31], [37] [42]
Identify which of the tools are the most well-known/popular. Analyze if the interviewee is conscious of the possibilities of AI utilization and if they are, to which degree. Answer RQ5.	Which AI tools supporting the management of dualities do you know/ utilize?	[9], [30], [31], [42]

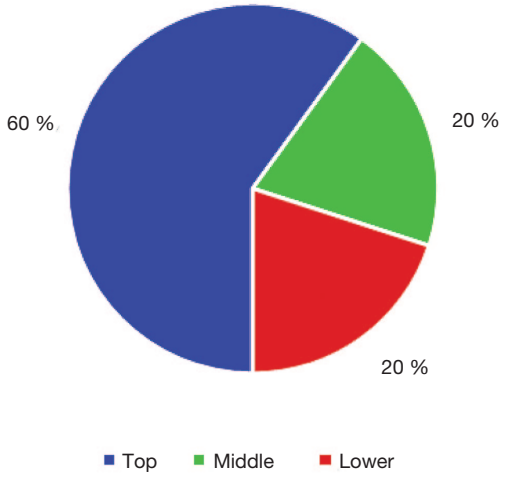


Figure 1: **Distribution of the Interviewees by the Levels of Management**

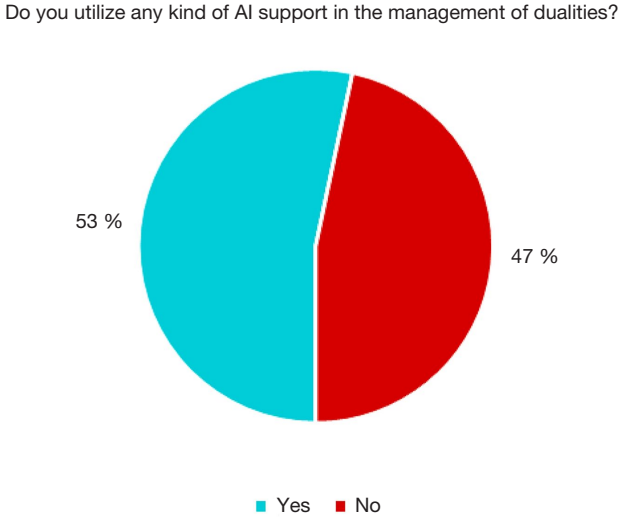


Figure 2: **Distribution of Responses to the Third Interview Question**

interview question were different both for the utilizers and for the non-utilizers. Lack of time, knowledge, good case studies, or even awareness of supportive systems existence as well as constraints of the solutions they are developing are stopping those who do not use AI tools. The utilizers have mentioned third-party opinion provision, usefulness, efficiency improvements, and its ubiquity as the main motivators for usage. At the same time, even those who utilize AI tools have pointed out some drawbacks of their utilization like data security issues, errors and imperfections in outputs as well as potential risks of overlaying onto support and loose of important professional skills. Also, some utilizers have mentioned the fact that the use of the tools is their personal initiative, and the companies they are working on do not support or even try to limit the utilization.

The answers to the last question were pretty much unified. The most of non-utilizers were unable to name any kind of specific tools they knew. Among the utilizers, almost all have mentioned the use of some kind of tools for information search or chatbots like ChatGPT or Microsoft Copilot (Bing AI). At the same time, only a small number are utilizing some specific systems, for example, Oracle Analytic or some custom-developed ones.

4 DISCUSSION

The research of literature, of course, has some constraints. One of them is the limited number of sources analyzed. Another is the fact that the search was conducted only in one database, nevertheless it is very large and well-recognized. Also, there was one minor exception in the selected methodology during the study of the available tools. However, this part is still structurally and pretty much comprehensively done. By the end, it can be indicated as complete and its results as useful. It achieved both its goals set at the start providing the basis for the interviews as well as a pretty much comprehensive overview of the topic of the research.

Similarly, research in the professional sphere has some limitations. First of all, in terms of geographical region. Second, the width of the professional sphere of software development is huge, and there could be differences in some specific subareas, while the research included all types of organizations and solutions without differentiation. At the same time, this part was designed and almost totally conducted according to the most important and recognized recommendations and equally represents companies of different sizes. For these reasons, it can be stated as comprehensive and useful as well. As well as the first part, it has achieved its goals.

As a result of the literature review, addressing the first research question (RQ1) it can be stated that the management of dualities is an important part of the management of any modern organization dealing with so-called paradoxes arising in the management of the organization. It can be enhanced with closely related decision-support concepts and systems implementing them. The latter can help in dealing with different types of dualities popular in software development, like outsourcing and risk-related ones.

Answering the second research question (RQ2), artificial intelligence has found a wide utilization in decision-supportive systems for a number of reasons. According to almost all researchers, it makes them much more efficient, useful, and faster. At the same time, some authors point out drawbacks related to staff's technology acceptance and training as well as to data utilization and security.

Addressing the third research question (RQ3), the management of software development companies, including its dualities-related part, has plenty of specifics, like agile principles utilization, some particular kinds of dualisms popular, and low-level decision-making. However, addressing the fourth research question (RQ4), it can be identified that artificial intelligence decision-support systems can be utilized there too. Most of the researchers point out the presence of both opportunities and threats described in the previous paragraph. But overall, those systems can bring a number of benefits improving management of dualities in terms of speed and quality. At the same time, addressing the fifth research question (RQ5) they must be utilized wisely considering previously mentioned data-related issues as well as potential problems with the quality of output and resistance from the business and management employees inside the organization. Here investments and provision of enhanced security of data, regular checks of results of work of the supportive system as well as argued, persuasive presentation of necessity and potential benefits of the implementation are recommended.

Also, it is important to take into consideration previously mentioned specifics of management of modern software development projects, especially already mentioned low-level decision-making typical for agile methodologies. Taking them into account there can be recommended implementation of decision-making supportive systems in the operational management of development teams similar to those at highest levels of the organization.

The results of the research in the business area are primarily consistent with the literature. The benefits and disadvantages of utilization of AI support in the management of dualities are similar in both parts. The same reasons stop organizations and their employees from implementing and active use of artificial intelligence-enhanced systems are mentioned both in the literature and by the interviewees. At the same time, in terms of the RQ5, these professionals have reported some additional causes, like constraints on the solutions they are working on and a lack of awareness of the existence of the supportive system. Also, there are differences in terms of the dualities faced in agile management (RQ3) and of the utilization of the specific tools (RQ5). Literature and web sources primarily mention some systems specifically intended for decision support and data analytics. However, most of the professionals interviewed utilize some substitutions like chatbots. Only a very small number of them use previously mentioned specific systems.

Of course, the differences may be caused by regional or some other limitations. However, there could be another reason, like inconsistency between the theoretical studies and the reality of the business sphere. Here the author can propose additional practical investigations both in general and considering potential regional specifics.

5 CONCLUSION

Addressing the research questions of both parts of the entire work and as a general conclusion, the author of this paper can identify that management of dualities is a part of the entire management process dealing with the arising dualities/paradoxes of normative and strategical nature. It can be stated that this part and its efficiency are very important for the success of any company nowadays, including those in the software development sector. Management of dualities can enhance such aspects as development by itself or maintenance and improvement of a company's reputation. For this reason, both artificial intelligence-enhanced and classical decision-support systems significantly improving this process should be implemented and actively utilized in the sphere. Of course, the last two activities must be conducted wisely, considering all the risks and drawbacks. Also, specifics of the area and its management play a significant role and must be taken into account.

Decisions on the implementation and its details are always up to the organization, and structured, proven reasoning and argumentation whether to decide upon it or not is very important. However still, if a company from the software development sector would like to be successful today those enhancements need to be made at some point in time. This way the organization can ensure its competitiveness and efficiency in the modern business market.

Based on the research of the professional area of development of software conducted by the author there were identified some differences between the previous investigations analyzed and the results of the practical research. For instance, in terms of reasons stopping the implementation and utilization as well as regarding the tools used. As was mentioned before, there could be recommended additional investigation into the theme. This paper and its results can be used as a good starting point for continuing research works on the topic.

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] Admin, »Worldmetrics - Turning Numbers into Insights,« Worldmetrics. Accessed Jul. 15, 2024. [Online]. Available: https://worldmetrics.org/ai-in-software-development-statistics/.

[3] »AI | 2024 Stack Overflow Developer Survey.« Accessed Aug. 17, 2024. [Online]. Available: https://survey.stackoverflow.co/2024/ai#1-ai-tools-in-the-development-process.

[4] K. Haan, »22 Top AI statistics and Trends in 2025,« Forbes Advisor. Accessed Jul. 15, 2024. [Online]. Available: https://www.forbes.com/advisor/business/ai-statistics.

[5] I. Shani, »Survey reveals AI's impact on the developer experience - The GitHub Blog,« The GitHub Blog. Accessed Jul. 15., 2024. [Online]. Available: https://github.blog/news-insights/research/survey-reveals-ais-impact-on-the-developer-experience/.

[6] R. Biloslavo, C. Bagnoli, and R. R. Figelj, »Managing dualities for efficiency and effectiveness of organisations,« *Industrial Management & Data Systems*, vol. 113, no. 3, pp. 423–442, Mar. 2013, doi: 10.1108/02635571311312695.

[7] B. Asgarova, E. Jafarov, N. Babayev, A. Ahmadzada, V. Abdullayev, and T. Triwiyanto, »Development process of decision support systems using data mining technology,« *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 36, no. 1, p. 703, Jul. 2024, doi: 10.11591/ijeecs.v36.i1.pp703-714.

[8] H. Mydyti, A. Kadriu, and M. P. Bach, »Using data mining to improve Decision-Making: Case study of a recommendation system development,« *Organizacija*, vol. 56, no. 2, pp. 138–154, May 2023, doi: 10.2478/orga-2023-0010.

[9] M. Nikitashin and B. Werber, »Artificial Intelligence and Software Development in Slovenia: Adoption, Challenges, and Opportunities«, submitted for publication.

[10] C. Hart, *Doing a literature review: releasing the social science research imagination*. 1998. [Online]. Available: http://ci.nii.ac.jp/ncid/BB14383612

[11] Z. G. Ma, *Scoping Review in One Week: A cookbook for beginners to conduct scoping reviews and write review papers with simple steps*. 2022. [Online]. Available: https://www.amazon.com/-/es/Zheng-Grace-Ma-ebook/dp/B09SGM5SSY

[12] S. Kvale, *InterViews: An introduction to qualitative research interviewing*., vol. 3, no. 20. SAGE Publications, 1996, pp. 287–288.

[13] P. Buljat, L. Zekanović-Korona, and J. Grzunov, »The Role of Visual Communication in Consumer Decision-Making in a Digital Environment,« in *2024 47th MIPRO ICT and Electronics Convention (MIPRO),* 2024, pp. 939–944. doi: 10.1109/mipro60963.2024.10569841.

[14] N. Grishchenko, »E-Commerce Cross-Border and Domestic Dynamics: decision tree and Spatial Insights on Seller Origin impact,« *Businesses*, vol. 4, no. 3, pp. 270–298, Jul. 2024, doi: 10.3390/businesses4030018.

[15] L. Jelovčan, A. Mihelič, and K. Prislan, »Outsource or not? An AHP Based Decision Model for Information Security Management,« *Organizacija*, vol. 55, no. 2, pp. 142–159, May 2022, doi: 10.2478/orga-2022-0010.

[16] J. A. Andersen, »EXPLAINING ORGANIZATIONAL EFFECTIVENESS—LEADERSHIP STYLES VS. MOTIVATION PROFILES VS. DECISION-MAKING STYLES: SUPPORTING OR COMPETING DIMENSIONS?,« *Dynamic Relationships Management Journal*, vol. 11, no. 1, May 2022, doi: 10.17708/drmj.2022.v11n01a03.

[17] N. Kadoić, M. G. Marković, and T. Jagačić, »Introducing the intensity of influence in Decision-Making style analysis,« *Organizacija*, vol. 57, no. 3, pp. 287–302, Aug. 2024, doi: 10.2478/orga-2024-0021.

[18] M. Roblek, V. Radisevljević, and A. Brezavšček, »Merjenje učinka uporabe strojnega učenja pri mikroplaniranju proizvodnje,« in *OTS 2024* vol. 61. 2024, pp. 103–114. doi: 10.18690/um.feri.4.2024.9.

[19] S. Vojvodić, K. Vidović, D. Leljak, and A. Najev, »Conceptual Design of IT Solution for Multimodality and Traffic Flow Optimization in Catchment Area of the Airport: Case Study of Zagreb Airport,« in *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, 2024, pp. 1030–1034. doi: 10.1109/mipro60963.2024.10569931.

[20] A. Ć. Hasibović and M. Hasić, »The Need for Transforming Business Models in Insurance Using AI – Case Study,« in *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, 2024, pp. 957–961. doi: 10.1109/mipro60963.2024.10569701.

[21] P. Karanikić, »Artificial Intelligence in the Digital Economy: Intellectual Property Protection Challenges,« in *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, 2024, pp. 1000–1005. doi: 10.1109/mipro60963.2024.10569286.

[22] A. Nikić, A. Topalović, and M. P. Bach, »From Data to Decision: Machine Learning in Football Team Management,« in *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, 2024, pp. 1059–1064. doi: 10.1109/mipro60963.2024.10569835.

[23] F. Graetz and A. C. T. Smith, »The role of dualities in arbitrating continuity and change in forms of organizing,« *International Journal of Management Reviews*, vol. 10, no. 3, pp. 265–280, Aug. 2008, doi: 10.1111/j.1468-2370.2007.00222.x.

[24] S. Chahal, »Agile methodologies for improved product management,« *Journal of Business and Strategic Management*, vol. 8, no. 4, pp. 79–94, Sep. 2023, doi: 10.47941/jbsm.1439.

[25] O. E. Ajewumi, »The influence of agile organizational design on employee engagement and performance in the digital age,« *International Journal of Research Publication and Reviews*, vol. 5, no. 10, pp. 25–39, Oct. 2024, doi: 10.55248/gengpi.5.1024.2702.

[26] N. J. Chukwunweike and N. O. E. Aro, »Implementing agile management practices in the era of digital transformation,« *World Journal of Advanced Research and Reviews*, vol. 24, no. 1, pp. 2223–2242, Oct. 2024, doi: 10.30574/wjarr.2024.24.1.3253.

[27] J. Langerman and W. S. Leung, »The effect of outsourcing and insourcing on Agile and DevOps,« *Journal of Information Technology Teaching Cases*, p. 204388692311768, May 2023, doi: 10.1177/20438869231176841.

[28] L. Mathiassen, J. Persson, and L. P. Heje, »Outsourcing Agile Projects with Time-and-Material Contracts: A Trust-Control Duality Perspective,« *International Journal of Business Information Systems*, vol. 1, no. 1, p. 1, Jan. 2022, doi: 10.1504/ijbis.2022.10048139.

[29] Z. Jin, »Integrating AI into Agile Workflows: Opportunities and Challenges,« *Applied and Computational Engineering*, vol. 100, no. 1, pp. 49–54, Nov. 2024, doi: 10.54254/2755-2721/100/20251754.

[30] A. Kovari, »AI for Decision Support: Balancing Accuracy, Transparency, and Trust Across Sectors,« *Information*, vol. 15, no. 11, p. 725, Nov. 2024, doi: 10.3390/info15110725.

[31] B. N. Z. Manzur, F. a. E. Bazán, P. Novoa-Hernández, and C. C. Corona, »In what ways do AI techniques propel decision-making amidst volatility? Annotated bibliography perspectives,« *Journal of Innovation and Entrepreneurship*, vol. 13, no. 1, Aug. 2024, doi: 10.1186/s13731-024-00396-2.

[32] K. Stødle, R. Flage, and S. D. Guikema, »Identifying power outage hotspots to support risk management planning,« *Risk Analysis*, Oct. 2024, doi: 10.1111/risa.17663.

[33] L. Bevanda and A. Cerić, »Decision Support System for risk management in infrastructure projects,« in *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, pp. 45–52, Sep. 2024, doi: 10.5592/co/phdsym.2024.5.

[34] M. V. Devarajan, M. Al-Farouni, R. Srikanteswara, R. R. V. S. S. Bharattej, and P. M. Kumar, »Decision Support Method and Risk Analysis Based on Merged-Cyber Security Risk Management,« *in 2024 Second International Conference on Data Science and Information System (ICDSIS)*, 2024, pp. 1–4. doi: 10.1109/icdsis61070.2024.10594070.

[35] A. Khovrat and V. Kobziev, »Algorithms for decision support in risk management in the development of information-sensitive socially oriented systems,« in *Press of the Kharkiv National University of Radioelectronics eBooks*, 2024. doi: 10.30837/mmp.2024.062.

[36] L. Osuszek and J. Ledzianowski, »Decision support and risk management in business context,« *Journal of Decision System*, vol. 29, no. sup1, pp. 413–424, Jul. 2020, doi: 10.1080/12460125.2020.1780781.

[37] D. A. Khamitov, »The impact of cognitive distortions on decision making in agile project management frameworks: Current positions and perspectives in the academic community,« *Management Sciences*, vol. 14, no. 2, pp. 104–115, Aug. 2024, doi: 10.26794/2304-022x-2024-14-2-104-115.

[38] »SAP BusinessObjects | Business Intelligence (BI) Platform & Suite,« SAP. Accessed Nov. 13, 2024. [Online]. Available: https://www.sap.com/products/technology-platform/bi-platform.html.

[39] »Wolfram Mathematica,« Wolfram. Accessed Nov. 13, 2024. [Online]. Available: https://www.wolfram.com/mathematica/.

[40] Salesforce, »SalesForce announces CRM Analytics, AI-Based insights for sales, marketing, and service teams in every industry,« Salesforce. Accessed Nov. 13, 2024. [Online]. Available: https://www.salesforce.com/news/stories/crm-analytics-2/.

[41] M. Freire, F. Antunes, and J. P. Costa, »Social Web Analysis for Decision Support: a case study,« in *Frontiers in artificial intelligence and applications*, 2023. doi: 10.3233/faia230770.

[42] S. Marocco, A. Talamo, and F. Quintiliani, »Applying design thinking to develop AI-Based Multi-Actor Decision-Support Systems: A case study on Human capital investments,« *Applied Sciences*, vol. 14, no. 13, p. 5613, Jun. 2024, doi: 10.3390/app14135613.

▪

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.

Tehnični vidik izboljšave prostorske identitete Slovenije v letalskih simulatorjih

Leon Alessio, dr. Miha Janež
Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, 1000 Ljubljana
leonalessio1@gmail.com, miha.janez@fri.uni-lj.si

Izvleček

Virtualna različica letanja je popularna alternativna osvajanju neba, ki jo uporabljajo tako amaterski navdušenci za zabavo kot tudi profesionalci in piloti za usposabljanje ter izboljšanje svojih sposobnosti. Za boljšo izkušnjo in učinkovito usposabljanje je ključnega pomena pokrajina, v kateri uporabnik znotraj simulatorja leti, žal pa je podoba Slovenije v simulatorjih klavrna in pomanjkljiva. Zato se je podjetje AformX, ki med drugim tudi usposablja bodoči letalski kader, skupaj z dvema članicama Univerze v Ljubljani podalo v projekt Prostorska identitete Slovenije v letalskih simulatorjih. Cilj je bil izdelava pokrajinskega dodatka za letalske simulatorje, ki dobro predstavlja Spodnjesavinjsko dolino ter Celjsko kotlino in tako omogoča potopitveno izkušnjo letenja v tej pokrajini znotraj simulatorja. Za dosego zadanega cilja smo v ekipi z raznolikimi predznanji pod mentorstvom profesorjev raziskali značilnosti te pokrajine in specifikacije delovanja simulatorja. S pomočjo odprtokodnih podatkov in programov, katere je bilo za naše potrebe potrebno predelati, smo nato razvili virtualno okolje za letalski simulator X-Plane, ki izboljša uporabniško izkušnjo in učinkovitost usposabljanja.

Ključne besede: Letenje, Letalski simulatorji, Spodnjesavinjska dolina, Celjska kotlina, Virtualno okolje

Technical aspect of improving slovenia's spatial identity in flight simulators

Abstract

Virtual flight has consistently held a prominent position as a viable alternative to traditional physical flying, with the backing of a dedicated and vocal community. It has also become an indispensable tool within the professional training process of prospective pilots. In order to achieve the most effective outcomes, the virtual environment must be crafted with a high degree of realism, fidelity, and interactivity. AformX, a company operating a flight academy, in collaboration with the University of Ljubljana, initiated a project with the objective of enhancing Slovenia's spatial identity in flight simulators. The project involved the creation of a virtual landscape representing the region of Lower Savinja Valley and Celje Basin, in which the AformX flight base is situated. To achieve our goal, we undertook extensive research into the characteristics of the region and the simulator itself. We then used modified OpenStreetMap and GURS spatial data together with a refined open-source program, OSM2XP, to create a virtual environment. This exceeded our expectations and fulfilled our goal. The project could serve as a launch pad for a bigger project spanning the whole country of Slovenia or even more.

Keywords: Flight, Flight simulators, Lower Savinjska valley , Celje basin, Virtual environment

1 UVOD

Sposobnost letenja je bila od nekdaj za človeško vrsto fascinantna. Tudi danes, ko nam je zračni transport omogočen s pomočjo zrakoplovov (letal, helikopterjev itd.) ima panoga letalstva številčno in predvsem strastno skupnost. Ker pa je s fizičnim letenjem povezano veliko spremenljivk od vremena do razpoložljivosti letal, pristajalnih stez in stroškov, je precej popularna virtualna različica letenja, pri kateri se s pomočjo letalskih simulatorjev simulira fizično le-

tenje. Uporablja se za zabavo, razvoj zračnih plovil in tudi za usposabljanje bodočega letalskega osebja. Omogoča približek letenju za veliko manjša finančna in fizična sredstva ter sposobnost nadzora okolja po uporabnikovih željah. Za doseganje izkušnje, čim bolj podobne fizičnemu letenju, je poleg drugih dejavnikov zelo pomembna virtualna pokrajina oziroma virtualno okolje, v katerem uporabnik leti znotraj letalskega simulatorja. Bolj kot je ta podobna pravi pokrajini, lažje se bo virtualni pilot vživel v letenje, kar poleg izboljšane izkušnje tudi pripomore pri izboljšanju uporabnikove prostorske orientacije. To je še posebej pomembno pri usposabljanju bodočih pilotov, saj jim bo tako lažje uporabiti na simulatorjih pridobljeno znanje, kadar bodo fizično leteli v pravi različice te pokrajine. Na žalost pa je podoba Slovenije v letalskih simulatorjih precej pomanjkljiva in neprepričljiva, kar onemogoča usposabljanje bodočega osebja na najvišji ravni. Zato sta se v sodelovanju podjetje AformX in Univerza v Ljubljani, bolj podrobneje Fakulteta za arhitekturo in Fakulteta za računalništvo in informatiko, lotili projekta Prostorska identiteta Sloveniji v letalskih simulatorjih [1], sofinanciranega s strani Republike Slovenije in Evropske unije. Osrednji cilj projekta je bil izdelava virtualnega pokrajinskega dodatka za letalski simulator X-Plane 12, ki bi dobro predstavljal Spodnjesavinjsko dolino in Celjsko kotlino, kjer pri AformX usposabljajo bodoči letalski kader. Hkrati je bil eden od ciljev tudi razviti in preizkusiti postopek razvoja takega dodatka. Poleg tega je bil namen projekta študentom omogočiti osvajanje novih veščin in sodelovanja z gospodarstvom.

2 METODE

Izdelavo dodatka virtualne pokrajine smo ločili na tri dele:

1. Specifični del, ki je vseboval pomembne orientacijske in prepoznavne točke, ki s svojo edinstvenostjo pripomorejo k izgledu pokrajine (na primer Celjski grad, značilna polja hmelja za to pokrajino, logistični center Lidl v Arji vasi itd.). Vsako tako točko smo ročno izdelali in vnesli v okolje.
2. Generični del, ki je vseboval večino stavb in zelenja, ki pa so zaradi številčnosti bile izdelane na drugačen način, ki je podrobneje opisan v nadaljevanju članka.
3. Ortofoto podlaga, pridobljena iz satelitskih fotografij, ki upodabljajo vse razen stavb (polja, reke itd.). Fotografije smo združili v enotno in zaključeno okolje.

Ta članek se poglobi predvsem v izdelavo generičnega okolja in združevanja treh delov v skupno celoto, saj je bilo to iz računalniškega vidika najzahtevnejše. Za ustvarjanje generičnega okolja smo uporabili bazo prostorskih podatkov OpenStreetMap (OSM) [2] in prostorske podatke Geodetske uprave Republike Slovenije (GURS) [3]. Prilagojeno verzijo, ki je obsegala izbrano pokrajino, smo poleg knjižnice generičnih objektov in zelenja uporabili kot vhod v modificirano različico odprtokodnega programa OSM2XP [4]. Ta je, po naših potrebah pretvoril to bazo prostorskih podatkov v virtualno okolje in poskrbel za povezavo med pretvorjenimi prostorskimi podatki in objekti v knjižnici.

2.1 Predelava prostorskih podatkov

OpenStreetMap je odprta baza prostorskih podatkov, ki predstavlja alternativo popularnim internetnim zemljevidom, kot je npr. GoogleMaps. Njena posebnost je v tem, da je posodobljena in vzdrževana s strani skupnosti uporabnikov. Njena licenca omogoča prosto uporabo baze za kakršnekoli namene. Podatki so najpogosteje shranjeni v .osm datotekah, ki temeljijo na XML strukturi in uporabljajo topološko podatkovno organizacijo. Osnovni gradnik teh podatkov je Node (vozlišče), ki predstavlja točko na zemljevidu in vsebuje njene koordinate ter unikaten identifikator. Več vozlišč se lahko poveže v Way (pot), ki je urejen in povezan seznam vozlišč ter predstavlja obliko oziroma poligon na zemljevidu (npr. reko, ceste, stavbe itd.). Oba opisana gradnika sta lahko vključena v Relation (relacija), ki določa povezave med vsebovanimi elementi (npr. področje omejitve hitrosti na cesti). Vsem trem omenjenim strukturam je mogoče dodati Tags (oznake). Vsaka ima ključ, ki narekuje kategorijo oznake (npr. building) in vrednost, ki narekuje vrednost (npr. school) znotraj domene ključa. Oznake dodajo poligonu (poti) smisel in namembnost [5]. OSM tudi nudi možnost izvoza določene regije zbranih podatkov, ki jih je možno z orodjem JOSM [6] urejati in nato po potrebi tudi naložiti nazaj.

Najprej smo podatke iz OSM za celotno regijo izvozili. Ugotovili smo, da lahko zaradi odprtosti urejanja kljub določenim smernicam, kako te podatke urejati, pride do nekonsistentnosti in pomanjkljivosti

podatkov, kar smo nujno morali popraviti za njihovo strojno obdelavo. Pred predelavo teh podatkov smo sestavili šifrant naših oznak, kategorij in vrednosti na podlagi potreb projekta (npr. če smo v končnem okolju želeli na tistem področju »postaviti« hišo, smo tistemu poligonu dodali oziroma spremenili vrednost oznake building na hiša1 oz. hiša3, če je le-ta v resnici imela ravno streho). Nato je sledilo ročno pregledovanje prostorskih podatkov v JOSM urejevalniku in dodajanje oziroma popravljanje oznak, kar je bilo dokaj zamudno.

Problematiko pomanjkanja podatkov o več tisoč objektih v pokrajini smo rešili s pretvorbo prostorskih podatkov Geodetske uprave Republike Slovenije (GURS). Zaradi neskladnosti različnih formatov in strukture obeh uporabljenih vrst prostorskih podatkov smo naleteli na več izzivov. Eden izmed njih je bila razlika med koordinatnima sistemoma, zaradi česar smo morali prenesene podatke iz GURS pretvoriti na koordinatni sistem WGS-84, ki ga uporablja OSM. Po združitvi na videz pravilnih pretvorjenih GURS podatkov in prekategoriziranih OSM podatkov v programu OSM2XP je pri pretvorbi teh podatkov v X-Plane okolje prišlo do nepojasnjenih napak. Po temeljitem pregledu vseh možnih razlogov za nenavadno delovanja programa smo ugotovili, da so se novim objektom pri pretvorbi določili negativen identifikatorji, kar program OSM2XP ni sprejemal in je zato prihajalo do napak med izvajanjem. Težavo smo rešili s skripto, ki je dodanim objektom spremenila identifikatorje v pozitivne številke, ki v sklopu datoteke še niso bile zasedene.

2.2 Dodelava programa OSM2XP

Z računalniškega stališča je bila ena izmed glavnih nalog dodelava odprtokodnega programa OSM2XP za potrebe projekta. Ker gre za kompleksen program z relativno pomanjkljivo dokumentacijo, je največ časa zahtevalo razumevanje delovanja programa, vendar je bilo to nujno za učinkovito dodelavo le-tega.

Prva večja sprememba je bila predelava obstoječe dokaj osnovne logike določanja višin stavb. Ob posvetu s študenti arhitekture in urbanizma smo določili razpon možnih višin objektov za vsako kategorijo, nakar smo v programu implementirali logiko izbiranja višine stavb, kadar ta ni bila eksplicitno podana, v obliki vrednosti oznake height. Ta je bila za simuliranje raznolikosti objektov izbrana naključno, znotraj intervala, določenega za vrednost oznake building trenutnega objekta.

Program je že vseboval osnovno logiko izbiranja enostavnih objektov (z dokaj enostavnimi tlorisi) glede na vrednost oznake building in zanjo izbral model iz zbirke 3D objektov, vključenih v datotečni sistem programa. Vendar to ni upoštevalo naših novih vrednosti za to oznako. Za te smo pripravili številne 3D modele objektov, ki so reprezentativno odražali tipične objekte Spodnjesavinjske doline in Celjske kotline. Zato je bilo treba logiko nalaganja in izbiranja objektov v podatkovne strukture znotraj programa dodelati. Ker smo okolje prekategorizirali ročno, smo predvidevali, da je kljub skrbnosti prišlo do napak in izpuščenih objektov. Zato smo logiko določanja objekta poligona predelali tako, da je program najprej preveril, ali je oznaka building trenutnega objekta vsebovana v seznamu naših kategorij. V primeru, da je vsebovana, je program glede na vrednost omenjene oznake in dimenzij stavbe naključno izbral med ustreznimi 3D modeli stavb. Če pa oznaka ni bila vključena v omenjen seznam, je program nadaljeval po privzetem delovanju, za katerega smo ocenili, da je za manjše število primerov primerno.

Pri generiranju objektov z zahtevnejšimi tlorisi se je program posluževal fasad - tekstur, ki jih je »nalepil« na objekt. Podoben postopek dodelave smo uporabil tudi tukaj in nadgradili osnovno logiko, da je za naše oznake izbirala »naše« dodelane fasade. Program ima še vrsto nastavitev, ki omogočajo prilagajanje izvajanja, npr. določanje največje dovoljene razlike v dimenzijah objektov in 3D modelov.

2.3 Združevanje, testiranje in izboljševanje

Datoteke, ki jih je generiral program OSM2XP, še niso bile primerne za vstavljanje v simulator X-Plane. Pred vnosom v simulator jih je bilo treba obdelati še v programu WorldEditor (WED) [7], v katerem smo lahko te datoteke predelali v okolje, primerno za uporabo v simulatorju.

X-Plane okolje je sestavljeno iz več plasti. V grobem opisano se na najnižji ravni nahaja osnovna plast (angl. Base/Mesh), ki predstavlja tla okolja. V našem dodatku scenariju smo to plast za ciljno regijo nadomestili z ortofoto podlago, ki smo jo v sklopu projekta pridobili iz satelitskih slik regije. Nad osnovno plastjo je nato lahko prikazanih več prekrivnih plasti (angl. Overlay). Te lahko predstavljajo različne stavbe, zelenje, ceste itd. Prekrivnih plasti je

lahko več in se lahko med seboj tudi prekrivajo, kar lahko povzroči razne nevšečnosti, kot so npr. podvojene stavbe na isti lokaciji, drevo sredi hiše itd. Da se temu izognemo, lahko v programu WED definiramo tako imenovana Izključitvena območja (angl. Exclusion Zone), ki zagotovijo, da se na tistih območjih ne prikazujejo objekti spodnjih plasti. To tehniko smo uporabili v generičnem okolju, da smo ustrezno pokrili objekte privzetega okolja X-Plane simulatorja. Težavo prekrivanja specifičnega in generičnega okolja pa smo rešili že korak prej, saj smo iz OSM okolja že pred pretvorbo v programu OSM2XP izbrisali objekte, prisotne v specifičnem okolju.

WorldEditor omogoča tudi pregled okolje in pripravo seznam napak, ki jih je treba popraviti pred vnosom v letalski simulator, da bo v njem okolje ustrezno delovalo. S pomočjo te funkcionalnosti smo popravili večje število poligonov, saj so se nekateri prekrižali, imeli vozlišča preblizu ali pa je bila pot usmerjena v napačno smer. Vse to je zmotilo končni prenos prostorskih podatkov v zapis, ki je razumljiv X-Plane simulatorju. Ko je bilo okolje validirano, smo lahko s pomočjo WED programa to okolje izvozili v obliko, pripravljeno za vnos v simulator (strukturiran datotečni sistem).

Nato smo se lotili iterativnega postopka testiranja s pomočjo pregledovanja pokrajine znotraj simulatorja. Začeli smo s temeljitimi pregledi manjših površin generičnega okolja in nato popravili opažene napake. Največkrat je šlo za moteče 3D modele in fasade, ki so neprijetno izstopale. Ta postopek smo večkrat ponovili. Kasneje smo testirali vedno večje okolje in popravili nevšečnosti, kot so izstopajoče barve streh, pomanjkanja gozdnih površin na mestih, kjer se gozd v resnici nahaja itd. Poleg tega smo tekom testiranja ugotavljali optimalne vrednosti za razne nastavitve programa OSM2XP. Najbolj smo se osredotočili na nastavitvi objSizeTolerance in objHeightTolerance, ki določata največje dovoljeno razmerje med dimenzijami 3D modela in poligona pri generiranju okolja. Višja vrednost pomeni, da dovolimo večjo razliko, kar vodi do večje pokritosti objektov. Vendar se lahko pri previsoki vrednosti pojavijo nesmiselno veliki objekti v simulatorju na mestu manjših stavb in obratno. Nižja vrednost vodi do nižje pokritosti, vendar pri tem poskrbi, da so objekti v simulatorju čim bolj podobni resničnim glede na velikost. Po več iteracijah testiranja smo se odločili vrednost za obe nastavitvi dvigniti iz 0,1 na 0,3, s čimer smo dosegli generiranje več kot 10000 novih objektov brez opaznih neujemanj v okolju.

Po testiranju vseh treh okolij posebej smo jih združili in pričeli z iterativnim testiranjem celotnega v sodelovanju z mentorji iz podjetja AformX. Osredotočili smo se predvsem na popravljanje in brisanje podvojenih objektov, saj so bili nekateri objekti hkrati generirani v generičnem okolju in ročno postavljeni v specifično okolje. V tem koraku smo izvedli tudi manjše prilagoditve, predvsem popravke barv fasad in ortofoto podlage.

3 REZULTATI

Končni rezultat projekta je virtualni pokrajinski dodatek, ki potopitveno predstavlja Spodnjesavinjsko dolino in Celjsko kotlino, in je vidna nadgradnja nad privzetim okoljem, ki je vsem uporabnikom na voljo ob vzpostavitvi simulatorja. Okolje je dostopno javnosti na povezavi [8] in preko QR kode na desni strani odstavka:

V novem okolju je veliko več objektov, ki izgledajo bolj pristno in bolj podobni svojim fizičnim različicam. Barve okolja so veliko bolj naravne in podobne temu, kar pilot običajno vidi ob pogledu iz kabine med preletom območja. Vredno je omeniti, da je v novem okolju na primernih mestih vsebovan tudi hmelj, ki velja za eno od značilnosti te regije, ki je v privzetem okolju manjkala. Zaradi značilnosti izdelka in pomanjkanja ustreznih testnih statistik nismo našli primernega načina kvantificiranja napredka, a je ta kljub temu očiten. Za podporo zgornjih trditev smo izvedli t. i. »Eye Test« in primerjali slike enakega območja v obeh okoljih. V tiskani izdaji razlike niso tako očitne, saj so slike črno-bele. Ogled slik v polni barvni obliki je možen v elektronski različici članka in na povezavi [9], ki je dostopna tudi preko QR kode na desni strani odstavka:

Prva izmed primerjav prikazuje območje Celja, vidno iz severozahodne smeri. Na levi strani slike je prikazano območje v privzetem okolju X-Plane simulatorja, na desni strani pa v našem izdelanem pokrajinskem dodatku.

Razlika med prikazanima podobama največjega mesta v tej regiji je zelo opazna. V privzetem okolju manjkajo glavne značilnosti mesta, med njimi tudi Celjski grad in pretežno industrijski predel mesta. V posodobljenem okolju pa so vse značilnosti Celja ja-

Slika 1: **Primerjava podobe Celja v privzetem in posodobljenem okolju**

sno vidne in tudi sama podoba mesta izgleda veliko bolj naravno in subjektivno tudi lepše. Poleg razlike v mestih, sama pokrajina (polja, gozdovi) izgleda veliko bolj pristno in podobna fizični različici te pokrajine. Možno je opaziti tudi večje število stavb in vasi v novem okolju, kar je še posebej opazno na desni strani obeh slik, saj je v privzetem okolju velika večina vasi izpuščena.

Slika 2: **Primerjava privzete in posodobljene podobe Spodnjesavinjske doline, prikazani so kraji Polzela, Parižlje, Breg pri Polzeli itd.**

Druga primerjava primerja podobi Spodnjesavinjske doline, podrobneje področje krajev Polzela, Parižlje, Breg pri Polzeli itd. Na levi strani slike je prikazano to območje v privzetem okolju X-Plane simulatorja, na desni strani pa v našem izdelanem pokrajinskem dodatku.

Razlike v pristnosti izgleda same pokrajine so zelo opazne tudi v drugi primerjavi, morda še najbolj pride do izraza prisotnost polj hmelja v našem novem okolju, oz. pomanjkanje hmelja v privzetem okolju, saj je ta rastlina eden izmed značilnosti te regije. Ponovno je opazna tudi očitna razlika v barvi pokrajine. Vidna je tudi razlika v reprezentativnosti manjših vasi, ki so v posodobljenem okolju veliko bolje zastopane, saj jih v privzetem praktično ni.

Poleg samega virtualnega dodatka smo razvili preizkušen postopek razvoja virtualnega pokrajinskega dodatka za letalske simulatorje. Ta je osredotočen na področje Slovenije, saj uporablja sredstva, kot so podatki GURS, a bi lahko podoben postopek uporabili tudi za druga območja. Tako študentje kot mentorji smo tekom projekta osvojili nova znanja, ki so prenosljiva tudi na druge domene, in hkrati pridobili stik z gospodarstvom ter izkušnje izven strogo akademskega okolja.

4 ZAKLJUČEK

Projekt Prostorska identiteta v letalskih simulatorjih je vključeval raziskavo pokrajinskih značilnosti regije, predelavo odprtih prostorskih podatkov OpenStreetMap in Geodetske uprave Republike Slovenije, izdelavo 3D modelov objektov, dodelavo odprtokodnega programa OSM2XP in iterativno testiranje ter izboljševanje okolja. Končni rezultat projekta je virtualni pokrajinski dodatek, ki realistično predstavlja Spodnjesavinjsko dolino in Celjsko kotlino ter prinaša vidno izboljšavo v primerjavi s privzetim okoljem. K izboljšavi so prispevale vključitev pomembnih prepoznavnih točk, kot je Celjski grad, vključitev večjega števila objektov, ki so v privzetem okolju manjkali, in uporaba realistične ortofoto podlage, pridobljene s pomočjo satelitskih slik pokrajine. Poleg okolja je bil v sklopu projekta razvit tudi postopek razvoja virtualnega pokrajinskega dodatka, ki bo lahko uporaben za podobne projekte v prihodnosti.

Projekt je bil časovno zahteven, saj je vsak izmed desetih študentov v povprečju porabil približno 140 ur v šestih mesecih, kar ne vključuje številnih dodatnih ur, ki so jih prispevali mentorji. Velik del časa, predvsem v začetni fazi, je bil namenjen spoznavanju novih tehnologij in okolij, kar je študentom in mentorjem prineslo pomembna nova znanja. Če bi postopek razvoja ponovili, bi bil zaradi pridobljenih izkušenj in vzpostavljenega delovnega procesa bistveno hitrejši. Smiselno bi bilo raziskati tudi alternativne vire prostorskih podatkov, ki bi lahko poenostavili in pospešili razvojni proces. To bi zahtevalo tudi prilagoditev ali razvoj novega orodja, primernega za drugačne vrste podatkov, saj bi bil OSM2XP v takšnih primerih neustrezna rešitev.

Projekt in njegovi rezultati lahko služijo kot odskočna deska za večje projekte podobnega tipa, morda nas v prihodnosti čaka izdelava celotne virtualne Slovenije.

LITERATURA

[1] Juvančič, M., Alessio, L., Knez, S., Gradič, V., & Bizjak, L. (2023). *Spodnjesavinjska dolina in Celjska kotlina (Slovenija): pokrajinski dodatek za X-Plane 11 in 12.* Ljubljana: Univerza v Ljubljani, Fakulteta za arhitekturo.

[2] Coast, S. (2004). Open Street Map – prostorski podatki https://www.openstreetmap.org/#map=8/46.150/14.975 (Dostopano dne: 3. marec 2023)

[3] Geodetska Uprava Republike Slovenije (n.d.). E-Prostor: Portal Prostor Geodetske uprave RS. https://www.e-prostor.gov.si/ (Dostopano dne: 24. marec 2023)

[4] OSM2XP – predelana različica programa za namene tega projekta je kot svoja veja dostopna preko platforme GitHub na povezavi https://github.com/leonalessio/osm2xp/tree/razvoj

[5] Coast, S. (2004). Open Street Map Wiki. https://wiki.openstreetmap.org (Dostopano dne: 2. februar 2024)

[6] Immanuel Scholz (2006). JOSM (Java OpenStreetMap Editor). https://josm.openstreetmap.de/ (Dostopano dne: 24. marec 2023)

[7] Supnik, B., Maggi, C. (2007). WorldEditor (WED). https://developer.x-plane.com/tools/worldeditor/ (Dostopano dne: 19. april 2023)

[8] razviti pokrajinski dodatek za X-Plane 11 in 12 https://forums.x-plane.org/index.php?/files/file/87635-spodnjesavinjska-valley-and-celje-basin-slovenia/ (Dostopano dne: 15. januar 2025)

[9] visoko ločljivi barvni posnetki razvitega virtualnega okolja https://ibb.co/album/39XH7j (Dostopano dne: 15. januar 2025)

▪

Leon Alessio je absolvent dodiplomskega programa Računalništva in informatika na Fakulteti za računalništvo in informatiko. V svoji diplomski nalogi napoveduje nizko energijsko razpoložljivost pri plesalcih s pomočjo strojnega učenja. Sodeloval je pri projektu Prostorska identiteta Slovenije v letalskih simulatorjih kot edini študent računalništva. Trenutno tudi sodeluje pri raziskovalnem projektu Deepfake DAD, katerega cilj je razvoj modelov za prepoznavanje globokih ponaredkov. Izkušnje nabira kot razvijalec zalednih sistemov pri IN2. Ukvarja se tudi z razvojem spletnih aplikacij.

▪

Dr. Miha Janež je zaposlen kot asistent na Fakulteti za računalništvo in informatiko. Na tej fakulteti je doktoriral z delom Metode razmeščanja in povezovanja logičnih primitivov kvantnih celičnih avtomatov. Raziskovalno se ukvarja predvsem s področji kvantnega računalništva, načrtovanja brezžičnih senzorskih omrežij in analizo podatkov, pridobljenih iz raznolikih velikih omrežij. Poleg poučevanja sodeluje tako pri študentskih projektih kot tudi pri interdisciplinarnih raziskovalnih projektih.

Pristopi k denormalizaciji besedil: pregled področja

Melanija Vezočnik in Marko Bajec
Univerza v Ljubljani Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana, Slovenija
melanija.vezocnik@fri.uni-lj.si, marko.bajec@fri.uni-lj.si

Izvleček

Sodobni sistemi za samodejno razpoznavo govora učinkovito pretvorijo govorjeni jezik v pisno obliko, vendar pogosto ustvarijo zgolj surov prepis brez ustrezno oblikovanih števil, datumov in časovnih izrazov, kar zmanjšuje njegovo berljivost in uporabnost. Denormalizacija je postopek, ki odpravlja te pomanjkljivosti tako, da preoblikuje prepis v standardizirano pisno obliko. Članek podaja sistematičen pregled in analizo glavnih pristopov k denormalizaciji, ki jih je mogoče razvrstiti v tri skupine: pristopi, ki temeljijo na pravilih, nevronski pristopi ter hibridni pristopi. Pristopi, ki temeljijo na pravilih, tipično izhajajo iz končnih avtomatov, nevronski pristopi uporabljajo nevronske mreže, hibridni pristopi pa združujejo elemente obeh pristopov. Pristopi, ki temeljijo na pravilih, dosežejo visoko natančnost, a ne upoštevajo konteksta besedila. Nasprotno nevronski pristopi upoštevajo kontekst besedila, vendar pa zahtevajo obsežne količine podatkov za učenje. Hibridni pristopi predstavljajo kompromisno rešitev, ki združuje prednosti obeh pristopov. Delo prispeva k razumevanju izzivov ter izboljšanju učinkovitosti denormalizacijskih sistemov.

Ključne besede: denormalizacija besedila, inverzna normalizacija besedila, pregled področja, samodejna razpoznava govora

Approaches to Inverse Text Normalization: A Review

Abstract

Modern automatic speech recognition systems effectively convert spoken language into written text. However, they often produce only a raw transcript without properly formatted numbers, dates, and time expressions, which can reduce readability and usability. Denormalization is a process that addresses these issues by transforming the transcript into a standardized written form. This article provides a systematic review and analysis of the main approaches to denormalization, which can be classified into three categories: rule-based, neural, and hybrid approaches. Rule-based approaches typically rely on finite-state machines, while neural approaches utilize neural networks. Hybrid approaches combine elements of both approaches. Rule-based approaches achieve high accuracy but tend to overlook the context of the text. In contrast, neural approaches consider the context but require large amounts of training data. Hybrid approaches offer a balanced solution that harnesses the strengths of both approaches. This work contributes to understanding the challenges and improving the efficiency of denormalization systems.

Keywords: automatic speech recognition, inverse text normalization, review, text denormalization

1 UVOD

S hitrim razvojem tehnologije in umetne inteligence so računalniški sistemi vse bolj sposobni obdelovati jezik na načine, ki posnemajo človekovo razumevanje. Sistemi za samodejno razpoznavo govora, ki omogočajo pretvorbo govorjenega jezika v pisno obliko, predstavljajo pomemben del tega napredka [1], [2], [3]. Generiranje naravnega, človeku berljivega besedila, je ključno tako za izboljšanje uporabniške izkušnje kot za zagotavljanje točnih in uporabnih informacij v različnih kontekstih. Med najpogostejšimi aplikacijami so pametni pomočniki, kot sta med drugim Siri in Google Assistant, samodejno generiranje podnapisov in sodobno iskanje informacij.

Čeprav so sodobni sistemi za samodejno razpoznavo govora zelo učinkoviti pri pretvorbi govorjenega jezika v pisno obliko, rezultat pogosto ostaja zgolj surov prepis, v katerem pisna znamenja, med drugim števila, datumi in čas, nimajo ustrezne oblike [4]. Na primer, številski izraz *dva tisoč petindvajset* se ne pretvori v številčno obliko 2025. Takšna odstopanja zmanjšujejo berljivost in uporabnost besedila, še posebej pri samodejnem generiranju podnapisov. Denormalizacija besedila je postopek, ki pretvori prepis v standardizirano pisno obliko, ki med drugim vključuje pravilne zapise števil, datumov, časovnih izrazov in okrajšav [5]. Učinkovita denormalizacija izboljša uporabniško izkušnjo, saj zagotavlja, da je izhodno besedilo usklajeno s pričakovanji končnega uporabnika. Zaradi jezikovnih dvoumnosti, raznolikih kontekstov in specifičnih oblikovnih pravil, vezanih na posamezna področja, denormalizacija še vedno predstavlja zahteven izziv [6].

Nekateri sodobni sistemi, kot je med drugim Whisper [7], omogočajo generiranje prepisov v standardizirani obliki in tako odpravljajo potrebo po ločenem koraku denormalizacije. Vendar pa takšne rešitve niso vedno dostopne ali ustrezno prilagojene specifičnim jezikom in aplikacijam. Poleg tega razvoj integriranih rešitev pogosto zahteva velike količine anotiranih podatkov, ki pri jezikih z omejenimi viri, kot je slovenščina, običajno ni na voljo. Nadalje zvočni signal na postopek denormalizacije ne vpliva, saj ne vsebuje indikacij o formatu zapisa [8]. Zato je denormalizacijo smiselno ločiti od sistema za samodejno razpoznavo govora, saj omogoča večjo prilagodljivost in boljšo uporabnost besedil za nadaljnjo obdelavo, na primer pri integraciji s sistemi, ki zahtevajo strukturirane podatke. Zato so samostojni pristopi k denormalizaciji še vedno široko razširjeni.

Kolikor nam je znano, celovitega preglednega članka, ki bi sistematično obravnaval področje denormalizacije besedil v kontekstu samodejne razpoznave govora, v obstoječi znanstveni literaturi še ni. V tem kontekstu članek podaja sistematičen pregled obstoječih pristopov k denormalizaciji, pri čemer analizira njihove značilnosti, prednosti in omejitve ter ocenjuje njihovo primernost za različne jezikovne in aplikativne kontekste. Na podlagi teh ugotovitev želimo v nadaljevanju razviti učinkovit denormalizator za slovenski jezik, ki je kot jezik z omejenimi viri pogosto spregledan v sodobnih rešitvah za obdelavo naravnega jezika. Takšno orodje lahko bistveno izboljša uporabnost govornih prepisov v različnih aplikacijah, kot so samodejno generiranje podnapisov avdio- in video vsebin, arhiviranje sodnih ali parlamentarnih sej, prepisovanje intervjujev v raziskovalne namene ter glasovno upravljanje v lokaliziranih pametnih asistentih. Nadalje tudi olajša dostop do informacij uporabnikom s posebnimi potrebami.

Struktura preostanka članka je naslednja. Drugi razdelek opisuje metodologijo izbire člankov, tretji obravnava pristope k denormalizaciji besedil in vsebuje primerjalno analizo identificiranih pristopov, četrti razdelek vključuje razpravo, peti pa sklepne ugotovitve.

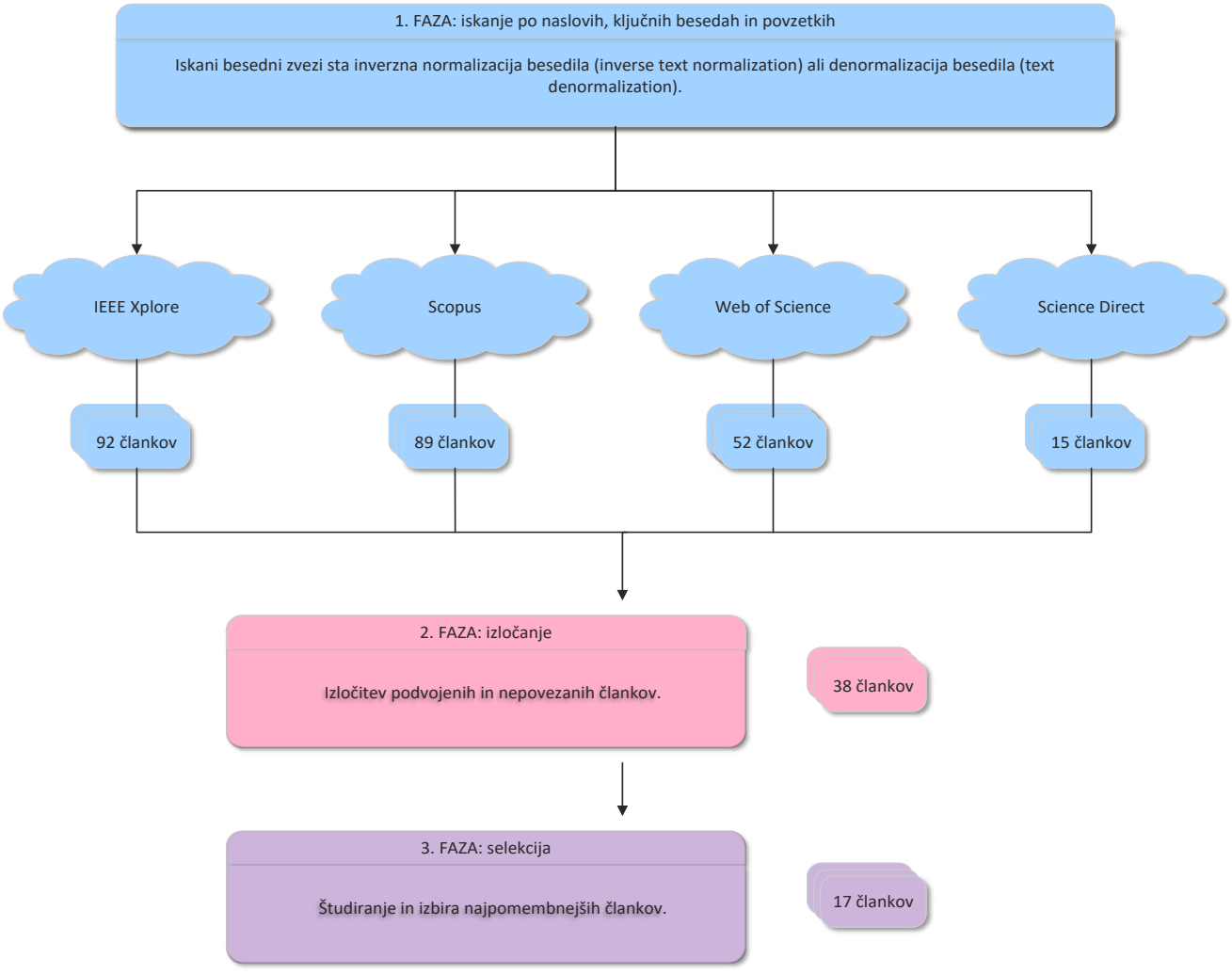
2 METODE

Postopek pregleda literature smo zasnovali sistematično in ga izvedli v treh fazah, kot prikazuje slika 1. Struktura članka dosledno sledi tej metodološki zasnovi. V prvi fazi smo skladno z zastavljenimi raziskovalnimi vprašanji identificirali in analizirali nabor relevantnih člankov na podlagi ključnih besed. V drugi fazi smo izvedli sistematični pregled pristopov k denormalizaciji besedil, kot jih predstavljajo relevantne raziskave v zadnjem desetletju. V tretji fazi smo izvedli poglobljeno primerjalno analizo izbranih rešitev, kjer smo se osredotočili na ključne značilnosti, prednosti in slabosti. Na podlagi ugotovitev, pridobljenih skozi celoten postopek, smo na koncu podali smernice in možnosti za nadaljnji razvoj raziskovalnega področja.

V prvi fazi pregleda področja smo najprej definirali raziskovalna vprašanja, ki so omogočila celovito obravnavo tematike denormalizacije besedil.

- Kakšna je zasnova obstoječih rešitev za denormalizacijo besedil?
- Katere kriterije je mogoče uporabiti za oceno učinkovitosti in natančnosti pristopov za denormalizacijo besedil?
- Na kakšen način denormalizacija besedil vpliva na praktične aplikacije in kakšne so njene širše posledice za področje obdelave naravnega jezika?
- Kakšni so odprti problemi in perspektivne usmeritve v raziskavah, povezanih z denormalizacijo besedil?

Literaturo smo pregledali z iskanjem po digitalnih bibliografskih bazah IEEE Xplore, Scopus, Web of Science in ScienceDirect. Za zajem ustreznega nabora relevantnih člankov smo definirali nabor ključnih besed, ki smo jih uporabili pri iskanju v naslovih,



Slika 1: **Diagram poteka postopka pregleda literature.**

povzetkih in ključnih besedah člankov. Uporabljena iskalna izraza sta bila:

- inverzna normalizacija besedila (inverse text normalization) ter
- denormalizacija besedila (text denormalization).

Ključna izraza smo iskali kot posamezne besedne zveze (in ne kot celotne fraze z navednicami), kar je omogočilo zajem širšega nabora zadetkov. Iskanje je bilo omejeno na naslove, povzetke in ključne besede člankov. Poleg tega smo uporabili filtra za angleški jezik ter časovno obdobje med leti 2015 in 2025. S tem smo zagotovili osredotočenost na aktualne in znanstveno relevantne prispevke.

V drugi fazi smo iz začetnega nabora 248 člankov najprej odstranili duplikate. Nato smo preučili povzetke in naslove preostalih člankov in izločili tiste članke, ki niso neposredno obravnavali problematike denormalizacije besedil ali so bili metodološko nepovezani z obravnavanim področjem. Po tej fazi je ostalo 38 člankov, ki smo jih podrobneje analizirali na osnovi celotnega besedila. Končni nabor je obsegal 17 člankov, ki so izpolnjevali vsebinske in metodološke kriterije ter ustrezali zastavljenim raziskovalnim vprašanjem. Izbrani članki celovito pokrivajo obravnavano temo ter predstavljajo osnovo za nadaljnjo analizo. Na podlagi vsebinske obravnave člankov smo pripravili sistematični pregled pristopov k denormalizaciji besedil, ki je predstavljen v naslednjem razdelku.

3 DENORMALIZACIJA BESEDILA

Denormalizacija besedila je postopek, ki prepis pretvori v standardizirano pisno obliko [5]. Ta običajno vključuje ustrezno zapisane glavne in vrstilne štev-

nike, decimalna števila, datume, časovne izraze, telefonske številke, elektronske naslove, kratice ter okrajšave [4]. Besede, ki sodijo v takšne specifične kategorije, uvrščamo v tako imenovane semiotične razrede.

Ker zvočni signal ne vsebuje informacij o tem, kako naj bodo izrazi iz različnih semiotičnih razredov zapisani v pisni obliki, denormalizacija običajno poteka kot ločen korak po fazi samodejne razpoznave govora [8]. Tako je vhod sistema za denormalizacijo besedila običajno prepis, ki ga je generiral sistem za samodejno razpoznavo govora. Izhod pa je konvencionalna, človeku bolj berljiva pisna različica prepisa. Učinkovitost sistemov za denormalizacijo besedil se običajno ocenjuje z deležem besednih napak, ki meri razlike med izhodnim in pričakovanim besedilom na ravni posameznih besed.

Pristope k denormalizaciji besedil lahko razvrstimo v tri glavne kategorije, in sicer v

- pristope, ki temeljijo na pravilih (razdelek 3.1),
- nevronske (razdelek 3.2) in
- hibridne pristope (razdelek 3.3).

Pristopi, ki temeljijo na pravilih, uporabljajo eksplicitna slovnična in formalna pravila za sistematično pretvorbo besedila v konvencionalno pisno obliko. Nevronski pristopi se učijo kompleksnih jezikovnih vzorcev iz velikih količin anotiranih podatkov z uporabo modelov globokega učenja. Hibridni pristopi združujejo elemente obeh pristopov ter vključujejo slovnična pravila in nevronske modele za obravnavo konteksta in variabilnih jezikovnih struktur.

V nadaljevanju podrobneje predstavimo značilnosti, prednosti in omejitve posameznih kategorij pristopov.

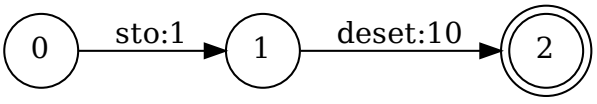
3.1 Pristopi, ki temeljijo na pravilih

Pristopi k denormalizaciji besedil, ki temeljijo na pravilih, tipično vključujejo uporabo končnih avtomatov ali sorodnih formalnih modelov, s katerimi definiramo gramatiko jezika [4]. Učinkovitost teh pristopov je v veliki meri odvisna od natančnosti gramatike, ki definira jezikovno specifična slovnična pravila in strukture. Razvoj tovrstnih sistemov zato zahteva poglobljeno jezikoslovno znanje ter natančno poznavanje strukture in značilnosti obravnavanega jezika.

Postopek denormalizacije besedil v sistemih je običajno dvofazni in sestoji iz klasifikacije in verbalizacije [4], [9], [10]. Prva faza zajema klasifikacijo enot, kot so besede ali fraze, v ustrezne semiotične razrede [4]. Semiotični razredi med drugim zajemajo glavne in vrstilne številnike, decimalna števila, datume, časovne izraze, telefonske številke, elektronske naslove, kratice, okrajšave in splošne besede. Za vsakega izmed identificiranih semiotičnih razredov je potrebno definirati ustrezno gramatiko, ki omogoča njihovo natančno prepoznavanje v fazi klasifikacije. Enote, ki jih ni mogoče uvrstiti v noben semiotični razred, se običajno razvrstijo v splošno kategorijo besed. Druga faza pa zajema verbalizacijo, ki pretvori enote v ustrezno pisno obliko [4].

Slika 2 prikazuje primer končnega avtomata, ki sprejme zaporedje slovenskih številskih besed »sto« in »deset« ter na podlagi prehodov preide med začetnim, vmesnim in končnim stanjem. Ta stanja so po vrsti označena s števili 0, 1, in 2. Prehoda med stanji sta dva. Iz stanja 0 v stanje 1 vodi prehod z oznako sto:1, kar pomeni, da ob vhodnem simbolu sto avtomat doda vrednost 1. Iz stanja 1 v stanje 2 vodi prehod z oznako deset:10, kar pomeni, da ob vhodnem simbolu deset avtomat doda vrednost 10. Skupaj avtomat ob zaporedju simbolov sto in deset doseže končno stanje in s tem sprejme vhod. Skupna številska vrednost prepoznanega zaporedja je 110.

Čeprav pristopi, ki temeljijo na pravilih, še zdaleč niso novi, še vedno ostajajo pomembni tudi v sodobnih sistemih. Večina tradicionalnih sistemov za denormalizacijo besedil uporablja gramatike, definirane z uporabo končnih avtomatov. Pogosto so ti sistemi uporabni tudi pri normalizaciji besedil, ki je obraten postopek od denormalizacije besedil. Primera takih sistemov sta Googlova Kestrel [10] in Sparrowhawk [9], ki predstavlja odprtokodno različico rešitve Kestrel [10]. Rešitev Kestrel [10] podpira angleščino in ruščino, njen povprečni delež besednih napak na naboru podatkov po meri pa je pod 10 %.



Slika 2: **Primer končnega avtomata.**

Zhang idr. [4] so predlagali rešitev NeMo Text Processing [11], ki temelji na ogrodju Sparrowhawk [9] ter podpira normalizacijo in denormalizacijo besedil. Omenjena rešitev uporablja dvofazni postopek, ki v prvi fazi razvrsti besede ali fraze v semiotične razrede, v drugi fazi pa jih pretvori v ustrezno obliko [4]. Zhang idr. [4] so predlagano rešitev za angleški jezik testirali na Googlovi testni množici za normalizacijo [12] in poročali povprečni delež besednih napak v višini 10,1 %. Knjižnica NeMo Text Processing [4] omogoča definiranje gramatik za druge jezike in izvoz definiranih gramatik v format za uporabo v ogrodjih Sparrowhawk [9] ali NVIDIA Riva [13]. Wang in Wang [14] sta prilagodila knjižnico NeMo Text Processing [4] specifični domeni dispečerstva v elektroenergetskem sektorju za kitajščino. Avtorja poročata o 10,5 % povprečnem deležu besednih napak na naboru podatkov po meri.

Kljub visoki učinkovitosti pa takšni pristopi kažejo omejeno prenosljivost med jeziki, saj vsak jezik zahteva definiranje posebej prilagojenih pravil in za slovnične pretvorbe. Poleg tega je razvoj tovrstnih sistemov časovno zahteven, saj vključuje ročno kodiranje kompleksnih jezikovnih pravil in struktur. Kljub navedenim omejitvam pa imajo pristopi, ki temeljijo na pravilih, še vedno pomembno vlogo v okoljih, kjer je ključna visoka stopnja nadzora nad vedenjem sistema, ali kjer primanjkuje podatkov za učenje modelov. V nadaljevanju predstavimo nevronske pristope, katerih razvoj je tesno povezan z napredkom na področju globokega učenja ter z vse večjo dostopnostjo obsežnih jezikovnih korpusov.

3.2 Nevronski pristopi

Poleg pristopov, ki temeljijo na pravilih in zahtevajo ročno definiranje slovničnih pravil, so se v zadnjem desetletju uveljavili nevronski pristopi. V primerjavi s pristopi, ki temeljijo na pravilih, izkazujejo visoko stopnjo prilagodljivosti in zmogljivosti pri denormalizaciji besedil, saj se učijo kompleksnih jezikovnih vzorcev iz velikih količin anotiranih podatkov z uporabo modelov globokega učenja, med katerimi prevladujejo nevronske mreže s povratno zanko, konvolucijske nevronske mreže, nevronske mreže z dolgim kratkoročnim spominom in transformerji [15], [16], [17].

Jedro večine nevronskih pristopov predstavlja t.i. model zaporedje v zaporedje (seq2seq) [15], [18], ki je učinkovit tudi pri denormalizaciji besedila. Rešitve, ki temeljijo na modelu seq2seq, običajno uporabljajo arhitekturno zasnovo kodirnik–dekodirnik, pri čemer kodirnik pretvori vhodno zaporedje v notranjo predstavitev, dekodirnik pa slednjo pretvori v izhodno ciljno zaporedje. Obe komponenti sta tipično realizirani z modeli globokega učenja [15], [18].

Suter in Novak [16] sta zasnovala rešitev na modelu seq2seq z uporabo transformerja. Njihovi modeli za angleščino, ruščino in nemščino so dosegli povprečni delež besednih napak 6,36 %, 4,88 % oziroma 5,23 %. Za angleščino ter ruščino so uporabili Sproutovo množico podatkov, ki je javno dostopna na platformi Kaggle [19], za nemščino pa so uporabili ParaCrawl [20]. Tudi Nayan in Haque [21] sta rešitev zasnovala na modelu seq2seq, pri čemer sta uporabila nevronsko mrežo z dolgim kratkoročnim spominom. Za izboljšanje učenja predlaganega modela za denormalizacijo in normalizacijo sta Nayan in Haque [21] uporabila tehniko CycleGAN, ki omogoča učenje brez popolnoma poravnanih parov podatkov. Za angleščino je predlagana rešitev dosegla povprečni delež besednih napak 33,6 % pri denormalizaciji na Googlovi testni množici [12]. Chen idr. [22] so rešitev prav tako zasnovali z uporabo modela seq2seq in uporabili nevronsko mrežo z dolgim kratkoročnim spominom in transformer. Predlagana rešitev uporablja bogatenje podatkov in nevronsko strojno prevajanje, ki sta zasnovana posebej za jezike z malo viri. Za angleščino je predlagana rešitev dosegla povprečni delež besednih napak 13,8 % na Googlovi testni množici [12] .

Pomembna omejitev modelov seq2seq je njihova dovzetnost za halucinacije, tj. tvorjenje jezikovno pravilnih, a vsebinsko napačnih izhodov. Za odpravo te težave so Antonova idr. [17] predlagali rešitev, ki denormalizacijo obravnava kot nalogo označevanja. Predlagana rešitev vsako besedno enoto bodisi zamenja, kopira brez spremembe ali pa izbriše, kar omogoča večji nadzor nad vedenjem sistema in zmanjšuje tveganje za halucinacije. Predlagana rešitev temelji na enoprehodnem klasifikatorju, ki je enostavnejši kot arhitekturni model seq2seq [17]. Predlagana rešitev je dosegla povprečni delež besednih napak 9,1 % za angleščino in 8,05 % za ruščino na Googlovi testni množici [12].

Ena ključnih prednosti nevronskih pristopov je sposobnost upoštevanja širšega konteksta pri odločanju. To je še posebej pomembno pri dvoumnih zapisih. Na primer, zaporedje *deset trideset* se lahko nanaša na časovni izraz 10.30 ali na glavna številka

10 in 30. Sistemi, ki temeljijo na pravilih, pogosto ne morejo zanesljivo razločiti med takimi primeri, medtem ko se nevronski modeli iz podatkov naučijo razlikovati med možnostmi na podlagi konteksta.

Učenje nevronskih modelov zahteva velike količine anotiranih podatkov, zato glavni izziv pogosto ni samo konstrukcija modela, temveč predvsem pridobivanje in označevanje podatkov. Na primer, Sproutova množica podatkov [19] vsebuje več kot milijardo besednih enot za angleški jezik. Zaradi zahtevnosti ročnega označevanja so bili številni podatkovni korpusi ustvarjeni s pomočjo sistemov, ki temeljijo na pravilih [23]. Kljub temu ostaja velik izziv pridobiti kakovostne učne pare, ki pokrivajo širok spekter semiotičnih razredov, kot so glavni in vrstilni števniki, datumi, časovni izrazi, denar, ulomki, decimalna števila, telefonske številke, mere, elektronski in spletni naslovi ter kratice. To je še posebej problematično za jezike z malo razpoložljivimi viri, kot je slovenščina. Način, kako se nevronski modeli naučijo povezav, ostaja neznan.

Kljub izjemnemu napredku nevronskih modelov pri denormalizaciji besedil se v praksi kaže potreba po večji zanesljivosti, interpretabilnosti in prilagodljivosti sistemov, zlasti v okoljih z omejenimi podatkovnimi viri ali tam, kjer so posledice napak lahko kritične. Nevronski pristopi sicer uspešno zajemajo kontekst in obvladujejo kompleksne jezikovne vzorce, vendar njihova odvisnost od obsežnih anotiranih korpusov, dovzetnost za halucinacije in zahtevnost ponovnega učenja ob spremembah predstavljajo pomembne omejitve. Hibridni pristopi kombinirajo nevronske pristope s pristopi, ki temeljijo na pravilih, in tako omogočijo boljšo obvladljivost sistemskega vedenja, večjo interpretabilnost ter modularno nadgradljivost, kar je še posebej pomembno v okoljih z visokimi zahtevami glede nadzora in zanesljivosti.

3.3 Hibridni pristopi

Hibridni pristopi združujejo nevronske pristope s pristopi, ki temeljijo na pravilih. Pri denormalizaciji besedila so še posebej učinkoviti, saj pravila omogočajo zanesljivo obravnavo sistematičnih vzorcev v besedilu, nevronski pristopi pa omogočajo ustrezno obravnavo kontekstualno pogojenih jezikovnih struktur [23]. Gramatike so tipično bistveno manjše in enostavnejše kot tiste v sistemih za denormalizacijo besedil, ki temeljijo izključno na pravilih. Njihova uporaba zajema klasifikacijo besedila ali korekcijo izhodov nevronskega modela. Modularna zasnova hibridnih sistemov omogoča selektivno posodabljanje posameznih komponent, kar bistveno olajša njihovo prenosljivost med jeziki, domenami in aplikacijskimi scenariji [6].

Pusateri idr. [23] so rešitev zasnovali z uporabo pravil in nevronske mreže z dolgim kratkoročnim spominom. Problem denormalizacije besedila so definirali kot problem označevanja in uporabili s pravili definirane gramatike za pripravo besedila za nevronsko mrežo. Predlagano rešitev so ovrednotili na podlagi pomenske enakovrednosti povedi, ki meri pravilnost celotne povedi. Predlagana rešitev je dosegla približno 99 % natančnost. Podobno so Alphonso idr. [24] predlagali hibridni pristop, ki temelji na nevronskih mrežah z dolgim kratkoročnim spominom, n-gramskem jezikovnem modelu in končnih avtomatih. Potencialne rešitve za pisno obliko so generirali z uporabo pravil, nato pa so jih rangirali z uporabo nevronskih pristopov. Povprečni delež besednih napak je znašal 5,6 %.

Tudi Tan idr. [25] so predlagali rešitev, ki združuje prednosti nevronskega učenja in deterministočnega označevanja z uporabo pravil. Predlagana rešitev generira kontekstualne predstavitve vhodnega besedila z uporabo transformerja, nato pa na podlagi pravil, ki temeljijo na končnih avtomatih, poskrbi za ustrezen format besedila. Podobno so Phan idr. [26] uporabili nevronski model seq2seq za klasifikacijo semiotičnih razredov, nakar so klasificirane segmente pretvorili z uporabo pravil. Glavna prednost njihovega pristopa je v jasni razmejitvi med komponento, ki realizira pravila, in nevronsko komponento, kar omogoča večjo modularnost. Podobnemu principu sledi tudi hibridni pristop, ki so ga predlagali Gaur idr. [8], kjer so uporabili nevronski model transformer za kategorizacijo besednih enot v semiotične razrede, nato pa na označenih segmentih uporabili pravila, specifična za vsak semiotični razred. Ta pretočna in računsko učinkovita arhitektura omogoča visoko natančnost ob znatno manjši kompleksnosti. Vse predlagane rešitve dosegajo primerljive rezultate z drugimi rešitvami na številnih množicah podatkov.

Uspešnost teh sistemov v veliki meri ostaja odvisna od pokritosti jezikovnih pravil, saj se v odsotnosti ustrezne gramatike sistem povrne k nevronskemu modelu. S tem se tveganje za halucinacije ponovno poveča. Nekateri pristopi vključujejo pravila že v fazi predobdelave, s čimer zmanjšajo odvisnost od

nevronskih modelov in izboljšajo robustnost sistema že pred dejanskim procesom denormalizacije. Na primer, Sunkara idr. [6] so uporabili pravila za obdelavo besedila v prvi fazi, nato pa v drugi fazi uporabili na transformerju osnovan model seq2seq. Njihova rešitev omogoča prenosljivost na nove jezike ne da bi potrebovali obsežno jezikovno znanje.

Guo idr. [27] so zasnovali rešitev, ki združuje transformer in pravila, implementirana s končnimi avtomati. Choi idr. [28] pa sistem za denormalizacijo besedil osnovali na arhitekturi, ki je uveljavljena pri strojnem prevajanju, in vključili velik jezikovni model ter transformer. Na ta način so rešili izzive, povezane s pomanjkanjem anotiranih podatkov. Na drugi strani so Wang idr. [14] predlagali hibridni pristop, ki temelji na medjezikovni destilaciji znanja. V sistem so vključili učiteljski model, naučen na jeziku z bogatimi viri, in študentski model, naučen na jeziku z omejenimi viri. Pravila, zasnovana s končnimi avtomati, so vključili kot dodatno obliko nadzora in tako zmanjšali napake in izboljšali kontekstualno interpretacijo.

Hibridni pristopi združujejo robustnost pravil in prilagodljivost nevronskih modelov. Rezultat združevanja različnih pristopov predstavljajo učinkovite rešitve, ki so natančne, razširljive in primerne za produkcijsko rabo. Kljub obetavnim rezultatom pa se ti pristopi razlikujejo glede na obseg in način uporabe pravil, nevronskih modelov, integracije komponent in aplikativnosti. V nadaljevanju podajamo primerjalno analizo obravnavanih pristopov, pri čemer izpostavimo njihove ključne značilnosti, prednosti in slabosti.

3.4 Primerjava pristopov

V nadaljevanju primerjamo tri glavne pristope k denormalizaciji besedil, in sicer pristope, ki temeljijo na pravilih, ter nevronske in hibridni pristope. Za njihovo sistematično primerjavo smo identificirali šest ključnih kriterijev, ki se nanašajo za učinkovitost, uporabnost in vzdrževanje rešitev v različnih jezikovnih in aplikativnih kontekstih. Poleg tega smo upoštevali tudi praktični razvoj, kot sta med drugim modularnost arhitekture in možnost nadgradnje. Izbrani kriteriji omogočajo celovito presojo prednosti in omejitev posameznih pristopov ter lahko predstavljajo osnovo za izbiro najprimernejšega pristopa glede na specifične zahteve aplikacije. Tabela 1 vsebuje primerjavo pristopov.

Tabela 1: **Primerjava pristopov za denormalizacijo besedil.**

kriterij	**pristopi, ki temeljijo na pravilih**	**nevronski pristopi**	**hibridni pristopi**
način delovanja	slovnična pravila, ki tipično temeljijo na končnih avtomatih	nevronski modeli, med drugim nevronske mreže z dolgim kratkoročnim spominom in transformerji, uporabljeni v modelu seq2seq	kombinacija pristopov, ki temeljijo na pravilih, in nevronskih pristopov
potreba po podatkih pri razvoju	ni potrebe po anotiranih podatkih	zahteva po veliki količini anotiranih podatkov	delna odvisnost od anotiranih podatkov
prenosljivost med jeziki	ne	zmerna	zmerna
robustnost	visoka za strukturirane podatke, nizka za odprte kontekste	visoka pri kontekstualni interpretaciji, nizka v primeru halucinacij	visoka, saj združuje prednosti uporabe pravil in upoštevanja konteksta
interpretabilnost	pravila so eksplicitna	delovanje modela težje razložljivo	pravila delno olajšajo razlago rezultatov
razvojna zahtevnost	visoka (zaradi ročnega kodiranja pravil)	srednja do visoka (zaradi zahteve po učenju modelov)	zmerna (zaradi modularne strukture je razvoj nekoliko olajšan)
natančnost	visoka za enostavne primere, nizka za nepodprte primere	visoka za kompleksne kontekste, obstaja možnost pojavljanja halucinacij	visoka zaradi združevanja natančnosti pravil in fleksibilnosti nevronskih modelov
odvisnost od konteksta	ne	da	da

Količina učnih podatkov lahko predstavlja ključen dejavnik pri izbiri pristopa. Pristopi, ki temeljijo na pravilih, so primerni za jezike, kjer ni na voljo večjih količin anotiranih korpusov. Nasprotno nevronski pristopi zahtevajo obsežne količine kakovostno anotiranih podatkov za učenje, kar je lahko omejitveni dejavnik pri manj raziskanih jezikih. Hibridni pristopi omogočajo zmanjšanje potrebe po podatkih, saj kombinacija pravil z učenjem omogoča, da se nevronski model nauči tudi na manjši količini podatkov, pri čemer pravila tipično opredelijo jezikovne strukture v besedilu.

Pristopi, ki temeljijo na pravilih, imajo nizko prenosljivost, saj pri denormalizaciji besedila izhajajo izključno iz definiranih gramatik. Vsak nov jezik zahteva novo definicijo zbirke pravil, kar pomeni visoke stroške razvoja. Nevronski pristopi imajo višjo stopnjo prenosljivosti, še posebej pri uporabi večjezikovnih modelov, ki omogočajo učenje na več jezikih hkrati. Vendar pa zahtevajo velike količine anotiranih podatkov pri učenju, ki pa za jezike z malo viri niso na voljo. Hibridni pristopi lahko z ustrezno zasnovo omogočajo boljšo prenosljivost, vendar še vedno zahtevajo nekaj prilagoditev pri prehodu na nov jezik. Stopnja prenosljivosti hibridnih pristopov je odvisna od razmerja med obsegom na pravilih osnovanih komponent in deležem nevronskih modelov ter načina njihove integracije.

Nevronski pristopi so zmožni upoštevati raznolike in kompleksne jezikovne vzorce, ki se pojavijo v učnih podatkih. Pristopi, ki temeljijo na pravilih, so v primerjavi z nevronskimi pristopi manj robustni, saj kontekstov, ki niso bili eksplicitno definirani s pravili, ne upoštevajo. Hibridni pristopi v tem pogledu ponujajo uravnoteženost, saj s pravili obravnavajo enostavne in ponovljive primere, medtem ko nevronske komponente izboljšajo učinkovitost sistema pri zahtevnejših vhodih.

Interpretacija rezultatov je ključna na primer v pravu ali medicini. Pristopi na osnovi pravil imajo visoko interpretabilnost, saj so pravila eksplicitno definirana in razumljiva. Način, kako se nevronski modeli naučijo povezav, ostaja neznan, zato pogosto ne moremo razložiti, zakaj je model sprejel določeno odločitev. Hibridni pristopi poskušajo ublažiti to pomanjkljivost tako, da za bolj predvidljive jezikovne pojave uporabijo pravila, kar prispeva k večji razložljivosti celotnega sistema.

Razvojna zahtevnost pristopov k denormalizaciji besedil se razlikuje glede na izbrano metodologijo. Pristopi, ki temeljijo na pravilih, zahtevajo ročno definicijo slovničnih pravil, kar zahteva poglobljeno jezikoslovno znanje. Nevronski pristopi zahtevajo obsežno količino anotiranih podatkov in poznavanje področja strojnega učenja. Postopek učenja modelov, optimizacija hiperparametrov in zagotavljanje kakovosti izhodov lahko predstavljajo izziv. Hibridni pristopi omogočajo ponovno uporabo komponent in hitrejšo prilagoditev aplikacijam.

Natančnost označuje stopnjo pravilnosti denormaliziranega besedila. Ena izmed najpogostejših metrik za določanje natančnosti predstavlja povprečni delež besednih napak. Za jezike, kjer so na voljo obsežne zbirke anotiranih podatkov, so nevronski pristopi pogosto najbolj natančni, saj lahko zajamejo kontekst in subtilnosti jezika, ki jih pravila ne upoštevajo. Pristopi, ki temeljijo na pravilih, dosegajo visoko natančnost pri omejenem in dobro definiranem domenskem besedilu, vendar pogosto dosegajo slabše rezultate pri variabilnejšem jeziku. Hibridni pristopi dosegajo primerljive rezultate, saj lahko kombinirajo prednosti obeh pristopov in tako zmanjšajo verjetnost sistematičnih napak.

Pristopi, ki temeljijo na pravilih, izhajajo iz vnaprej določenih pravil, ki niso občutljiva na širši kontekst povedi ali besedila. Zato lahko v primerih, kjer je interpretacija izrazov odvisna od pomena v konkretnem okolju, pride do napačnih pretvorb ali nedoslednosti. Po drugi strani zasnova nevronskih pristopov omogoča upoštevanje širšega konteksta besedila. Vendar pa so nevronski modeli nagnjeni k halucinacijam. Hibridni pristopi združujejo navedena pristopa ter pravila uporabljajo za sistematične in kontekstualno nevtralne primere, medtem ko nevronske komponente omogočajo interpretacijo izrazov v širšem besedilnem kontekstu. Takšna kombinacija omogoča večjo zanesljivost in prilagodljivost.

4 DISKUSIJA

Pregled literature je temeljil na sistematičnem iskanju znanstvenih prispevkov v recenziranih virih. Pri zasnovi sistematičnega pregleda smo se zavestno omejili na vključitev recenziranih znanstvenih virov, indeksiranih v uveljavljenih zbirkah, kot so IEEE Xplore, Science Direct, Scopus in Web of Science. Čeprav Google Scholar in Semantic Scholar predstavljata pomembni zbirki znanstvene literature, ki vključujeta tudi predobjave ter druge nerecenzirane prispevke, smo se odločili, da njune vsebine v ana-

lizo ne vključimo. Ključni razlog za to odločitev je zagotavljanje konsistentne ravni znanstvene verodostojnosti, saj zbirki ne zagotavljata preglednega nadzora nad kakovostjo objavljenih del. Posledično bi vključitev nerecenziranih virov lahko vplivala na zanesljivost ugotovitev ter otežila primerjalno vrednotenje obravnavanih pristopov. Zavedamo se, da so predobjave, zlasti na hitro razvijajočih se področjih, kot je obdelava naravnega jezika, pogosto nosilci najnovejših raziskovalnih trendov.

V pregledu pristopov k denormalizaciji besedil smo ugotovili, da vsak od treh glavnih identificiranih pristopov izkazuje specifične lastnosti, ki vplivajo na samo natančnost delovanja rešitve. Pristopi, ki temeljijo na pravilih, omogočajo visoko stopnjo nadzora in natančnosti nad predvidenim vhodnim besedilom, saj temeljijo na ročno definiranih jezikovnih pravilih in gramatikah. Njihovo glavno pomanjkljivost predstavljajo visok strošek razvoja, slaba prenosljivost med jeziki ter omejena sposobnost obravnave jezikovnih variacij, ki niso bile predhodno definirane s pravili. Nevronski pristopi predstavljajo sodobnejšo alternativo, saj omogočajo učenje kompleksnih jezikovnih vzorcev iz učnih podatkov brez potrebe po ročnem definiranju pravil. Njihova učinkovitost pa je močno pogojena z razpoložljivostjo obsežnih in raznolikih anotiranih besedil, kar predstavlja izziv predvsem pri jezikih z omejenimi viri. Poleg tega ti pristopi niso vedno zanesljivi, saj lahko generirajo halucinacije ali izhode, ki kljub formalni pravilnosti niso ustrezni v danem kontekstu. Hibridni pristopi, ki združujejo prednosti pristopov, ki temeljijo na pravilih, in nevronskih pristopov, so še posebej obetavni za produkcijska okolja, kjer je potrebna visoka natančnost ob ohranjanju fleksibilnosti. Vendar pa ostajajo izzivi pri zagotavljanju optimalne integracije obeh komponent, še posebej pri usklajevanju rezultatov iz obeh virov.

Kritičen dejavnik pri nevronskih in hibridnih pristopih ostaja dostopnost ustreznih podatkovnih množic. Zaradi omejene razpoložljivosti anotiranih besedil za večino jezikov se pogosto uporabljajo metode za bogatenje podatkov, kar lahko vpliva na kakovost in robustnost naučenih modelov. Pomanjkanje standardiziranih podatkovnih množic za različne jezike prav tako otežuje neposredno primerjavo rezultatov med različnimi študijami. V prihodnje bi bilo smiselno raziskati možnosti prenosa znanja iz virov z bogatimi podatki, kot je na primer angleščina, na jezike z manj viri, kot je slovenščina. Prav tako bi bilo koristno razviti bolj transparentne modele, ki bi omogočali boljšo razlago odločitev, s čimer bi povečali zaupanje uporabnikov v delovanje sistemov, še posebej v občutljivih domenah, kot sta medicina ali pravo.

Sklepamo lahko, da optimalna rešitev za denormalizacijo besedil trenutno še ni univerzalna. Učinkovitost posameznega pristopa je v veliki meri odvisna od razpoložljivih virov, zahtevane stopnje natančnosti ter ciljne domene. Nadaljnje raziskave bi morale stremeti k razvoju prilagodljivih, razširljivih in razložljivih sistemov, ki bodo kos kompleksnosti naravnega jezika v različnih kontekstih.

Denormalizacija besedil ostaja odprt raziskovalni izziv, ki zahteva usklajevanje jezikovno-specifičnih potreb, razpoložljivih virov in kontekstualnih zahtev. V prihodnje bo ključno razvijati prilagodljive in robustne rešitve, ki bodo omogočale učinkovito delovanje tudi v pogojih omejenih podatkovnih virov. Posebno pozornost bo treba nameniti tudi večjezičnosti, razložljivosti modelov ter etičnim in praktičnim vidikom uporabe v realnih aplikacijah.

Med jeziki z omejenimi viri izstopa tudi slovenščina, ki zahteva prilagojene pristope zaradi svojih morfosintaktičnih posebnosti in majhnega števila anotiranih podatkovnih množic. Pregledane usmeritve tako odpirajo možnosti za nadaljnje raziskave, usmerjene v razvoj zanesljivih in razložljivih rešitev za denormalizacijo slovenskega jezika, ki bodo lahko učinkovito podprle tako raziskovalne kot tudi praktične aplikacije v govornih in jezikovnih tehnologijah.

5 ZAKLJUČEK

V članku smo predstavili pregled pristopov k denormalizaciji besedil in jih umestili v širši kontekst jezikovnih tehnologij, s poudarkom na jezikih z omejenimi podatkovnimi viri, kot je slovenščina. Bistvene ugotovitve so sledeče. Pristopi, ki temeljijo na pravilih, so zanesljivi, vendar pogosto premalo prilagodljivi za zajem jezikovne raznolikosti. Nevronski pristopi ponujajo večjo fleksibilnost in sposobnost učenja iz podatkov, a zahtevajo obsežne in kakovostno anotirane korpuse, ki jih za slovenščino primanjkuje. Hibridni pristopi se zato kažejo kot najbolj obetavni za praktično rabo, saj združujejo prednosti obeh pristopov.

Zato izbira optimalnega pristopa ni univerzalna, temveč mora upoštevati ciljno aplikacijo, specifičnost jezika in razpoložljivost virov. V kontekstu slovenščine se kot ključni izzivi pojavljajo pomanjkanje javno

dostopnih anotiranih podatkov, odsotnost standardiziranih evalvacijskih okvirov in omejene zmogljivosti obstoječih orodij za razlago odločitev nevronskih modelov.

V prihodnje bo ključno usmeriti raziskave v razvoj jezikovno prilagojenih denormalizatorjev, zasnovanih posebej za značilnosti slovenskega jezika. Čeprav se v širšem kontekstu pogosto predlaga prenos znanja iz resursno bogatejših jezikov prek večjezičnih ali samonadzorovanih pristopov, se takšni prenosi izkažejo za omejeno uporabne pri slovenščini zaradi izrazitih morfoloških, skladenjskih in pravopisnih posebnosti. Zato je še toliko pomembneje razvijati namenske rešitve, ki upoštevajo slovenske jezikovne norme in rabo.

Poleg tega ostaja izziv tudi razložljivost modelov, ki je ključna za sprejemljivost in zanesljivost uporabe v realnih aplikacijah. Nadaljnji razvoj naj se zato osredotoči tudi na transparentnost sistemov in njihovo prilagodljivost specifičnim okoljem – od samodejnega generiranja podnapisov do glasovnega upravljanja ter digitalne dostopnosti za slovensko govoreče uporabnike. Na podlagi teh ugotovitev želimo v nadaljevanju razviti učinkovit denormalizator za slovenski jezik, ki je kot jezik z omejenimi viri pogosto spregledan v sodobnih rešitvah za obdelavo naravnega jezika.

LITERATURA

[1] R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schlüter, and S. Watanabe, »End-to-End Speech Recognition: A Survey,« *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 325–351, 2024, doi: 10.1109/TASLP.2023.3328283.

[2] D. O'Shaughnessy, »Trends and developments in automatic speech recognition research,« *Computer Speech & Language*, vol. 83, p. 101538, 2024, doi: https://doi.org/10.1016/j.csl.2023.101538.

[3] S. Alharbi *et al.*, »Automatic Speech Recognition: Systematic Literature Review,« *IEEE Access*, vol. 9, pp. 131858–131876, 2021, doi: 10.1109/ACCESS.2021.3112535.

[4] Y. Zhang, E. Bakhturina, and B. Ginsburg, »NeMo (Inverse) Text Normalization: From Development To Production,« in *INTERSPEECH 2021*, in Interspeech. 2021, pp. 4857–4859.

[5] J. Liao, Y. Shi, and Y. Xu, »Automatic Speech Recognition Post-Processing for Readability: Task, Dataset and a Two-Stage Pre-Trained Approach,« *IEEE Access*, vol. 10, pp. 117053–117066, 2022, doi: 10.1109/ACCESS.2022.3219838.

[6] M. Sunkara, C. Shivade, S. Bodapati, and K. Kirchhoff, »Neural inverse text normalization,« in *ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2021, pp. 7573–7577.

[7] H. Ahlawat, N. Aggarwal, and D. Gupta, »Automatic Speech Recognition: A survey of deep learning techniques and approaches,« *International Journal of Cognitive Computing in Engineering*, vol. 6, pp. 201–237, 2025, doi: https://doi.org/10.1016/j.ijcce.2024.12.007.

[8] Y. Gaur *et al.*, »Streaming, Fast and Accurate on-Device Inverse Text Normalization for Automatic Speech Recognition,« in *2022 IEEE spoken language technology workshop (SLT)*, 2023, pp. 237–244. doi: 10.1109/SLT54892.2023.10022543.

[9] »Sparrowhawk.« Na spletu: https://github.com/google/sparrowhawk, 2015.

[10] P. Ebden and R. Sproat, »The Kestrel TTS text normalization system,« *Natural Language Engineering*, vol. 21, no. 3, pp. 333–353, 2015, doi: 10.1017/S1351324914000175.

[11] »NeMo Text Processing.« Na spletu: https://github.com/NVIDIA/NeMo-text-processing, 2021.

[12] »Google text normalization dataset.« Na spletu: https://www.kaggle.com/datasets/google-nlu/text-normalization, 2016.

[13] »NVIDIA Riva.« Na spletu: https://docs.nvidia.com/deeplearning/riva/user-guide/docs/index.html, 2024.

[14] L. Wang *et al.*, »Zero-Shot Text Normalization via Cross-Lingual Knowledge Distillation,« *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 4631–4646, 2024, doi: 10.1109/TASLP.2024.3407509.

[15] A. Vaswani *et al.*, »Attention is all you need,« in *Proceedings of the 31st international conference on neural information processing systems*, in NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 6000–6010.

[16] B. Suter and J. Novak, »Neural Text Denormalization for Speech Transcripts,« in *Interspeech 2021*, 2021, pp. 981–985. doi: 10.21437/Interspeech.2021-1814.

[17] A. Antonova, E. Bakhturina, and B. Ginsburg, »Thutmose tagger: Single-pass neural model for inverse text normalization,« in *INTERSPEECH 2022*, in Interspeech. 2022. doi: 10.21437/Interspeech.2022-10864.

[18] C. Peyser, H. Zhang, T. N. Sainath, and Z. Wu, »Improving Performance of End-to-End ASR on Numeric Sequences,« vol. abs/1907.01372, pp. 2185–2189, 2019, doi: 10.21437/Interspeech.2019-1345.

[19] »Text Normalization for English, Russian and Polish.« Na spletu: https://www.kaggle.com/datasets/richardwilliamsproat/text-normalization-for-english-russian-and-polish, 2020.

[20] M. Esplà, M. Forcada, G. Ramírez-Sánchez, and H. Hoang, »ParaCrawl: Web-scale parallel corpora for the languages of the EU,« in *Proceedings of machine translation summit XVII: Translator, project and user tracks*, M. Forcada, A. Way, J. Tinsley, D. Shterionov, C. Rico, and F. Gaspari, Eds., Dublin, Ireland: European Association for Machine Translation, Aug. 2019, pp. 118–119. Available: https://aclanthology.org/W19-6721/

[21] Md. M. R. Nayan and M. A. Haque, »Parallel Training of TN and ITN Models Through CycleGAN for Improved Sequence to Sequence Learning Performance,« in *2022 asia-pacific signal and information processing association annual summit and conference (APSIPA ASC)*, 2022, pp. 1137–1141. doi: 10.23919/APSIPAASC55919.2022.9980278.

[22] S.-J. Chen, D. Paul, Y. Pang, P. Su, and X. Zhang, »Language Agnostic Data-Driven Inverse Text Normalization,« in *INTERSPEECH 2023*, in Interspeech. 2023, pp. 451–455. doi: 10.21437/Interspeech.2023-2066.

[23] E. Pusateri, B. R. Ambati, E. Brooks, O. Platek, D. McAllaster, and V. Nagesha, »A Mostly Data-Driven Approach to Inverse Text Normalization,« in *INTERSPEECH*, Stockholm, 2017, pp. 2784–2788.

[24] I. Alphonso, N. Kibre, and T. Anastasakos, »Ranking Approach to Compact Text Representation for Personal Digital Assi-

stants,« in *2018 IEEE spoken language technology workshop (SLT)*, 2018, pp. 664–669. doi: 10.1109/SLT.2018.8639542.
[25] S. Tan, P. Behre, N. Kibre, I. Alphonso, and S. Chang, »Four--in-One: a joint approach to inverse text normalization, punctuation, capitalization, and disfluency for automatic speech recognition,« in *2022 IEEE spoken language technology workshop (SLT)*, IEEE, 2023, pp. 677–684.
[26] T. A. Phan, N. D. Nguyen, H. L. Thanh, and K.-H. N. Bui, »Neural Inverse Text Normalization with Numerical Recognition for Low Resource Scenarios,« in *Intelligent information and database systems*, N. T. Nguyen, T. K. Tran, U. Tukayev, T.-P. Hong, B. Trawiński, and E. Szczerbicki, Eds., Cham: Springer International Publishing, 2022, pp. 582–594.
[27] K. Guo, S. S. Clarke, and K. Kalyanam, »Inverse text normalization of air traffic control system command center planning telecon transcriptions,« in *AIAA AVIATION FORUM AND ASCEND 2024*, 2024, pp. 4360–4365.
[28] H. Choi *et al.*, »Spoken-to-written text conversion with Large Language Model,« in *Proc. Interspeech 2024*, 2024, pp. 2410–2414.

▪

Melanija Vezočnik je asistentka v Laboratoriju za podatkovne tehnologije na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Njeno trenutno raziskovalno delo je usmerjeno v področje govornih in jezikovnih tehnologij. Leta 2023 je na isti fakulteti doktorirala iz računalništva in informatike. Med doktorskim študijem se je raziskovalno ukvarjala z analizo hoje z inercijskimi senzorji, še posebej z oceno dolžine koraka. Rezultate svojega raziskovalnega dela redno objavlja v znanstvenih revijah, za raziskovalne dosežke med doktorskih študijem pa je prejela priznanje dekanje.

▪

Marko Bajec je redni profesor na Fakulteti za računalništvo in informatiko (Univerza v Ljubljani) in vodja Laboratorija za podatkovne tehnologije. Ukvarja se z načrtovanjem in razvojem podatkovno intenzivnih sistemov. V zadnjih letih se posveča jezikovnim in govornim tehnologijam ter digitalizaciji slovenskega jezika. Svoje rezultate redno objavlja v domačih in tujih revijah ter konferencah. Je prejemnik več nagrad in priznanj za raziskovalno, pedagoško in aplikativno delo.

Iz Islovarja

Islovar je spletni terminološki slovar informatike in računalištva, ki ga objavlja jezikovna sekcija Slovenskega društva Informatika. Navajamo nekaj izpisov na temo medij:

cíljno oddájanje -ega -a s *(angl. narrowcasting, narrowcast)* oddajanje za določeno publiko, ciljno skupino; prim. oddajanje

digitálni médij -ega -a m *(angl. digital media)* medij, ki omogoča zapis, prenos in shranjevanje digitalnih podatkov

drúžbeni médij -ega -a m *(angl. social media)* spletna tehnologija, ki se uporablja za ustvarjanje in izmenjavo digitalnih vsebin v virtualni skupnosti

elektrónski médij -ega -a m *(angl. electronic media)* medij v elektronski ali elektromehanski obliki za predstavitev analognih ali digitalnih vsebin, npr. televizija, radio, internet; prim. digitalni medij

hípermédij -a m *(angl. hypermedia)* multimedija s hiperpovezavami; prim. nadbesedilo

múltimédija -e ž *(angl. multimedia)* medij (3), ki ima zmožnost hkratne predstavitve podatkov z besedilom, sliko, gibljivo sliko, zvokom

oddájanje -a s *(angl. broadcast, broadcasting)* razpošiljanje avdio- ali videovsebin razpršenemu občinstvu po elektronskih medijih; prim. ciljno oddajanje

pódkast -a m *(angl. podcast, POD)* digitalna vsebina, ki jo je mogoče v obliki avdio- ali videodatotek prenašati s spleta na računalnik ali prenosno napravo

splétna oddája -e -e ž *(angl. webcast)* multimedijska vsebina, ki jo je mogoče s spletne strani prenesti na uporabnikov računalnik

Mednarodni simpozij s področja operacijskih raziskav v Sloveniji (SOR '25)

Bled, Slovenija, 24. do 26. september 2025

https://sor.fov.um.si/

18. mednarodni simpozij s področja operacijskih raziskav v Sloveniji (SOR '25) bo potekal od 24. do 26. septembra 2025 na Bledu v Sloveniji.

Organizirali ga bodo Slovensko društvo Informatika, Sekcija za operacijske raziskave (SDI-SOR, Ljubljana, Slovenija), Univerza v Mariboru, Fakulteta za organizacijske vede (UM-FOV, Kranj, Slovenija), in Rudolfovo - Znanstveno in tehnološko središče Novo mesto (Novo mesto, Slovenija).

Pomembni datumi

Oddaja prispevkov: 10. junij 2025
Poročila recenzentov: 10. julij 2025
Predložitev revidiranih prispevkov: 25. avgust 2025

Družabni program

Sreda, 24. september 2025, popoldne: Druženje z izletom in večerjo.
Četrtek, 25. september 2025, zvečer: Druženje.

Publikacije

e-Zbornik SOR'25 bo na voljo pred začetkom simpozija na spletni strani SDI-SOR. Predvidene so posebne številke revij Central European Journal of Operational Research in Business Systems Research Journal, v katerih bodo objavljene revijalne različice izbranih prispevkov.