

METODOLOGIJA UZORKOVANJA ISKUSTVA (ESM) KORISNIKA U MOBILNOM RAČUNARSTVU**EXPERIENCE SAMPLING METHODOLOGY (ESM) FOR ELICITING USER REQUIREMENTS IN MOBILE APPROXIMATE COMPUTING**Luna Živković, Milan Vidaković, *Fakultet tehničkih nauka, Novi Sad***Oblast – PRIMENJENE RAČUNARSKE NAUKE I INFORMATIKA**

Kratak sadržaj: Eksponencijalne promene nastale usled sve intenzivnije upotrebe informacionih tehnologija, predstavljaju zabrinjavajući faktor zagađenja životne sredine. Razvoj hardvera mobilnih telefona sve teže prati svakodnevno korišćenje istih od strane milijardi korisnika, kao i sve veću kompleksnost aplikacija koje zahtevaju znatnu upotrebu energetske resursa. Takođe, ograničenja baterije izričito ugrožavaju dalju evoluciju mobilnih telefona. Jedan od novih načina redukovanja potrošnje energije koju telefoni zahtevaju, jeste "približno računanje" (eng. *approximate computing*) koje žrtvuje preciznost zarad očuvanja energije. Vodeći se ovim, razvili smo *context-aware* okvir koji se fokusira na ispunjavanje korisničkih očekivanja, dok koristi minimalnu moguću energiju.

Ključne reči: *Android, Java, Aware, Weka Yandex, mobile*

Abstract: *The exponential change in the way Information and Communication Technology is consumed has been so significant that there is an increasing awareness of the potential environmental effects. Underlying mobile hardware does not keep pace with the increased usage of mobile phones in everyday life, as well as the complexity of new apps which demand great energy resources. Limitations in battery technology are especially threatening further mobile computing evolution. A novel approach for reducing the energy appetite of mobile apps comes from the approximate computing field, which proposes techniques that, in a controlled manner sacrifice computation accuracy for higher energy savings. Following this train of thought, we built a context-aware framework that focuses on fulfilling users expectation while using the lowest amount of energy possible.*

Keywords: *Android, Java, Aware, Weka Yandex, mobile*

1. UVOD

Zagađenje životne sredine predstavlja jednu od najaktuelnijih tema današnjice, dok eksponencijalne promene nastale usled sve intenzivnije upotrebe informacionih tehnologija, predstavljaju jedan od glavnih uzroka zagađenja.

Imajući u vidu da danas preko 3 milijarde ljudi svakodnevno koristi pametne telefone, način na koji komuni-

ciramo, poslujemo, putujemo i upoznajemo ljude se znatno transformisao. Iako upotreba mobilnih telefona donosi mnoge koristi, „karbonski otisak“, koji je emitovan u gotovo svakom stadijumu proizvodnje i upotrebe istih, predstavlja značajnu pretnju za životnu sredinu. Na osnovu studija sprovedenih na temu emisije ovih sistema, pretnja se neće vremenom smanjiti sama od sebe, već je potrebno preduzeti brze i znatne mere kako u industrijskom, tako i u političkom sektoru.

Sve veća upotreba i kompleksnost aplikacija zahteva znatne energetske resurse, što razvoj hardvera mobilnog telefona sve teže prati. [1]. Da bi poboljšali dužinu trajanja baterije i smanjili uticaj ICT-a (drugim rečima, kako bi smanjili potrošnju energije), prvo treba da razumemo korisnika i njegove potrebe. U različitim situacijama, korisnici su zadovoljni različitim rezultatima i stoga se treba fokusirati na minimum koji možemo da predstavimo korisnicima, a da oni budu zadovoljni. Imajući u vidu da će svaku aplikaciju, svaki proizvod koristiti različiti ljudi, sa različitim profilima, obrazovanjem i shvatanjima, programeri moraju ostati što objektivniji pri izradi aplikacije.

Ovde uvodimo pojam „približnog računanja“ (eng. *approximate computing*), kao nov načina redukovanja potrošnje energije koju telefoni zahtevaju. Ono se fokusira na smanjenje preciznosti zarad očuvanja energije. Vodeći se ovim, razvili smo okvir koji ispunjava korisnička očekivanja, dok koristi minimalnu moguću energiju.

Zadatak rada predstavlja kreiranje *Android (Java)* okvira koji će omogućiti praćenje zadovoljstva korisnika tokom upotrebe određene aplikacije i dalju primenu dobijenih rezultata za usavršavanje aplikacije. Krajnji cilj je ispunjenje korisničkih očekivanja uz maksimalno smanjenje potrošnje energije. Kao primer primene okvira, korišćena je aplikacija *NewPipe* za pregled videa, inspirisana postojećim rešenjem (*YouTube*). Uključivanjem minimalnog seta senzora (senzori za praćenje baterije, lokacije, korisničke aktivnosti i osvetljenja ekrana), pratimo kontekst u kom se korisnik nalazi. Dobijene rezultate upoređujemo sa prethodno prikupljenim podacima, gde je korisnik davao ocene na kraju gledanja različitih videa u različitim kontekstima. Kao rezultat povezivanja ovih podataka, aplikacija prikazuje video najniže prihvatljive rezolucije. Na ovaj način produžava se život baterije, to jest, smanjuje se potrošnja energije, a zadovoljstvo korisnika ostaje na visokom nivou. Uvažavajući individualizam u preferencijama gledanja, razvijamo metod za iterativnu personalizaciju prikazanog sadržaja. Imajući u vidu da kontekstno svesne aplikacije zahtevaju kontinuirano

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji sje mentor bio dr Milan Vidaković, red. prof.

prikupljanje i tumačenje podataka iz jednog ili više očitavanja senzora, naš metod se zasniva na kombinaciji *active* i *reinforcement* učenja. Pomoću ova dva načina učenja, smanjuje se količina upita (na kraju gledanja videa gde korisnik daje ocenu kao znak zadovoljstva rezolucijom u datom kontekstu) koja je potrebna da bi se obradile preferencije pojedinačnih korisnika, što opet doprinosi uštedi energije, kao i poboljšanju korisničkog iskustva. Ono što je najvažnije, ovo istraživanje ima za cilj da stvori i jasan generički okvir koji će voditi razvoj budućih sličnih aplikacija.

2. PREGLED AKTUELNOG STANJA U OBLASTI

U ovom poglavlju predstavljene su postojeće literature, čije teme su slične temi koja je opisana u ovom radu. Spomenute literature su razmatrane u cilju boljeg razumevanja *experience sampling*-a i *approximate computing*-a. Nakon dužeg istraživanja velikog broja javno dostupnih literatura, detaljnije je istražena nekolicina istih, gde svaki od radova na svoj način ima značajan uticaj u razumevanju i dubljem razmišljanju o načinu rešavanja problema na koje se susretalo pri izradi.

2.1. ESM

Experience sampling metoda postoji već duže vreme i odnosi se na procenu psihičkog stanja korisnika u određenim situacijama. Danas, pored automatskog prikupljanja podataka, mobilni telefoni omogućuju interakciju sa korisnikom tokom čitavog dana, odnosno, u prirodnom okruženju korisnika, umesto prisilnog okruženja u kojem se ispitanici često ponašaju drugačije. [2] Međutim, postoje neka ograničenja, kao što je sprovođenje ankete bez ometanja načina života korisnika ili pitanje kako odlučiti koji je trenutak relevantan ako korisnici žive nepredvidivim životom (mnogo se kreću) ili koje kriterijume koristiti pri izboru ispitanika (sada kada gotovo svi koji poseduju mobilne uređaje mogu da učestvuju). Tehničke poteškoće u vezi sa ESM-om primarno predstavlja trajanje baterije. Da bi smanjili potrošnju, programeri treba da su u mogućnosti da definišu koliko često treba da se vrši uzorkovanje podataka. U studiji [3] pokazano je da bi idealan broj učesnika bio 19, studija ne bi trebalo da traje duže od mesec dana i predlaže da se anketa povuče nakon par minuta, jer ako se povuče kasnije, kontekst će biti drugačiji i odgovor neće biti relevantan. Prilikom izrade ankete, fokus treba da bude i na tome kada je prikazati korisniku. S jedne strane, od istraživača se traži da prikupi što bogatije podatke, a sa druge strane, da ne smeta korisniku. Loša strana *esensing*-a predstavlja značajnu potrošnju energije zbog stalne potrebe za praćenjem trenutnog konteksta u kojem se korisnik nalazi. Štaviše, samo snimanje konteksta u kojem se korisnik nalazi možda neće biti dovoljno jer u stvarnom životu mnogo više može uticati na mišljenje korisnika.

2.2 Approximate computing

Široka upotreba mobilnog računarstva nam omogućava da proverimo kako se menja zadovoljstvo korisnika dobijenim rezultatima sa promenom konteksta. Brojni aspekti situacije i okruženja mogu uticati na percepciju korisnika o rezultatu. Dakle, smatramo da je „kontekst“

složen termin, čiji pogled dobijamo preko senzora ugrađenih u mobilni uređaj.

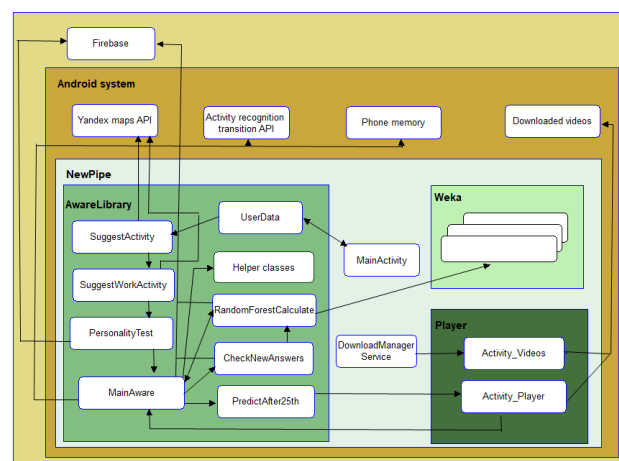
Approximate computing, odnosno približno računanje predstavlja tehnologiju koja žrtvuje tačnost rezultata zarad energetske efikasnosti. Koristeći ovaj metod, u mogućnosti smo da korisniku damo minimum koji će zadovoljiti njegove potrebe u cilju uštede energije. Međutim, do danas nema dovoljno podrške za pisanje i modelovanje *approximate* programa na mobilnim platformama, što usporava ukupan napredak. Eksperimenti su pokazali značajne uštede resursa, na primer, utrošuću energetska efikasnost sa aproksimacijama zasnovanim na neuronskim mrežama [4], ili 2,5 puta veću brzinu kada se određeni obrasci zamene *approximate* kodom. Percepcija korisnika o rezultatu se menja sa okruženjem i situacijom u kojoj se nalazi i stoga kontekst treba predstaviti kao veoma složen pojam. Uz pomoć ESM-a ćemo pitati korisnika koliko je zadovoljan sadržajem koji se prikazuje u određenom kontekstu, dok nam mašinsko učenje pomaže da povežemo korisnika i situaciju u kojoj se nalazi kako bismo modelirali promenu očekivanja kvaliteta rezultata u različitim kontekstima [5].

3. KORIŠĆENE TEHNOLOGIJE

Za implementaciju okvira korišćen je *Android Studio* [6], programski jezik *Java* [7]. Važne komponente projekta predstavljaju *AWARE* [8] biblioteka, koja je korišćena za praćenje senzora tokom sesije, kao i *WEKA* [9] biblioteka pomoću koje implementiramo *random forest* [10] algoritam. Za sugerisanje i pretraživanje lokacije na kojoj korisnik živi ili radi korišćen je *Yandex* API [11]. Svi značajni podaci čuvaju se u *Firestore* [12] bazi podataka.

4. SPECIFIKACIJA APLIKACIJE

4.1. Android ESM sensing biblioteka



Slika 1 - Arhitektura aplikacije i veze između klasa

U svetlo plavoj boji je aplikacija *NewPipe*, koju smo uzeli kao osnovu. *AwareLibrary* okvir je označen zelenom bojom. Takođe, tamno zelenom bojom označen je *Player* koji koristimo za reprodukciju download-ovanih videa, a svetlo zelenom bojom je naznačena *Weka* biblioteka koju koristimo za implementaciju *random forest* algoritma. *AwareLibrary* okvir je zasnovan na aplikaciji otvorenog koda *NewPipe* [13]. Sve funkcionalnosti koje imple-

mentira *NewPipe* su i dalje dostupne korisniku. Jedina promena koju korisnik može da primeti odnosi se na deo sa preuzetim videima. S obzirom da *NewPipe* aplikacija ne pruža mogućnost preuzimanja videa različitih rezolucija, ovaj deo je morao biti odrađen preko modula *Player*. Kada korisnik otvori stranicu sa preuzetim videima, redirektovan je na aplikaciju koja prikazuje sve videe koji se nalaze u uređaju.

Dalje, iz ove aplikacije pozivamo praćenje senzora, čuvanje i obradu podataka koji su važni za ovaj rad, a predstavljaju deo *AwareLibrary* okvira.

Da bismo razumeli vezu između prikazivanja aplikacije na određeni način i konteksta u kojima je prikazana, napravili smo okvir koji povezuje navedeno. Koristeći prikupljanje podataka ugrađenih u naš okvir, izračunali smo zadovoljstvo korisnika određenim video zapisima koji su pušteni, a kasnije, uz korišćenje *active* i *reinforcement learning*-a, napravili smo odgovarajući model koji odgovara preferencijama svake osobe pojedinačno uz najmanju moguću količinu upita.

Kada se aplikacija otvori prvi put, tražimo od korisnika da unese svoju kućnu i adresu na kojoj radi, koja kasnije pomaže da se utvrdi lokacija od značaja. Ovo je obično važno jer se korisnik može osećati drugačije na različitim lokacijama što može uticati na njegova očekivanja. Nakon toga, korisnik treba da odgovori na deset kratkih pitanja u vezi procene ličnosti koja takođe mogu imati značajan psihološki uticaj na preferencije. Ukoliko aplikacija nije otvorena prvi put i u bazi već postoje podaci vezani za isti uređaj, u klasi *AlarmReminder* postavljamo *scheduler* koji svaki dan u ponoć setuje broj odgledanih videa dnevno na 0. U klasi *HelpClassReminder* postavljamo *scheduler* koji u 18h svaki dan proverava da li je korisnik odgledao minimum 5 videa do tad i ako nije prikazuje se notifikacija kako bi podsetili korisnika da pogleda videe. Nakon toga, prikazuje se početna strana *NewPipe* aplikacije.

Nakon toga, sam programer, koristeći naš okvir, može da pozove jednostavnu funkciju gde označava koje senzore želi da uključi tokom koje aktivnosti i na koje parametre želi da se fokusira i na osnovu toga izgradi model. Funkciju poziva bilo gde u svom programu, gde smatra da je važno započeti prikupljanje podataka o kontekstu. U ovom primeru, uključeni su svi mogući senzori na početku gledanja videa.

Pri pozivu ove funkcije proverava se da li je korisnik odgledao više od 25 videa ukupno. Ako je pregledao, na osnovu prikupljenih podataka računamo koji kvalitet videa bi najviše odgovarao datom kontekstu i iz modula *Player* pozivamo odgovarajući video odgovarajuće rezolucije.

Ukoliko korisnik još nije pregledao 25 videa, na kraju svakog gledanja, poziva se druga funkcija koja zaustavlja prikupljanje podataka i pokreće obradu istih. Nakon zaustavljanja senzora prikupljamo podatke iz *AWARE* biblioteke i izračunavamo za svaki senzor koja je vrednost bila najčešća.

Zatim, koristeći *random forest* algoritam, predviđamo kako će korisnik oceniti ovu sesiju (u našem slučaju gledanje videa), nakon čega biramo jednu od četiri strategije *reinforcement* učenja koje su objašnjene u odeljku 4.3. U zavisnosti od rezultata algoritma, korisniku prikazujemo ili ne prikazujemo anketu.

Pored toga, da bismo motivisali korisnike, nakon određenog vremenskog perioda nagrađujemo ih otkrivanjem podataka o ličnosti, statistika o korišćenju aplikacije itd.

4.2. Razumevanje percepcije približnog mobilnog računarstva (approximate computing)

Approximate computing je računarska paradigma koja omogućava programerima da kompromituju rezultat računanja u cilju uštede energije. *Approximate computing* za mobilne uređaje uveden je sa mobilnim telefonima i razlikuje se u pogledu konteksta korišćenja koji se menja u toku upotrebe i čini predviđanje zadovoljstva još složenijim.

Imajući u vidu da na korisnikovo mišljenje utiču i karakteristike njegove ličnosti, njih lako računamo pomoću već definisanih pitanja za test ličnosti: Big Five Inventori. BFI-10 je izmišljen za kontekst sa ograničenim vremenom, sadrži samo 10 kratkih pitanja koja mogu brzo odrediti kakva je korisnik osoba. Pošto na zadovoljstvo korisnika značajno utiču lokacija na kojoj se korisnik nalazi, trajanje baterije, aktivnost korisnika i doba dana (osvetljenje), uzećemo u obzir ta četiri senzora.

Programer ima mogućnost programskog pokretanja i zaustavljanja senzora jednostavnim pozivanjem funkcije. Pošto se češće dešava da se kontekst (aktivnost, lokacija) promeni tokom jedne sesije gledanja, na kraju nje izračunavamo najčešću lokaciju/aktivnost/prosečno trajanje baterije/prosečno osvetljenje. Zatim, koristeći *random forest* algoritam, izračunavamo verovatnoću da će korisnik biti zadovoljan u određenom kontekstu, gledajući video sa određenim kvalitetom.

4.3. Inteligentno ispitivanje i modelovanje korisničkih očekivanja

Da bismo predvideli zadovoljstvo korisnika određenim parametrima, razvili smo model koji omogućava iterativno učenje preferencija korisnika u nekom kontekstu.

Aktivno učenje predstavlja strategiju u kojoj algoritam mašinskog učenja postiže veću tačnost ako je u mogućnosti da izabere iz kojih podataka će učiti. U našem našem slučaju, ono može smanjiti broj anketa koje se šalju korisniku i povećati zadovoljstvo korisnika jer bi ga prekidao samo kada njegov odgovor može značajno poboljšati kvalitet modela.

Da bismo odredili da li da korisniku pošaljemo anketu ili ne, razvili smo četiri različite strategije aktivnog učenja, kao i okvir koji pomaže u određivanju kako se približiti strategiji koja najbolje odgovara svakom korisniku. Počevši od *random forest* algoritma, izračunavamo pouzdanost da će korisnik dati video oceniti određenom ocenom (1-5). Posle svakog izračunavanja primenjujemo jednu od strategija aktivnog učenja:

- Margin of confidence*: odlučuje da pita korisnika ako razlika između prva dva najpouzdanija predviđanja padne ispod 20%.

- Random sampling*: nasumično bira da li da pita korisnika ili ne.

- Random sampling* zasnovan na vrednosti fizičke aktivnosti korisnika: pošto aktivnost korisnika predstavlja ključni parametar, u ovoj strategiji pitamo korisnika da li je u vozilu ili hoda. Imajući u vidu da je najčešće stanje korisnika mirovanje, prioritet dajemo situacijama u kojima korisnik nije miran.

•*Least confidence sampling*: traži od korisnika da oceni kvalitet videa ako najpouzdanije predviđanje padne ispod 0,5.

Da bismo balansirali između ova četiri, a ne samo fokusirali se na onu koja izgleda kao strategija koja najbolje funkcioniše, koristimo reinforcement algoritam učenja koji se naziva *upper confidence bound* (UCB). Osnovna jednačina UCB-a je sledeća:

$$A_t = \operatorname{argmax} (Q_t(a) + c \ln(t) N_t(a))$$

Za svaku od *active learning* strategija izračunavamo $A(t)$. Prvi deo jednačine predstavlja eksploatacioni deo. $N_t(a)$ je trenutna procena nagrade za akciju a predstavlja prosečnu vrednost prethodnih nagrada. Ako uzmemo u obzir samo ovaj deo jednačine, on će uvek birati akciju sa najvećom nagradom. Drugi deo jednačine predstavlja istraživanje mogućnosti. Ako je broj pokušaja akcije a mali, posledično će nesigurnost u rezultat biti velika, što će činiti verovatnije da će ova radnja biti odabrana.

Nakon odabira strategije i odluke da li da pitamo korisnika ili ne, ako je odluka bila pozitivna, a korisnikov odgovor promenio predviđanje u pravu klasu, nagradujemo akciju jednim bodom i razlikom između poslednjeg i novog predviđanja. Ako predviđanje ostane isto, nagrada će biti jedan poen. S druge strane, ako je odlučeno da anketu ne prikažemo korisniku, nakon trećeg puta nepitanja, korisnik će dobiti upitnik i ako je odgovor drugačiji od prethodno predviđenog, akcija će dobiti negativne poene, međutim ako je odgovor isti kao što je predviđeno, dobiće jedan bod. U početku će UCB izabrati svaku akciju jednom, pošto u bazi nema prethodnih podataka.

5. OPIS APLIKATIVNOG REŠENJA

U ovom poglavlju opisano je aplikativno rešenje koje obuhvata sve funkcionalnosti opisane u prethodnim poglavljima, koristeći delove šeme baze podataka namenjene za čuvanje podataka o korisnicima, podešavanjima naloga, vezama između korisnika, fotografijama i komentarima. Aplikativno rešenje je zamišljeno kao veb aplikacija i implementirano je pomoću tehnologija koje su prethodno navedene. Svaka funkcionalnost je detaljno opisana, navodeći ulogu korisnika za koju je namenjena kao i postupak koji je neophodno uraditi za izvršavanje iste. Prilikom popunjavanja formi, navedeni su koraci koje je neophodno ispratiti, tako što je naglašeno koje informacije treba uneti, kao i koje dugme kliknuti za potvrđivanje željene radnje.

6. ZAKLJUČAK

Osnovni zadatak ovog rada je bio razvoj i demonstracija Android okvira za prikupljanje i obradu informacija o korisničkom iskustvu. Osnovna ideja je bila smanjenje potrošnje energije prilikom korišćenja mobilnog telefona, odnosno, optimizacija delova aplikacija u skladu sa kontekstom u kom se korisnik nalazi.

Za demonstraciju ovog okvira, korišćena je aplikacija nalik YouTube-u koja podržava funkcionalnosti vezane za video zapise.

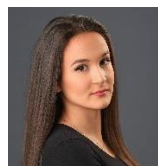
Dosadašnji rad obuhvata beleženje podataka iz senzora mobilnog telefona u toku korišćenja aplikacije, obradu tih podataka i predlaganje korisničkog zadovoljstva različitim parametrima u različitim kontekstima.

Aplikacija je razvijana u trenutno aktuelnim tehnologijama i ima dobru osnovu za dalji razvoj.

7. LITERATURA

- [1] Joseph A Paradiso and Thad Starner. 2005. Energy scavenging for mobile and wireless electronics. *IEEE Pervasive computing* 4, 1 (2005), 18–27
- [2] https://link.springer.com/chapter/10.1007/978-3-319-31413-6_8
- [3] Abdesslem et al. 2010; Froehlich et al. 2007
- [4] <http://lrss.fri.uni-lj.si/Veljko/docs/Pejovic18AMC.pdf>
- [5] <https://ieeexplore.ieee.org/abstract/document/7092486>
- [6] file:///C:/Users/lunaz/OneDrive/Desktop/Master%20rad/haberl_silva_pmf_2017_diplo_sveuc.pdf
- [7] https://www.theseus.fi/bitstream/handle/10024/133782/Lyytinen_Jere.pdf?sequence=1&isAllowed=y
- [8] <https://awareframework.com/what-is-aware/>
- [9] [https://en.wikipedia.org/wiki/Weka_\(machine_learning\)](https://en.wikipedia.org/wiki/Weka_(machine_learning))
- [10] <https://yandex.com/dev/maps/mapsapi/?from=mapsapi>
- [11] https://link.springer.com/chapter/10.1007/978-1-4842-2943-9_3
- [12] https://www.researchgate.net/profile/Chunnu-Khawas/publication/325791990_Application_of_Firebase_in_Android_App_DevelopmentA_Study/links/5bab55ed45851574f7e6801/Application-of-Firebase-in-Android-App-Development-A-Study.pdf
- [13] <https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm#:~:text=A%20Random%20Forest%20Algorithm%20is,more%20it%20will%20be%20robust.>

Kratka biografija:



Luna Živković, rođena 14.02.1999. u Novom Sadu. Završila osnovne akademske studije na Fakultetu tehničkih nauka u Novom Sadu, smer Računarstvo i automatika.

Kontakt: lunazivkovic@gmail.com



Milan Vidaković rođen je u Novom Sadu 1971. Doktorirao je na Fakultetu tehničkih nauka 2003. god, a od 2014. je u zvanju redovni profesor iz oblasti Primenjene računarske nauke i informatika.