

Uporaba dnevniških zapisov v namen profiliranja obremenitev spletne aplikacije

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

Izvešček

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

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

Using Log Records for Web Application Load Profiling

Abstract

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

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

1 UVOD

Iz analitičnega poročila družbe Sitefy LLP [1] je razvidno, da v letu 2024 je na internetu registriranih več kot 193 milijonov aktivnih spletnih mest. Spletne aplikacije so še vedno prevladujoča vrsta aplikacij, saj je do njih mogoče dostopati s katere koli naprave, povezane z internetom, vključno z računalniki, tabličnimi računalniki in pametnimi telefoni, ter z različnimi operacijskimi sistemi brez potrebe po prilagajanju. Takšne aplikacije zahtevajo

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

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

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

Glavne vrste testiranja zmogljivosti vključujejo [5]:

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

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

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

Napačna izbira števila sočasnih uporabnikov in napačni scenariji uporabe funkcij aplikacije lahko ne dajo popolne slike o zmogljivosti ali pa preobremenijo sistem preko realnih pogojev. Testiranje zmogljivosti aplikacij se lahko izvede pred izdajo različice programske opreme ali po njej. Za testiranje zmogljivosti pred izdajo nove različice programske opreme (z novimi lastnostmi in funkcionalnostmi) se lahko delovna obremenitev aplikacije določi na podlagi rezultatov pregleda mnenja uporabnikov in strokovnjakov ter rezultatov uporabe beta

različice aplikacije s strani skupine reprezentativnih uporabnikov. Testiranje zmogljivosti po izdaji nove različice programske opreme je mogoče izvesti s posodobljenimi ključnimi scenariji uporabe aplikacije, ki temeljijo na podatkih iz dnevniških zapisov spletnega strežnika in odražajo dejansko uporabo aplikacije (dejansko pogostost zahtevkov in zaporedje dejanj uporabnikov). To omogoča natančnejše modeliranje delovnih obremenitev in ocenjevanje zmogljivosti v pogojih, ki so blizu dejanski uporabi.

Predlagana metoda ni namenjena iskanju funkcionalnih napak v programski kodi, temveč določanju ali potrjevanju značilnosti hitrosti, razširljivosti in/ali stabilnosti testirane programske opreme, pri čemer se upoštevajo podatki o različici programske opreme, ki že deluje v realnih pogojih in je izpostavljena realni obremenitvi. Takšna rešitev omogoča sestavo natančnejšega profila obremenitev (profiliranje obremenitev) ob upoštevanju dejanskega obnašanja uporabnikov ter prepoznavanje težav, ki se morda ne pokažejo v fazi testiranja pred izdajo. Pri izbiri obdobja za analizo podatkov je treba upoštevati skladnost dnevniških zapisov s testno različico (funkcionalnostjo) aplikacije.

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

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

2 SORODNA DELA

2.1 Pristopi k obravnavi dnevniških datotek spletnega strežnika

Pridobivanje znanja iz dnevniških datotek – računalniških dnevnikov spletnega strežnika je pomembno področje raziskav, saj te datoteke vsebujejo dragocene informacije o uporabnikih, njihovih navadah in vzorcih obnašanja na spletnem mestu. Veliko število avtorjev in obstoječih programskih rešitev za obdelavo podatkov iz dnevniških datotek spletnega strežnika se osredotoča na statistično obdelavo teh podatkov [6], [11], [12]. To vključuje analizo obdobj največje aktivnosti uporabnikov, identifikacijo značilnosti uporabnikov in ugotavljanje najbolj priljubljenih strani. Na primer, s statistično analizo lahko ugotovimo, ob katerih urah je naše spletno mesto najbolj obiskano, katere strani so najbolj obiskane in kateri uporabniki so najpogostejši obiskovalci. Vendar pa ta pristop pogosto zanemarija odnose med posameznimi obiskanimi stranmi, kar pomeni, da ne upošteva, kako uporabniki »premikajo« po spletnem mestu in katere spletne strani pogosto obiskujejo v določenem zaporedju. Poleg statistične obdelave obstajajo tudi druge naloge za obdelavo podatkov iz dnevniških datotek spletnega strežnika, ki se osredotočajo na kompleksnejše analize. Ena izmed takih nalog je rudarjenje asociacijskih pravil, ki vključuje iskanje nizov povezanih strani [9], [11], [12], [13]. To pomeni, da lahko odkrijemo, katere strani so pogosto obiskane skupaj, kar nam lahko pomaga pri optimizaciji strukture spletnega mesta. Druga naloga je rudarjenje zaporednih vzorcev, kjer ugotovljamo urejene verige obiskov strani. Na ta način lahko razumemo, kako se uporabniki premikajo po spletnem mestu in katere poti

najpogosteje izberejo. Gručenje je še ena pomembna naloga, pri kateri združujemo uporabnike s podobnimi značilnostmi v skupine [11], [12], [14]. Na primer, lahko združimo uporabnike glede na njihove demografske podatke ali glede na strani, ki jih najpogosteje obiskujejo. Klasifikacija je podobna naloga, vendar se osredotoča na uvrščanje uporabnikov v vnaprej določene skupine na podlagi njihovih značilnosti. Na primer, uporabnike lahko uvrstimo v skupine glede na njihove nakupovalne navade ali glede na to, kako pogosto obiskujejo spletno mesto.

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

Za našo raziskavo je še posebej zanimiv koncept rudarjenja spletne uporabe, ki pomaga pridobiti zanimive vzorce iz dnevniških zapisov dostopa do spletnega strežnika [10]. Ta pristop omogoča razvrščanje uporabniških zahtev v skupine brez človeškega sodelovanja na podlagi merila podobnosti. Na primer, lahko avtomatsko združimo uporabnike, ki pogosto obiskujejo določene strani ali ki imajo podobne vzorce obnašanja na spletnem mestu. S tem pristopom lahko pridobimo dragocene vpoglede v obnašanje uporabnikov in optimiziramo spletno mesto glede na njihove potrebe.

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

Najbolj pomembna med njimi sta:

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

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

2.2 Podobna orodja

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

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

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

3 METODOLOGIJA

V predstavljeni raziskavi smo uporabili naslednje metode in orodja:

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

3.1 Raziskovalni pristop razvoja in načrtovanja

Metodologija načrtovanja in razvoja je učinkovito orodje za reševanje kompleksnih problemov, kot je »profiliranje obremenitev spletne aplikacije na podlagi dnevniških zapisov spletnega strežnika«. Osredotoča se na ustvarjanje in vrednotenje artefaktov, ki obravnavajo ugotovljene probleme ter zagotavljajo znanstvene prispevke in praktične koristi [22], [24], [25].

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

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

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

3.2 CRISP-DM metodologija

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

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

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

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

Četrta faza je modeliranje. Tu ustvarimo in preizkusimo različni modeli za analizo dnevniških zapisov. Za našo nalogo so to modeli za gručenje. Algoritme za gručenje lahko uporabimo za prepoznavanje skupin uporabnikov s podobnim obnašanjem.

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

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

3.3 Metoda za podporo profiliranju obremenitev

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

1. Izbira časovnega obdobja. V prvi fazi določimo časovno obdobje, za katero se podatki analizirajo. Pomembno je izbrati reprezentativno obdobje ob upoštevanju največjih obremenitev sistema, tako da rezultat odraža najbolj kritične trenutke delovanja aplikacije.
2. Čiščenje podatkov. Ta korak vključuje odstranitev nepravilnih ali nepopolnih zapisov, ki vsebujejo informacije, ki bi lahko popačile rezultate nadaljnjih analiz.
3. Združevanje podatkov po uporabnikih. To omogoča sistematizacijo dejanj posameznega uporabnika v okviru analize njegovih interakcij s spletno aplikacijo.



Slika 1: Postopek analize podatkov in oblikovanja profilov obremenitev.

4. Združevanje podatkov po sejah. Ta pristop omogoča analizo dejanj vsakega uporabnika v okviru ene seje, kar je potrebno za natančno modeliranje obnašanja uporabnikov.
5. Vektorizacija zahtevkov. V tem koraku pretvorimo uporabniške zahteve v numerične vektorje za nadaljnjo analizo. Ta korak je ključnega pomena za nadaljnje gručenje in analizo podatkov.
6. Gručenje IP-naslovov. Za nadaljnjo analizo razdelimo IP-naslovi uporabnikov v gruče, kar omogoča prepoznavanje podobnih skupin uporabnikov in analizo njihovega obnašanja.
7. Označevanje sej. Vsako sejo označimo z lastnostmi, ki omogočajo natančnejšo razlago obnašanja uporabnikov znotraj posameznih gruč.
8. Združitev in vektorizacija zahtevkov. Uporabniške zahteve iz različnih sej združimo in pretvorimo v vektorje za natančnejšo analizo vzorcev obnašanja v preučevanem časovnem obdobju.
9. Gručenje sej. S tehnikami strojnega učenja seje uporabnikov razvrstimo v skupine na podlagi podobnih značilnosti in obnašanja. To omogoča prepoznavanje glavnih vzorcev interakcije z aplikacijo.
10. Interpretacija vsebine gruč. Na podlagi pridobljenih gruč interpretiramo njihovo vsebino, da določimo značilni vzorci obnašanja uporabnikov in vrste sej.
11. Izračun deleža scenarijev za profile. Zadnji korak vključuje izračun deleža različnih scenarijev uporabe spletnih aplikacij za vsak profil obremenitev. To omogoča oblikovanje natančnih profilov obremenitev za testiranje zmogljivosti programske opreme.

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

4 REZULTATI

4.1 Razumevanje problema in podatkov

Podatke smo pridobili na spletnem mestu Kaggle (<https://www.kaggle.com/datasets/eliasdabbas/web-server-access-logs>) [29]. To je nabor 3,3 GB zapisov iz spletne trgovine – spletnega strežnika zanbil.ir. Struktura datoteke ustreza formatu CLF (angl. Common Log Format) [30]. Ta nabor podatkov je bil izbran zaradi njegove dostopnosti in razpoložljivosti podatkov o dejanskih zahtevah na spletnem mestu. Vendar se lahko metoda, predlagana v tem članku, uporabi tudi za druga spletna mesta, saj so ključni analizirani parametri, kot so IP-naslov klienta, datum in čas zahteve, URL in koda stanja odgovora, skupni večini spletnih mest ne glede na vsebino ali posebnosti spletnega mesta.

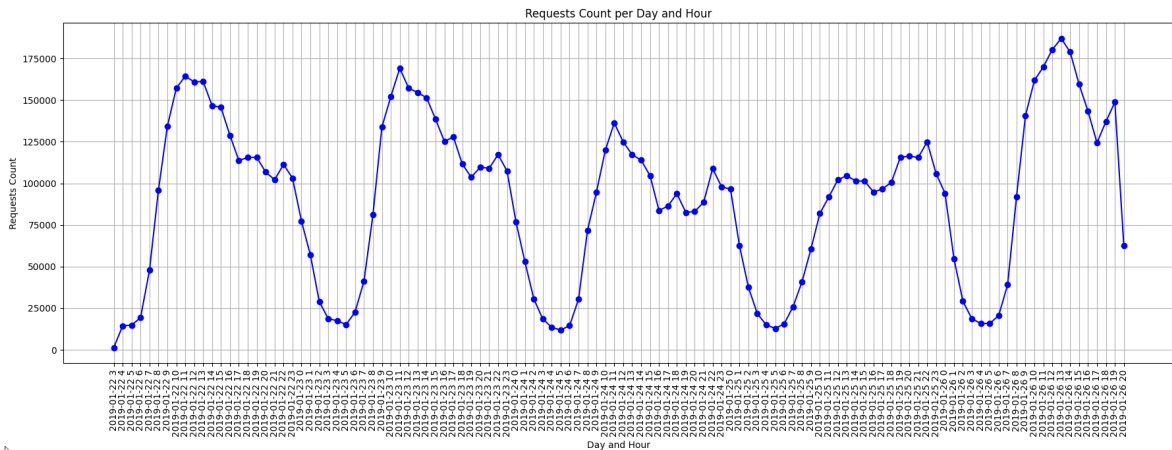
Podatki vsebujejo 10 atributov in 10365152 zapisov v obliki CSV-datoteke. Znak presledka se uporablja kot ločilo, pri kompleksnih vrednostih nizov pa se upošteva njihovo okvirjanje z narekovaji. Primer in struktura sta navedena v tabeli 1.

Tabela 1: Primer in struktura podatkov.

Parameter	Vrednost
remotehost	54.36.149.41
rfc931	-

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

Datoteka vsebuje podatke za obdobje od 22.01.2019, in sicer od 03:56:14 naprej, do 26.01.2019, in sicer do 20:29:13 (prikazano na sliki 2).



Slika 2: Število zahtev po datumu in času.

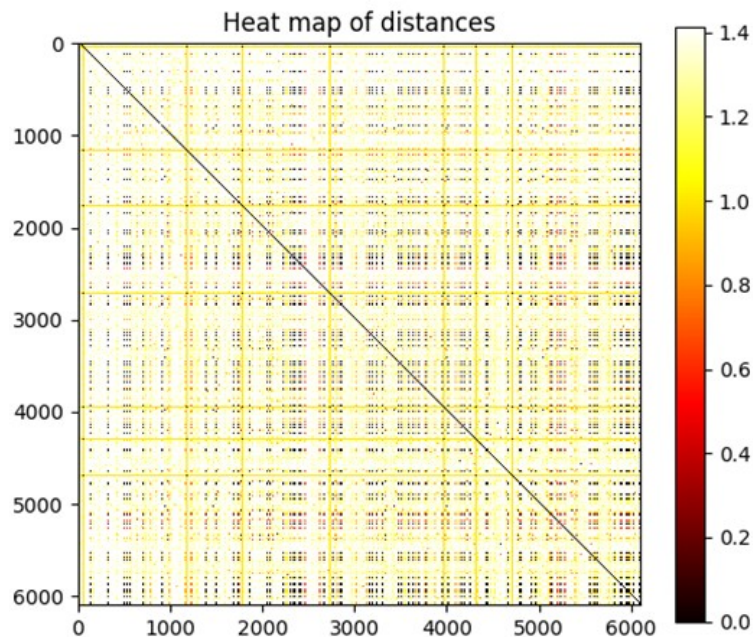
Iz slike 2 je razvidno, da se število uporabniških zahtev spreminja vsak dan na enak ciklični način. Opazimo jasne vzorce, ki kažejo na dnevne cikle v obiskanosti spletnega mesta, na primer, da se število zahtev znatno poveča ob določenih urah dneva, kot so dopoldanske ure, ko je aktivnost uporabnikov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

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

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

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

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



Slika 3: Toplotni zemljevid.

4.2 Priprava podatkov

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

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

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

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

mobilnih naprav, krajši zaradi višjih hitrosti dostopa in uporabe hitrih tehnologij, kot je AJAX. Predhodna analiza je pokazala, da nam zmanjšanje intervala na 20 minut omogoča ustrezno razvrstitev podatkov za analizo obremenitev.

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

4.3 Modeliranje in ocenjevanje pridobljenih rezultatov

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

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

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

Opazovanje spreminjanja porazdelitve podatkov v gruče je pokazalo, da ko je dosežena razdelitev podatkov v določeno število gruči N , nadaljnje povečevanje števila gruči ne vpliva bistveno na kakovost rezultatov gručenja, saj nove gruče vsebujejo tako majhno količino podatkov, da jih lahko zanemarimo brez večje izgube informacijske sestavine. Ta pristop pomaga najti razpon vztrajnosti k , pri katerem nadaljnje povečanje števila gruči bistveno ne zmanjša vztrajnosti, kar kaže na optimalno število gruči za ta problem. Sprememba vztrajnosti je bila vzeta kot osnovna metrika za spremembo v izgubi informacij. Vztrajnost k je metrika, ki meri vsoto kvadratov razdalj od vsake točke do najbližjega centroida. Uporablja se za ocenjevanje kakovosti gručenja, saj pokaže, kako kompaktne so gruče. Celoten postopek lahko opišemo na naslednji način:

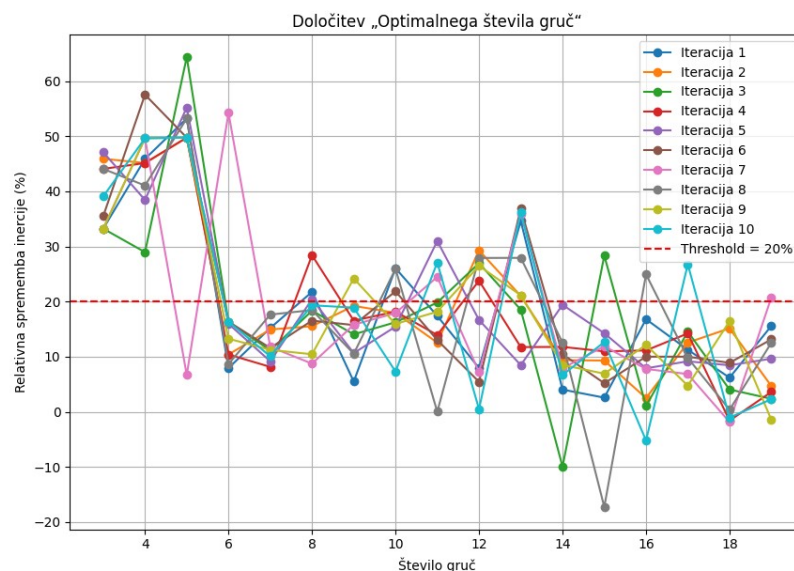
1. Opredelitev razpona vrednosti k , za katere bo izvedeno gručenje.
2. Izvedba gručenja in ohranjanje vztrajnosti: za vsako vrednost k ustvarimo in usposobimo model `K-Means`. Vztrajnost se nato shrani na seznam vztrajnosti.
3. Izračun relativne spremembe vztrajnosti: za vsako vrednost k izračunamo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

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

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

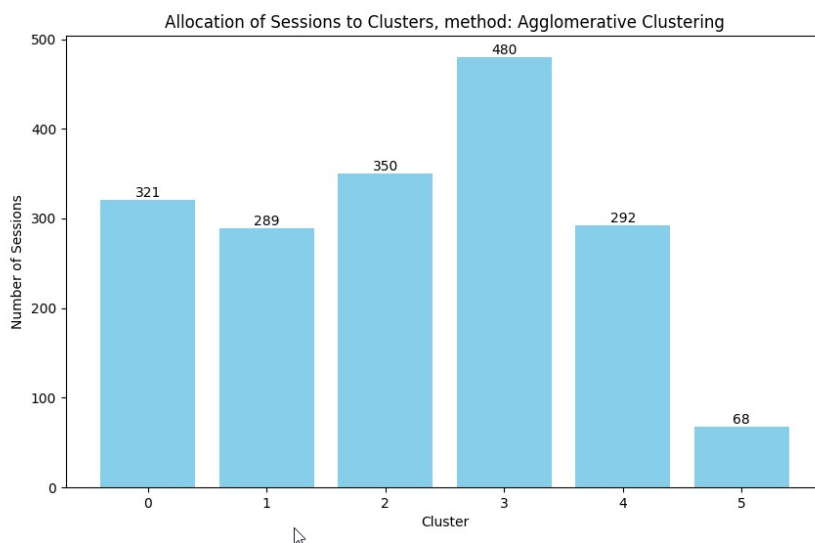
V naslednjem koraku smo analizirali značilnosti sej, ki so razvrščene v isto gručo. Uporabniki aktivno uporabljajo mobilne različice strani in optimizirane strani AMP (angl. Accelerated Mobile Pages) za hiter dostop do vsebine ter iščejo določene izdelke, kar počnejo z iskalnikom. AMP je Googlova pobuda,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

Na sliki 4 prikazujemo rezultate določanja optimalnega števila gruč na podlagi relativne spremembe vztrajnosti za drugo skupino uporabnikov s spreminjanjem števila gruč od 2 do 20. Izračun smo ponovili desetkrat, da bi dobili aritmetično sredino optimalnega števila gruč glede na metriko »relativna sprememba vztrajnosti«. Os X prikazuje število gruč, os Y pa spremembo vztrajnosti v odstotkih. Posamezne krivulje (slika 4) ustrezajo eni od 10 iteracij algoritma gručenja, rdeča črtkana premica pa označuje prag 20%, pri katerem postane nadaljnje povečevanje števila gruč neučinkovito. Videti je, da se pri številu gruč okoli 6 vrednost vztrajnosti znatno spremeni, čemur sledi stabilizacija pod 20-odstotnim pragom, kar kaže na optimalno število gruč za ta problem.



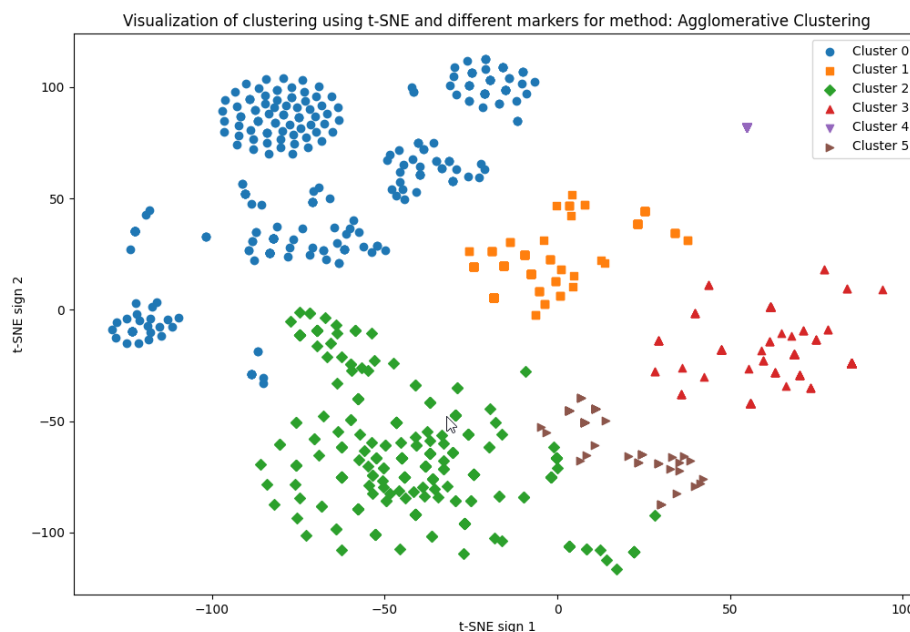
Slika 4: Izračun povprečja optimalnega števila gruč.

Rezultati porazdelitve sej znotraj gruč v drugi skupini uporabnikov so prikazani na sliki 5 in sliki 6. Aglomerativno gručenje je dobro razdelilo podatke v več homogenih skupin.



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

Koeficient silhuete 0,7659 in indeks Kalinskega-Harabasha 2771,7697 kažeta na jasno ločitev gruč in visoko homogenost znotraj gruč. Tudi Davis-Bouldinov indeks 0,5800 potrjuje kakovostno gručenje z nizko stopnjo prekrivanja med gručami. Nizka vrednost Dunnovega indeksa (0,1152) pa kaže na morebitne težave z gostoto in ločevanjem gruč.



Slika 6: Vizualizacija znotraj gruč v drugi skupini uporabnikov.

V nadaljevanju opisujemo glavne vzorce in značilnosti sej za vsako nastalo gručo.

Gruča 0:

- skoraj vse seje vključujejo zahteve na stran '/site/enamad', ki je verjetno ključna stran;
- uporabniki pogosto obiskujejo kategorije, kot so gospodinjski aparati, avdio in video oprema, mobilni telefoni;

- številne seje vključujejo zahteve za filtriranje izdelkov in uporabo AJAX zahtev;
- nekatere seje vključujejo ogled podrobnosti o izdelku, dodajanje izdelkov v košarico in oddajo naročila;
- v več sejah uporabniki uporabljajo iskanje in samodejno dopolnjevanje.

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

Gruča 1:

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

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

Gruča 2:

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

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

Gruča 3:

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

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

Gruča 4:

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

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

Gruča 5:

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

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

5 DISKUSIJA

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

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

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

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

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

Rezultati gručenja sej kažejo na dobro kakovost gručenja: koeficient Silhouette 0,7659 in indeks Kalinskega-Harabasha 2771,7697 kažeta na jasno ločitev gruč in visoko homogenost znotraj gruče. Davies-Bouldinov indeks 0,5800 potrjuje kakovostno gručenje z nizko stopnjo prekrivanja med gručami. Nizka vrednost Dunnovega indeksa (0,1152) pa kaže na morebitne težave z gostoto in ločevanjem gruč. Glede na dobljene izračune je najboljšo število gruč za delitev šest. Vizualizacija na sliki 6 prikazuje dobro porazdelitev sej v gruče, ki potrjujejo jasne meje med njimi.

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

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

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

6 ZAKLJUČEK

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

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

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

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

ZAHVALA

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

Literatura

- [1] NJ, „How Many Websites Are There in the World? (2024)“, Siteefy. Pridobljeno: 15. julij 2024. [Na spletu]. Dostopno na: <https://siteefy.com/how-many-websites-are-there/>
- [2] E. Bocchi, L. De Cicco, in D. Rossi, „Measuring the Quality of Experience of Web users“, v *Proceedings of the 2016 workshop on QoE-based Analysis and Management of Data Communication Networks*, v Internet-QoE '16. New York, NY, USA: Association for Computing Machinery, avg. 2016, str. 37–42. doi: 10.1145/2940136.2940138.
- [3] *ISO/IEC 25010:2023(en), Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuARE) — Product quality model*, International standard 25010:2023, november 2023. Pridobljeno: 15. julij 2024. [Na spletu]. Dostopno na: <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:25010:ed-2:v1:en>
- [4] Gartner Peer Community, „Automated Software Testing Adoption and Trends“, Gartner, One-Minute Insight Report, jun. 2023. Pridobljeno: 14. september 2024. [Na spletu]. Dostopno na: <https://www.gartner.com/peer-community/home>
- [5] Microsoft Corporation, *Performance Testing Guidance for Web Applications*, 1st edition. United States? Microsoft Press, 2007.
- [6] K. . Dharmarajan in M. A. Dorairangaswamy, „Discovering User Pattern Analysis from Web Log Data using Weblog Expert“, *Indian Journal of Science and Technology*, let. 9, št. 42, nov. 2016, doi: 10.17485/ijst/2016/v9i42/99300.
- [7] H. Karthakula in N. Gayatri, „Customer Behaviour Prediction Using Web Usage Mining“, *Journal of Cardiovascular Disease Research*, let. 12, št. 03, 2021.
- [8] A. Neetu in H. Saba, „ijcsit2012030251.pdf“, *International Journal of Computer Science and Information Technologies*, let. 3 (2), str. 3536–3539, mar. 2012.
- [9] O. Prakash in A. Jaya, „Analyzing and predicting user behavior pattern from weblogs“, *International Journal of Applied Engineering Research*, let. 11, str. 6278–6283, jan. 2016.
- [10] G. Sudhamathy, S. B. Illango, in J. C. Venkateswaran, „A Comparative Study on Web Log Clustering Approaches“, v *GLOBAL TRENDS IN COMPUTING AND COMMUNICATION SYSTEMS, PT 1*, P. V. Krishna, M. R. Babu, in E. Ariwa, Ur., v Communications in Computer and Information Science, vol. 269. Berlin: Springer-Verlag Berlin, 2012, str. 402-+. Pridobljeno: 11. november 2023. [Na spletu]. Dostopno na: <https://www.webofscience.com/wos/woscc/summary/b7f7fef5-90cb-436e-a3be-ed08b07844b5-b366957b/relevance/1>
- [11] L. K. Joshila Grace, V. Maheswari, in D. Nagamalai, „Analysis of Web Logs And Web User In Web Mining“, *IJNSA*, let. 3, št. 1, str. 99–110, jan. 2011, doi: 10.5121/ijnsa.2011.3107.
- [12] J. Srivastava, R. Cooley, M. Deshpande, in P.-N. Tan, „Web usage mining: discovery and applications of usage patterns from Web data“, *SIGKDD Explor. Newsl.*, let. 1, št. 2, str. 12–23, jan. 2000, doi: 10.1145/846183.846188.
- [13] M. Grcar, „USER PROFILING: WEB USAGE MINING“, predstavljeno na Conference on Data Mining and Warehouses (SiKDD 2004), Ljubljana, Slovenia: J.Stefan Institute, okt. 2004. [Na spletu]. Dostopno na: <https://aile3.ijs.si/dunja/SiKDD2004/Papers/MihaGrcar-WebUsageMining.pdf>
- [14] M. H. A. Elhiber in A. Abraham, „Access Patterns in Web Log Data: A Review“, *Journal of Network and Innovative Computing*, let. Vol. 1 (2013), str. 348–355, okt. 2013.
- [15] Z. Pabarskaite in A. Raudys, „A process of knowledge discovery from web log data: Systematization and critical review“, *J. Intell. Inf. Syst.*, let. 28, št. 1, str. 79–104, feb. 2007, doi: 10.1007/s10844-006-0004-1.
- [16] Y. K. Woon, W. K. Ng, X. Li, in W. F. Lu, „Efficient web log mining for product development“, v *2003 INTERNATIONAL CONFERENCE ON CYBERWORLDS, PROCEEDINGS*, T. L. Kunii, S. H. Soon, in A. Sourin, Ur., Los Alamitos: IEEE Computer Soc, 2003, str. 294–301. doi: 10.1109/CYBER.2003.1253468.
- [17] Z. L. Yang, Y. T. Wang, in M. Kitsuregawa, „An effective system for mining web log“, v *FRONTIERS OF WWW RESEARCH AND DEVELOPMENT - APWEB 2006, PROCEEDINGS*,

- let. 3841, X. F. Zhou, J. Li, H. T. Shen, M. Kitsuregawa, in Y. Zhang, Ur., v *Lecture Notes in Computer Science*, vol. 3841. , Berlin: Springer-Verlag Berlin, 2006, str. 40–52. Pridobljeno: 11. november 2023. [Na spletu]. Dostopno na: <https://www.webofscience.com/wos/woscc/summary/73145dd1-9bcf-421e-a023-1cde96a73ed4-b3669701/relevance/1>
- [18] Z. X. Chen, A. W. C. Fu, in F. C. H. Tong, „Optimal algorithms for finding user access sessions from very large web logs“, *World Wide Web*, let. 6, št. 3, str. 259–279, sep. 2003, doi: 10.1023/A:1024606901978.
- [19] Z. Pabarskaite, „Implementing advanced cleaning and end-user interpretability technologies in Web log mining“, v *ITI 2002: PROCEEDINGS OF THE 24TH INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY INTERFACES*, V. Glavinica, V. H. Dobric, in D. Simic, Ur., Zagreb: Univ Zagreb, 2002, str. 109–113. Pridobljeno: 11. november 2023. [Na spletu]. Dostopno na: <https://www.webofscience.com/wos/woscc/summary/458315da-0901-4ce0-bdaf-3f388d9f1836-b3669ae5/relevance/1>
- [20] O. R. Zaiane, „Web Usage Mining for a Better Web-Based Learning Environment“, University of Alberta, Edmonton, Alberta, Canada, Technical report TR01-05, 2001. Pridobljeno: 8. junij 2024. [Na spletu]. Dostopno na: <https://era.library.ualberta.ca/items/0a182195-ce39-4b5d-a1c1-291ed91a0f36/view/336c2a34-b149-4d9d-95a1-ad78c08ee35c/TR01-05.pdf>
- [21] J. Lorenzo in J. Domenech, „A Tool for Web Usage Mining“, let. 4881, str. 704, dec. 2007, doi: 10.1007/978-3-540-77226-2_70.
- [22] A. Hevner *idr.*, „Design Science in Information Systems Research“, *Management Information Systems Quarterly*, let. 28, str. 75, mar. 2004.
- [23] U. Kannengiesser in J. S. Gero, „MODELLING THE DESIGN OF MODELS: AN EXAMPLE USING CRISP-DM“, *Proc. Des. Soc.*, let. 3, str. 2705–2714, jul. 2023, doi: 10.1017/pds.2023.271.
- [24] M. Kljajić Borštnar, *RAZISKOVANJE INFORMACIJSKIH SISTEMOV. UČBENIK. UNIVERZA V MARIBORU, FAKULTETA ZA ORGANIZACIJSKE VEDE*, 2021.
- [25] K. Peffers, T. Tuunanen, M. Rothenberger, in S. Chatterjee, „A design science research methodology for information systems research“, *Journal of Management Information Systems*, let. 24, str. 45–77, jan. 2007.
- [26] A. Hevner, J. vom Brocke, in A. Maedche, *Design Science Research. Cases*, 1. izd. v *Progress in IS*. Springer International Publishing, 2020. doi: 10.1007/978-3-030-46781-4_1.
- [27] J. Brzozowska, J. Pizoń, G. BAYTIKENOVA, A. Gola, A. ZAKIMOVA, in K. Piotrowska, „DATA ENGINEERING IN CRISP-DM PROCESS PRODUCTION DATA – CASE STUDY“, *Applied Computer Science*, let. 19, str. 83–95, sep. 2023, doi: 10.35784/acs-2023-26.
- [28] V. Plotnikova, M. Dumas, in F. Milani, „Adapting the CRISP-DM Data Mining Process: A Case Study in the Financial Services Domain“, v *Research Challenges in Information Science*, let. 415, S. Cherfi, A. Perini, in S. Nurcan, Ur., v *Lecture Notes in Business Information Processing*, vol. 415. , Cham: Springer International Publishing, 2021, str. 55–71. doi: 10.1007/978-3-030-75018-3_4.
- [29] E. Dabbas, „Web Server Access Logs“. <https://www.kaggle.com/datasets/eliasdabbas/web-server-access-logs>. Pridobljeno: 29. april 2024. [Csv]. Dostopno na: <https://www.kaggle.com/datasets/eliasdabbas/web-server-access-logs>
- [30] A. Foundation, „Log Files - Apache HTTP Server Version 2.4“. Pridobljeno: 29. april 2024. [Na spletu]. Dostopno na: <https://httpd.apache.org/docs/2.4/logs.html>
- [31] A. Amira, A. M. Huda, in B. Nagwa, „Automation of Performance Testing: A Review“, *IJICIS*, str. 1–16, dec. 2022, doi: 10.21608/ijicis.2022.161846.1219.
- [32] G. Din, I. Schieferdecker, in R. Petre, „Performance Test Design Process and Its Implementation Patterns for Multi-services Systems“, predstavljeno na *International Workshop on Formal Approaches to Software Testing*, Berlin, Heidelberg: Springer, jan. 2008, str. 135–152. doi: 10.1007/978-3-540-68524-1_11.
- [33] S. Pargaonkar, „A Comprehensive Review of Performance Testing Methodologies and Best Practices: Software Quality Engineering“, *International Journal of Science and Research (IJSR)*, let. 12, str. 2008–2014, nov. 2023, doi: 10.21275/SR23822111402.

[34] H. Sarojadevi, „Performance Testing: Methodologies and Tools“, *Journal of Information Engineering and Applications*, let. Vol 1, No.5, 2011, 2011.

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

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

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