



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



ЗБОРНИК РАДОВА ФАКУЛТЕТА ТЕХНИЧКИХ НАУКА

Едиција: Техничке науке - зборници

Година: XXXIX

Број: 9/2024

Нови Сад

Едиција: „Техничке науке – Зборници“

Година: XXXIX

Свеска: 9

Издавач: Факултет техничких наука Нови Сад

Главни и одговорни уредник: проф. др Срђан Колаковић, декан Факултета техничких наука у Новом Саду

Уредништво:

Проф. др Срђан Колаковић
Проф. др Александар Купусинац
Проф. др Борис Думнић
Проф. др Дарко Стефановић
Проф. др Себастиан Балоиш
Проф. др Дејан Лукић
Проф. др Јован Дорић
Проф. др Мирослав Кљајић
Проф. др Немања Тасић
Проф. др Дејан Убавин

Проф. др Милан Видаковић
Проф. др Мирјана Дамњановић
Проф. др Јелена Атанацковић Јеличић
Проф. др Игор Пешко
Проф. др Драган Јовановић
Проф. др Небојша Ралевић
Доц. др Сања Ожват
Проф. др Немања Кашиковић
Проф. др Теодор Атанацковић

Редакција:

Проф. др Дарко Стефановић, главни уредник
Проф. др Жељен Трповски, технички
уредник

Проф. др Драгољуб Новаковић
Проф. др Иван Пинђер
Бисерка Милетић

Језичка редакција:

Бисерка Милетић, лектор
Софија Рацков, коректор
Мр Марина Катић, преводилац

Савет за библиотечку и издавачку делатност ФТН,
проф. др Стеван Станковски, председник.

Штампа: ФТН – Графички центар ГРИД, Трг Доситеја Обрадовића 6, Нови Сад

CIP-Каталогизација у публикацији
Библиотека Матице српске, Нови Сад

378.9(497.113)(082)
62

ЗБОРНИК радова Факултета техничких наука / главни и одговорни уредник
Срђан Колаковић. – Год. 7, бр. 9 (1974)-1990/1991, бр.21/22 ; Год. 23, бр 1 (2008)-. – Нови Сад : Факултет
техничких наука, 1974-1991; 2008-. – илустр. ; 30 цм. –(Едиција: Техничке науке – зборници)

Месечно

ISSN 0350-428X

COBISS.SR-ID 58627591

ПРЕДГОВОР

Поштовани читаоци,

Пред вама је девета овогодишња свеска часописа „Зборник радова Факултета техничких наука“.

Часопис је покренут давне 1960. године, одмах по оснивању Машинског факултета у Новом Саду, као „Зборник радова Машинског факултета“, а први број је одштампан 1965. године. Након осам публикованих бројева у шест година, пратећи прерастање Машинског факултета у Факултет техничких наука, часопис мења назив у „Зборник радова Факултета техничких наука“ и 1974. године излази као број 9 (VII година). У том периоду у часопису се објављују научни и стручни радови, резултати истраживања професора, сарадника и студената ФТН-а, али и аутора ван ФТН-а, тако да часопис постаје значајно место презентације најновијих научних резултата и достигнућа. Од броја 17 (1986. год.), часопис почиње да излази искључиво на енглеском језику и добија поднаслов «Publications of the School of Engineering». Једна од последица нарастања материјалних проблема и несрећних догађаја на нашим просторима јесте и привремени прекид континуитета објављивања часописа двобројем/двогодишњаком 21/22, 1990/1991. год.

Друштво у коме живимо базирано је на знању. Оно претпоставља реорганизацију наставног процеса и увођење читавог низа нових струка, као и квалитетну организацију научног рада. Значајне промене у структури високог образовања, везане за имплементацију Болоњске декларације, усвајање нове и активне улоге студената у процесу образовања и њихово све шире укључивање у стручне и истраживачке пројекте, као и покретање нових мастер и докторских студија, доносе потребу да ови, веома значајни и вредни резултати, постану доступни академској и широј јавности. Оживљавање „Зборника радова Факултета техничких наука“, као јединственог форума за презентацију научних и стручних достигнућа, пре свега студената, обезбеђује услове за доступност ових резултата.

Због тога је Наставно-научно веће ФТН-а одлучило да, од новембра 2008. год. у облику пилот пројекта, а од фебруара 2009. год. као сталну активност, уведе презентацију најважнијих резултата свих мастер радова студената ФТН-а у облику кратког рада у „Зборнику радова Факултета техничких наука“.

Поред студената мастер студија, часопис је отворен и за студенте докторских студија, као и за прилоге аутора са ФТН или ван ФТН-а.

Зборник излази у два облика – електронском на веб сајту ФТН-а (www.ftn.uns.ac.rs) и штампаном, који је пред вама. Обе верзије публикују се сваки месец, у оквиру промоције дипломираних мастера.

У овом броју штампани су радови студената мастер студија, сада већ мастера, који су радове бранили у периоду од 16.11.2023. до 30.05.2024. год., а који се промовишу 21.09.2024. год. То су оригинални прилози студената са главним резултатима њихових мастер радова.

Известан број кандидата објавили су радове на некој од домаћих научних конференција или у неком од часописа. Њихови радови нису штампани у Зборнику радова.

Велик број дипломираних инжењера–мастера у овом периоду био је разлог што су радови поводом ове промоције подељени у две свеске.

У овој свесци, са редним бројем 9, објављени су радови из области:

- машинства,
- електротехнике и рачунарства,
- саобраћаја и
- мехатронике.

У свесци са редним бројем 10. објављени су радови из области:

- грађевинарства,
- архитектуре,
- инжењерског менаџмента,
- инжењерства заштите на раду и заштите животне средине.
- математике у техници,
- геодезије и геоматике,
- управљања ризиком од катастрофалних догађаја и пожара
- инжењерства информационих система,
- биомедицинског инжењерства,
- анимација у инжењерству и
- чистих енергетских технологија.

Уредништво се нада да ће и професори и сарадници ФТН-а и других институција наћи интерес да публикују своје резултате истраживања у облику регуларних радова у овом часопису. Ти радови ће бити објављивани на енглеском језику због пуне међународне видљивости и проходности презентованих резултата.

У плану је да часопис, својим редовним изласком и високим квалитетом, привуче пажњу и постане довољно препознатљив и цитиран да може да стане раме-уз-раме са водећим часописима и заслужи своје место на СЦИ листи, чиме ће значајно допринети да се оствари мото Факултета техничких наука:

„Високо место у друштву најбољих“

Уредништво

SADRŽAJ

STRANA

Radovi iz oblasti: Mašinstvo

1. Marko Zagorac,
UPOREDNA ANALIZA PRIMENE TOPLOTNIH PUMPI ZA POTREBE GREJANJA I PRIPREME
SANITARNE TOPLE VODE PORODIČNOG OBJEKTA 1117-1120
2. Srđan Terzin,
PROJEKAT VENTILACIONOG SISTEMA BIM METODOLOGIJOM ZA KOMERCIJALNI
OBJEKAT POVRŠINE 580 m² 1121-1124
3. Цвијетин Пејчић, Радомир Ђокић,
ПОДЛОГЕ ЗА ТЕХНИЧКУ КОНТРОЛУ ЕЛЕКТРИЧНИХ ЛИФТОВА СА СТАНОВИШТА
БЕЗБЕДНОСТИ И ЗАШТИТЕ НА РАДУ 1125-1128
4. Boško Vuković,
VEZA IZMEĐU TVRDOĆE I STEPENA DEFORMACIJE KOD ZAPREMINSKOG
DEFORMISANJA CILINDRIČNIM ALATOM 1129-1132
5. Балша Дачевић,
ИСТРАЖИВАЊЕ УТИЦАЈА МОДЕЛА ТРЕЊА НА КАРАКТЕРИСТИКЕ ПРОЦЕСА
КОМБИНОВАНОГ ИСТИСКИВАЊА ПУТЕМ НУМЕРИЧКИХ СИМУЛАЦИЈА 1133-1136
6. Петар Ивановић,
ИЗБОР И ПРОРАЧУН МАШИНСКЕ ОПРЕМЕ ЦРПНЕ СТАНИЦЕ ЗА ДРЕНАЖУ
ЕНЕРГЕТСКЕ ГАЛЕРИЈЕ УЗВОДНЕ ГЛАВЕ БРОДСКЕ ПРЕВОДНИЦЕ ХЕ ЂЕРДАП 1 1137-1140

Radovi iz oblasti: Elektrotehnika i računarstvo

1. Saša Taloši,
DECENTRALIZOVANO USAGLAŠAVANJE SISTEMSKIH SATOVA U ČVOROVIMA AD-HOK
BEŽIČNE MREŽE 1141-1144
2. Marija Čamparević,
KONCEPT NAPAJANJA SLOŽENIH TEHNOLOŠKIH CELINA NA PRIMERU GASNE
KOMPRESORSKE STANICE – VELIKA PLANA 1145-1148
3. Jelena Bojanić, Milan Vidaković,
RAZVOJ NPM PAKETA ZA EKSTRAKCIJU NAJFREKVENTNIJIH BOJA SA SLIKE 1149-1152

4. Лука Ступар, ИМПЛЕМЕНТАЦИЈА МЕХАНИЗМА АРБИТРАЖНОГ ТРГОВАЊА КРИПТОВАЛУТАМА У iOS МОБИЛНОЈ АПЛИКАЦИЈИ	1153-1156
5. Јована Матковић, ТРОСЛОЈНА АРХИТЕКТУРА КАО СОФТВЕРСКО РЕШЕЊЕ СИСТЕМА ЗА ЕВИДЕНЦИЈУ ТРЕНИНГА	1157-1160
6. Miloš Živić, ČET-BOTOVI U DOMENU PUTOVANJA I TURIZMA	1161-1164
7. Igor Malešević, SISTEM ZA PREGLED I PRETRAGU VESTI BAZIRAN NA GRAPHQL API-U I RAVENDB-U	1165-1168
8. Nikola Milosavljević, SOFTVERSKI ALAT ZA OPORAVAK DATOTEKA U SISTEMU DATOTEKA NTFS	1169-1172
9. Aleksandar Birmančević, ESTIMACIJA RADIO KANALA NA OSNOVU PILOTSKIH SEKVENCI	1173-1176
10. Petar Stošić, MIKROSERVISNA ARHITEKTURA NA PRIMERU WEB APLIKACIJE ZA PERSONALNE TRENERE	1177-1180
11. Лука Курељушић, АЛАТ ЗА ДЕКРИПЦИЈУ ZIP АРХИВА ПРИМЕНОМ NVIDIA CUDA ТЕХНОЛОГИЈЕ	1181-1184
12. Саша Бојанић, Иван Каштелан, ИМПЛЕМЕНТАЦИЈА ПОДРШКЕ ЗА ПРЕНОС ДАТОТЕКА ПОМОЋУ CANOPEN И HTTP ПРОТОКОЛА У ИНДУСТРИЈСКОМ ИНТЕРНЕТУ СТВАРИ	1185-1188
13. Светозар Вулин, СИСТЕМ ЗА ПРАЋЕЊЕ ПРИСУСТВА НА ДОГАЂАЈИМА	1189-1192
14. Milan Govedarica, KOMPARATIVNA ANALIZA SISTEMA ZA UPRAVLJANJE SADRŽAJEM	1193-1196
15. Милица Митровић, МЕРНО-УПРАВЉАЧКИ PID СИСТЕМ СА ЈЕДНОСМЕРНИМ МОТОРОМ	1197-1201
16. Ana Mičić, Vladimir Popović, РАМЕТНЕ БЕЖИЧНЕ КОМУНИКАЦИОНЕ ТЕХНОЛОГИЈЕ У СИСТЕМУ ЕЛЕКТРИЧНОГ ВОЗИЛА	1202-1205
17. Dražen Stojšin, ODREĐIVANJE PARAMETARA ANTENE I PRILAGOĐENJE PREOSTALOM DELU KOLA	1206-1209
18. Јелена Стојаковић, ФОРЕНЗИКА РАЧУНАРСКИХ МРЕЖА	1210-1213
19. Jelena Marinković, OPTIMIZACIJA UPOTREBE RESURSA U REALIZACIJI EMBEDDED SISTEMA ZA DALJINSKO UPRAVLJANJE I MONITORING	1214-1217
20. Aleksa Đoković, IMPLEMENTACIJA I VERIFIKACIJA SISTEMA ZA PROCESIRANJE PAKETA PRIMENOM FORMALNIH METODA	1218-1221
21. Андреј Скала, ДИЗАЈН И ЕВАЛУАЦИЈА АЛГОРИТАМА ЗА МЕРЕЊЕ УДАЉЕНОСТИ ЗАСНОВАНЕ НА ВРЕМЕНУ ДОЛАСКА СИГНАЛА (TOA) ЗА IEEE 802.11AZ	1222-1225
22. Marko Čančar, UTICAJ PARAMETARA DISTRIBUTIVNE MREŽE I DISTRIBUIRANOG GENERATORA NA VREDNOST DIFERENCIJALNE STRUJE PARALELNIH VODOVA	1226-1229
23. Марија Петровић, РАЗВОЈ БЕКЕНД ПОДСИСТЕМА БЕЗ СЕРВЕРА ЗА ВЕБ 3.0 ПЛАТФОРМУ ЗА УПРАВЉАЊЕ КУРСЕВИМА	1230-1233
24. Stefan Ljubović, RAZVOJ BLOKČEJN PODSISTEMA ZA VEB 3.0 PLATFORMU ZA UPRAVLJANJE KURSEVIMA	1234-1237

	STRANA
25. Катарина Жерајић, Јелена Сливка, ПРЕДИКЦИЈА КАШЊЕЊА АВИОНСКИХ ЛЕТОВА КОРИШЋЕЊЕМ АЛГОРИТАМА МАШИНСКОГ УЧЕЊА	1238-1241
26. Милица Прањкић, Јелена Сливка, МОДЕЛ ВРЕМЕНСКЕ СЕРИЈЕ ЗА ПРЕДВИЂАЊЕ ЦЕНА AWS СПОТ ИНСТАНЦИ	1242-1245
27. Наташа Ивановић, СУМАРИЗАЦИЈА НАУЧНИХ РАДОВА НА СРПСКОМ ЈЕЗИКУ ПРИМЕНОМ NLP МЕТОДА .	1246-1249
28. Mladen Planojević, VIZUALIZACIJA AUTOSAR LOGGING/TRACING PROTOKOLA	1250-1253
29. Predrag Nikolić, МАТЕМАТИЧКО MODELOVANJE I PID REGULACIJA TERMALNE ZONE U HIL OKRUŽENJU ...	1254-1257
30. Srđan Šuković, SOFTVERSKI SISTEM ZA SEMANTIČKU PRETRAGU TEKSTA PRIMENOM VEKTORSKIH BAZA PODATAKA	1258-1261
31. Sonja Tomčić, ЕКСТРАКТИВНА SUMARIZACIJA RECENZIJA HOTELA UPOTREBOM WORD2VEC, DOC2VEC I GPT-3.5 MODELA	1262-1265
32. Мирослав Богдановић, Владимир Рајс, БЕЖИЧНИ ПРЕНОС ЕЛЕКТРИЧНЕ ЕНЕРГИЈЕ ЗА ПОТРЕБЕ ПУЊЕЊА ЛАКИХ ЕЛЕКТРИЧНИХ ВОЗИЛА	1266-1269
33. Plema Lažetić, USLUGA UPRAVLJANJA KLJUČEVIMA ZA AMAZON VEB SERVICE	1270-1273
34. Миљана Симић, КОМПАРАТИВНА АНАЛИЗА ПЕРФОРМАНСИ РЕЛАЦИОНИХ И НЕРЕЛАЦИОНИХ СИСТЕМА БАЗА ПОДАТАКА	1274-1277
35. Aleksandar Pavić, Boris Dumnić, ТЕХНИЧКИ АСПЕКТИ ПРИКЛУЧЕЊА FOTONAPONSKIH ELEKTRANA NA DISTRIBUTIVNI SISTEM REPUBLIKE SRBIJE	1278-1281
36. Ксенија Радоњић, ИМПЛЕМЕНТАЦИЈА И ТЕСТИРАЊЕ ХИПЕРВИЗОРА ЗА ПАРАЛЕЛНО ИЗВРШАВАЊЕ ПАРТИЦИЈА НА ZEDBOARD ПЛАТФОРМИ	1282-1285
37. Aleksa Šaršanski, ЕКСПЕРИМЕНТИ СА VEKTORSKIM REPRESENTACIJAMA TEKSTA ZA EKSTRAKTIVNU SUMARIZACIJU	1286-1289
38. Sara Panić, DIZAJN I IMPLEMENTACIJA JEZIKA SPECIFIČNOG ZA DOMEN LOGIČKIH ENIGMATSKIH IGARA	1290-1293
39. Душан Радишић, РАЗВОЈ МОДУЛА ЗА ИНТЕГРАЦИЈУ ПРОЦЕСОРА У КОНФИГУРАБИЛНОЈ HALOWONE™ WI-FI IP ПЛАТФОРМИ	1294-1297
40. Vladimir Vrbica, Milan Vidaković, IMPLEMENTACIJA MIKROSERVISNE ARHITEKTURE KROZ VEB APLIKACIJU ZA PRODAJU AUTOMOBILA	1298-1301
41. Тијана Ђуркић, МУЛТИМОДАЛНО ПРЕПОЗНАВАЊЕ ЕМОЦИЈА ПОМОЋУ КОМПРИМОВАНИХ ГРАФОВСКИХ НЕУРОНСКИХ МРЕЖА.....	1302-1305
42. Alisa Milanko, RAZVOJ VEB APLIKACIJE U REALNOM VREMENU NA PRIMERU CHAT-a	1306-1309
43. Ksenija Prčić, Nikola Luburić, KOMONENTE ZA PODRŠKU SISTEMA ZA OPTIMIZACIJU RASPOREDA NASTAVE FAKULTETA TEHNIČKIH NAUKA	1310-1313

44. Marko Stanić, HORIZONTALNO SKALIRANJE K8S POD-OVA POMOĆU KEDA AUTOSCALER-A.....	1314-1316
--	-----------

Radovi iz oblasti: Saobraćaj

1. Милан Милановић, Вук Богдановић, АНАЛИЗА ЕФЕКТА СРЕДСТАВА ЗА УСПОРАВАЊЕ САОБРАЋАЈА	1317-1321
2. Anđela Vasić, ZNAČAJ POŠTANSKIH USLUGA U LANCIMA VREDNOSTI ELEKTRONSKE TRGOVINE U SRBIJI	1322-1325

Radovi iz oblasti: Mehatronika

1. Nebojša Gradinski, SISTEM ZA PRAĆENJE TEŽINE KOŠNICE SA DISTRIBUIRANE LOKACIJE	1326-1329
2. Adrijana Delić, BIOMETRIJSKE TEHNOLOGIJE U NAPREDNIM SISTEMIMA ZA POMOĆ VOZAČIMA	1330-1333
3. Srđan Bijelić, Vladimir Rajs, SISTEM PAMETNE KUĆE	1334-1337

UPOREDNA ANALIZA PRIMENE TOPLOTNIH PUMPI ZA POTREBE GREJANJA I PRIPREME SANITARNE TOPLE VODE PORODIČNOG OBJEKTA

COMPARATIVE ANALYSIS OF HEAT PUMP APPLICATION FOR HEATING AND SANITARY HOT WATER PREPARATION - A FAMILY HOUSE EXAMPLE

Marko Zagorac, Fakultet tehničkih nauka, Novi Sad

Oblast: MAŠINSTVO

Kratak sadržaj - U radu je prikazana analiza sistema toplotnih pumpi za potrebe grejanja i pripreme sanitarne tople vode. Opisani su neki glavni fizički procesi koji se odvijaju u toplotnim pumpama, data je i objašnjena njihova podela po više odlika, ali i načini i postupci po kojima se određeni tipovi tih toplotnih pumpi dimenzionišu. Ceo rad je zaokružen tehnokonomskom analizom i poređenjem više vrsta sistema toplotnih pumpi na izabranom porodičnom objektu preko specijalizovanog programskog paketa. Na kraju su upoređeni i prokomentarisani dobijeni rezultati.

Glavne reči: Grejanje, priprema sanitarne tople vode, toplotne pumpe, izbor sistema, uporedna analiza.

Abstract - The paper presents an analysis of the heat pump system for heating and sanitary hot water supply. Some main physical processes that take place in heat pumps are described, their division by several characteristics is given and explained, as well as the ways and procedures by which certain types of these heat pumps are dimensioned. The entire work is rounded off with a techno-economic analysis and a comparison of several types of heat pump systems on a selected family property using a specialized software package. At the end, the obtained results were compared and commented.

Key words: Heating, Domestic Hot Water, Heat pumps, System selection, Comparative analysis

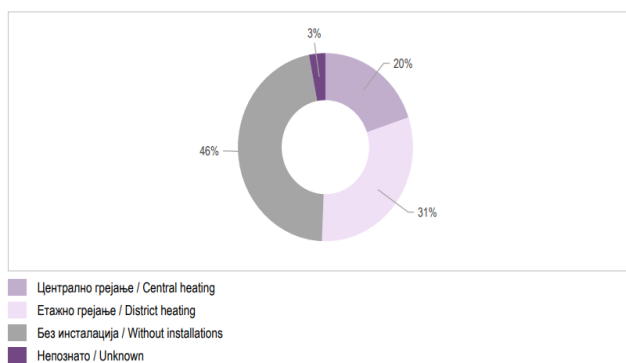
1. UVOD

U prvom poglavlju rada dat je trenutni presek stanja u Republici Srbiji po pitanju grejanja rezidencijalnih objekata, tj. postojanja instalacija grejanja u samim objektima i vrsti energenata koji se najčešće koriste.

U daljem delu ovog poglavlja se govori o potrebi za povećanjem energetske efikasnosti, smanjenju zagađenja koje proizilazi upotrebom konvencionalnih sistema za grejanje koji koriste fosilna goriva, ali i uštedi i finansijskom aspektu prelaska na nekonvencionalne sisteme koji koriste ekološki prihvatljivija rešenja.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Andelković, vanr. prof.

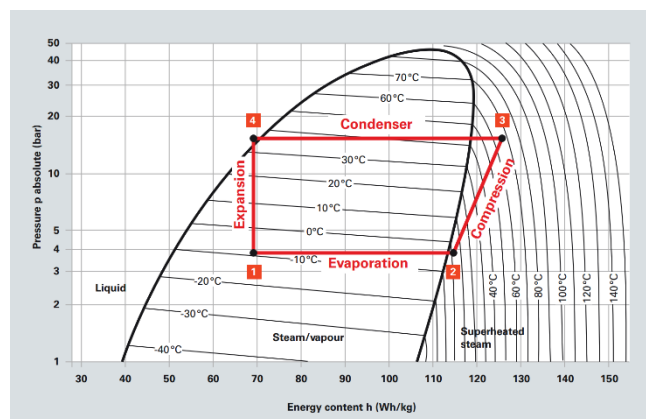


Slika 1: Opremljenost stanova instalacijama grejanja, Republika Srbija [1].

2. KRATAK ISTORIJSKI RAZVOJ TOPLOTNE PUMPE

Pronalasci Nicolas Leonard Sadi Carnot-a, koji je opštepriznat kao „otac termodinamike“, bili su od fundamentalnog značaja. Istraživao je procese (i njihove uzroke i posledice) u parnim mašinama, i otkrio da su svi procesi koji su uključivali konverziju toplote u snagu uglavnom reverzibilni. Ukoliko se uzme pravi medijum za prenos toplote, koristeći energiju, ne samo da toplota može da se doda u medijum već može i da se izvuče iz istog, nezavisno od ambijentalne temperature [2].

3. FIZIČKI PRINCIPI TOPLOTNE PUMPE



Slika 2: PH dijagram stanja rashladnog sredstva R 407C [2]

Koeficijent performansi je mera efikasnosti i definiše se kao odnos između troškova i koristi. U odnosu na

toplotnu pumpu ovo se definiše na sledeći način: prednost predstavlja količinu toplote prenetu na visokom temperaturnom nivou (kondenzacija), dok je trošak potrebna pogonska energija (kompresija) da bi se ovo postiglo.

Primenom logaritamskog p-h dijagrama, koeficijent performansi toplotne pumpe definisan je kao (jednačina 1.) [2]:

$$COP = \frac{h_3 - h_4}{h_3 - h_2} \quad (1)$$

gde je:

h_2 – entalpija na početku kompresije [J/kg].

h_3 – entalpija na kraju kompresije /početkuprenosa toplote [J/kg].

h_4 – entalpija na kraju kondenzacije /kraju prenosa toplote.

4. ZAKONSKA REGULATIVA

Što se tiče zakonske regulative u Srbiji na polju obnovljivih izvora energije, njihovo zakonsko uređenje, podsticaji države i mere koje se sprovode navedeni su u strategiji razvoja energetike Republike Srbije do 2025. sa projekcijama do 2030. godine, kao i aktivnostima koje se sprovode ili treba da budu sprovedene u tom pogledu, uz pomoć EU [3].

5. VRSTE I REŽIMI RADA TOPLOTNIH PUMPI

Toplotne pumpe se mogu podeliti prema izvoru dodatne energije na [4]:

- kompresione,
- apsorpcione.

Toplotna pumpa može da radi u:

- monovalentnom režimu rada, gde toplotna pumpa pokriva 100% godišnjeg toplotnog opterećenja grejanja objekta,
- bivalentnom režimu, gde toplotna pumpa pokriva deo godišnjeg toplotnog opterećenja grejanja objekta [4].

Kod bivalentnog načina rada razlikujemo:

1. bivalentno-alternativne,
2. bivalentno-paralelne,
3. delimično paralelne režime rada [4].

6. PRIMARNI IZVORI ENERGIJE ZA TOPLOTNU PUMPU

Tri primarna izvora energije za toplotne pumpe su [2]:

1. zemlja (kolektori i geotermalne sonde)
2. podzemna voda
3. vazduh

Broj cevni krugova ili petlji proizilazi iz maksimalne dužine pojedinačnih krugova od 100 m, specifične dužine cevi i potrebne ukupne površine geotermalnih kolektora (jednačina 2.) [2]:

$$N_{RK} = \frac{F_E \cdot L_{RL}}{100 m} \quad (2)$$

N_{RK} – broj cevni krugova / petlji,

F_E – ukupna površina geotermalnih kolektora [m^2],

L_{RL} – specifična dužina cevi po m^2 [m^{-1}].

Potrebna površina kolektora izračunava se po sledećem izrazu (formula 3.) [2]:

$$F_E = \frac{Q_K}{q_e} \quad (3)$$

F_E – potrebna površina geotermalnih kolektora [m^2],

Q_K – rashladni kapacitet toplotne pumpe [W],

q_e – maksimalna stopa ekstrakcije za specifično područje [W/m^2].

Potrebna dužina geotermalne sonde dobija se na osnovu jednačine (4) [2]:

$$l_{geo} = \frac{P_{TP,hl}}{\dot{q}_L} \quad (4)$$

gde su:

$P_{TP,hl}$ – rashladni kapacitet toplotne pumpe [W]

$\dot{q}_L$ – specifična toplota koja se preuzima iz zemljišta [W/m].

Voda je takođe veoma pogodan izvor toplote za toplotne pumpe. Čak i u hladnim zimskim danima, podzemne vode održavaju konstantnu temperaturu između 7 °C i 12 °C. Da bi se podzemna voda koristila za toplotnu pumpu prvo se mora izvući kroz dovodni bunar i transportovati do isparivača toplotne pumpe tipa voda/voda. Ohlađena voda se zatim usmerava kroz povratni bunar [2].

Ako se toplotna pumpa koristi za zagrevanje STV (sanitarne tople vode) tokom leta, pretpostavlja se da temperatura vazduha od 35 °C određuje maksimalnu snagu. Sa ovim dizajnom može se pretpostaviti širenje od 10 K (izraz 5.) [2]:

$$\dot{Q} = \dot{m} \cdot c \cdot \Delta t \quad (5)$$

$\dot{Q}$ – snaga toplotne pumpe [W],

$\dot{m}$ – protok [l/h],

c – toplotni kapacitet vode u [Wh/kgK],

Δt – temperaturna razlika [K].

7. SEKUNDARNI GREJNI KRUG. GREJANJE, SANITARNA TOPLA VODA I HLAĐENJE.

Sistemi za pripremu sanitarne tople vode (STV) [2]:

Zagrevanje STV pomoću toplotnih pumpi može biti pokriveno – u zavisnosti od zahteva – sledećim sistemima:

- Centralizovano zagrevanje STV pomoću a toplotne pumpe za STV.
- Centralizovano zagrevanje STV skladištenjem u rezervoar STV.

- Centralizovano zagrevanje STV pomoću čuvanje vode iz grejanog kruga u bufferu cilindru i centralizovanom modul za pijaću vodu.
- Decentralizovano grejanje preko individualnih stambenih podstanica.
- Decentralizovano grejanje na struju protočnim bojlerom [2].

Potrebna količina STV-a može se izračunati iz ukupne potrošnje energije tokom referentnog perioda (izraz 6.) [2]:

$$V_{DP} = \frac{Q_{DPB}}{c_w \cdot (t_{set} - t_{cw})} \quad (6)$$

V_{DP} – potrebna količina STV – a tokom referentnog perioda [l],

Q_{DPB} – potražnja za energijom tokom referentnog perioda [kWh],

c_w – specifični toplotni kapacitet

($= 1,163 \frac{Wh}{kgK}$ za vodu),

t_{set} – podešena temperatura rezervoara [K],

t_{cw} – temperatura hladne vode [K].

8. PROJEKTOVANJE SISTEMA GREJANJA TOPLOTNIM PUMPAMA POMOĆU PROGRAMSKOG PAKETA „GEOTSOL“

Prilikom projektovanja sistema grejanja pomoću GeoTSOL-a prvi korak podrazumeva odabir lokacije. Pomoću potprograma MeteoSyn bira se lokacija i povlače se klimatski podaci za tu lokaciju. Izvor klimatskih podataka je servis „Meteonorm“. Treba imati u vidu da su dostupni podaci samo do 2015.god [5].

Nakon definisanja i dimenzionisanja svih delova sistema, simulacijom se dobijaju podaci o efikasnosti sistema, potrošnji električne energije, dobijenoj količini toplotne energije, gubicima i tako dalje [5].

U delu finansijske analize vrši se analiza ekonomske opravdanosti sistema. Finansijskom analizom se dobija modifikovana stopa povrata, odnosno MIRR, koji predstavlja jasan indikator isplativosti, odnosno što je MIRR veći, sistem je isplativiji. Osim MIRR, dobija se i cena grejanja po kWh toplotne energije, koja je značajna u pogledu poređenja sistema sa nekim drugim sistemima [5].

9. TEHNOEKONOMSKA ANALIZA RAZLIČITIH SISTEMA SA TOPLOTNOM PUMPOM

Izvršena je analiza tri različita sistema sa toplotnim pumpama.

U tabelama u nastavku dati su rezultati proračuna za različite sisteme sa toplotnim pumpama. Dati su sezonski faktori performansi, količine dobijene toplotne energije, kao i potrošnje električne energije.

Dobijeni rezultati su potom međusobno upoređeni [5].

Tabela 1: Rezultati proračuna sistema sa geo-sondama [5].

Rezultati finansijske analize		Vrednost
Dobijena toplotna energija		20185 kWh/godišnje
Neto sadašnja vrednost		158.20 €
Investicioni anuitet	Sistem sa TP	1512.67 €
	Drugi sistem	150 €
Godišnji operativni troškovi	Sistem sa TP	564.39 €
	Drugi sistem	1422.30 €
Godišnji troškovi održavanja	Sistem sa TP	303.72 €
	Drugi sistem	303.72 €
Cena grejanja	Sistem sa TP	0.093 €/kWh
	Drugi sistem	0.0935 €/kWh
Povrat (MIRR)		0,05 %
Višak		158.20 €

Tabela 2: Rezultati proračuna sistema podzemnom vodom [5].

Rezultati finansijske analize		Vrednost
Dobijena toplotna energija		20171 kWh/godišnje
Neto sadašnja vrednost		813.45 €
Investicioni anuitet	Sistem sa TP	750.00 €
	Drugi sistem	150.00 €
Godišnji operativni troškovi	Sistem sa TP	542.28 €
	Drugi sistem	1182.90 €
Godišnji troškovi održavanja	Sistem sa TP	303.72 €
	Drugi sistem	303.72 €
Cena grejanja	Sistem sa TP	0.079 €/kWh
	Drugi sistem	0.081 €/kWh
Povrat (MIRR)		0,33 %
Višak		813.45 €

Tabela 3: Rezultati proračuna hibridnog sistema sa vazдушnom TP i kotlom [5].

Rezultati finansijske analize		Vrednost
Dobijena toplotna energija		20087 kWh/godišnje
Neto sadašnja vrednost		2773.54 €
Investicioni anuitet	Sistem sa TP	300.61 €
	Drugi sistem	149.83 €
Godišnji operativni troškovi	Sistem sa TP	690.40 €
	Drugi sistem	1178.53 €
Godišnji troškovi održavanja	Sistem sa TP	303.71 €
	Drugi sistem	303.71 €
Cena grejanja	Sistem sa TP	0.064 €/kWh
	Drugi sistem	0.081 €/kWh
Povrat (MIRR)		6,05 %
Višak		2773.54 €

Tabela 4: Poređenje podataka različitih sistema sa TP [5]

Sistem sa TP	Cena sistema	Cena grejanja	SPF	MIRR
TP sa geotermalnim sondama	20 000 €	0,093 €/kWh	3,9	0,05
TP sa podzemnim vodama	15 000 €	0,079 €/kWh	4,15	0,33
Hibridni sistem sa kotlom	6000 €	0,064 €/kWh	3,97	6,05

10. ZAKLJUČAK

U Republici Srbiji je možda i najvažniji zadatak, animirati ljude, obično građanstvo, javno mnjenje, o potrebi prelaska na ovakve vidove dobijanja energije, i upoznati ih o svim posledicama koje nečinjenje sada može imati za njihova pokoljenja u budućnosti. Naravno, i predstaviti im sve benefite kojih je zaista mnogo.

Kod sistema sa toplotnim pumpama smo u ovome radu videli da imamo tu mogućnost da biramo odgovarajući, prema našim potrebama i finansijskim i ostalim mogućnostima, i da će se skoro sigurno naći neki koji je baš „idealni“ za naš dom. Bilo da je to toplotna pumpa sa geosondama ili kolektorima, sa podzemnim vodama ili vazdušna. Sva ova analiza koja je urađena je ustvari pokazala jednu možda i ključnu stvar. Koji god da od analiziranih sistema odaberemo i implementiramo, sigurno nećemo pogrešiti, a to je kada se sve sabere i oduzme, najvažnije.

11. LITERATURA

1. <https://popis2022.stat.gov.rs/sr-latn/publikacijenajava/> - **dostupno na:** <https://publikacije.stat.gov.rs/G2023/pdf/G20234011.pdf> (datum poslednjeg pristupanja 21.11.2023)
2. https://cdn0.scrvt.com/2828ebc457efab95be01dd36047e3b52/ceab8eb0c449dcd2/103428a45b62/Technical-Guide-Heat-Pumps_04-2012_GB.pdf?fbclid=IwAR1M1Ktiak9MJxjduvZ2FZ4imxo2o70BTOCXeK7DC_pTONPSipukXlj5Xyk (datum polednjeg pristupanja 14.11.2023)
3. <https://www.pravno-informacioni-sistem.rs/SlGlasnikPortal/eli/rep/sgrs/skupstina/ostalo/2015/101/1/r> (datum poslednjeg pristupanja 02.11.2023)

4. mr. Aleksandra Čenejac, prof. dr. Radivoje Bjelaković, Slobodan Petrović, Miloš Čenejac, *STUDIJA O MOGUĆNOSTIMA PRIMENE TOPLOTNIH PUMPI NA TERITORIJI AP VOJVODINE, KAKO ZA INDIVIDUALNE I KOMERCIJALNE OBJEKTE POJEDINAČNO, TAKO I ZA DALJINSKE SISTEME GREJANJA PO UGLEDU NA EU, A U SKLADU SA AKTUELNOM DIREKTIVOM*, Fakultet tehničkih nauka Novi Sad, maj 2012 godine.

5. Dragan Lazić, *UPOREDNA ANALIZA SISTEMA GRIJANJA TOPLLOTNIM PUMPAMA I HIBRIDNIM SISTEMIMA*, Master rad, Novi Sad, 2018 godine

Kratka biografija:



Marko Zagorac rođen je u Novom Sadu, SR Jugoslavija 1998. god. Master rad odbranio je 2023. godine na Fakultetu tehničkih nauka iz oblasti Mašinstva – Termotehnika.
Kontakt:
zagoracmarko1@gmail.com



Aleksandar Andelković rođen je u Šapcu 1981. god. Doktorirao je 2015. god. Od 2020. je u zvanju vanrednog profesora. Oblast interesovanja su Nekonvencionalni sistemi grejanja i hlađenja.

**PROJEKAT VENTILACIONOG SISTEMA BIM METODOLOGIJOM ZA
KOMERCIJALNI OBJEKT POVRŠINE 580 m²****VENTILATION SYSTEM PROJECT USING BIM METHODOLOGY FOR A
COMMERCIAL BUILDING OF 580 m² AREA**Srđan Terzin, *Fakultet tehničkih nauka, Novi Sad***Oblast: MAŠINSTVO**

Kratak sadržaj - U radu je prikazana primena BIM metodologije u sistemima grejanja, klimatizacije i ventilacije. Opisani su glavni fizički procesi koji se odvijaju u ventilacionim sistemima kao što su termodinamika, prenos toplote i mehanika fluida. Takođe, objašnjena je podela KGH sistema po više odlika, vrsta, ali i načina i postupaka po kojima se određeni tipovi tih sistema dimenzionišu i usvajaju za projekat. Ceo rad je obuhvaćen tehnoeкономskom analizom sistema grejanja, ventilacije i klimatizacije na izabranom komercijalnom objektu preko specijalizovanog programskog paketa "Autodesk Revit", sa fokusom na računski zadatak sistema ventilacije čiji su dobijeni rezultati na kraju izvedeni i prokomentarisani.

Ključne reči: Grejanje, ventilacija, klimatizacija, BIM metodologija, izbor sistema.

Abstract - The paper presents an analysis of the ventilation system implemented in a commercial building. Some main physical processes that take place in heating, ventilation and air conditioning are described, their division by several characteristics is given and explained, as well as the ways and procedures by which certain types of these systems are dimensioned and selected. The entire work is rounded off with a techno-economic analysis and a comparison of several types of heating, ventilation and air conditioning systems on a selected commercial building using a specialized software package "Autodesk Revit", with the focus on ventilation system calculation task and its result.

Keywords: Heating, ventilation, air conditioning, BIM methodology, system selection.

1. UVOD

U prvom poglavlju rada prikazana je istorija BIM metodologije (Istorija BIM metodologije, 2023) kao i njena primena u sistemima grejanja, ventilacije i klimatizacije [6].

Prvi softverski alati za razvoj projekata u zgradarstvu pojavili su se u periodu 70-ih godina prošloga veka, a za

koncept pojave "upravljanjem informacijama o zgradama" nije zaslužna jedna osoba već kontinuitet

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Anđelković, vanr. prof.

industrijske revolucije, počevši od CAM (eng. Computer-Aided Manufacturing) sistema kompjuterskog konstruisanja do ArchiCAD-a [3], koji je širu primenu stekao 1987. godine i praktično bio prvi moderniji BIM softver upotrebljen na poznatim operativnim sistemima Windows i Apple.



Slika 1. Prikaz objekta u trodimenzionalnom prostoru ArchiCAD [3]

2. TEORIJSKI KONCEPTI RADA KGH SISTEMA

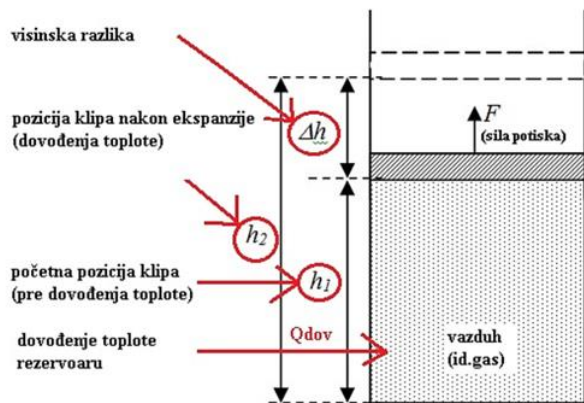
Prvi zakon termodinamike [1], odnosi se na promenu unutrašnje energije sistema prilikom razmene toplote i vršenja rada. Ovaj princip su prvi zaključili i otkrili fizičari Robert Majer (nem. Robert Mayer) i Džejms Džul (eng. James Prescott Joule).

Ukoliko posmatramo idealni gas (npr. vazduh) zatvoren u cilindru sa pokretnim klipom, i ukoliko se njemu dovede spoljašnja toplota, unutrašnja energija gasa se povećava (ekspanzija) i vrši se određeni rad (pomeranje klipa):

Eksperimentalno je dokazano (I princip termodinamike, bez dat.) da je dovedena količina toplote (Q_{dov}) jednaka zbiru promene unutrašnje energije sistema (ΔU) i izvršenog rada (A):

$$Q = \Delta U + A$$

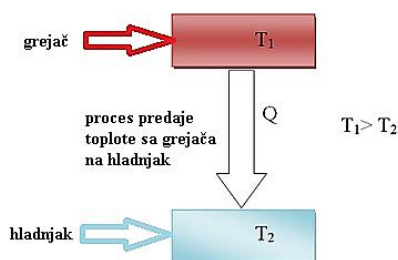
Može se zaključiti da dovedena količina toplote sistemu se troši na povećanje njegove unutrašnje energije i na vršenje rada sistema.



Slika 2. Primer [1]

Drugi zakon termodinamike [2], govori nam da se ukupna entropija (mera za energiju koja se nasuprot slobodnoj više ne može pretvoriti u rad) izolovanog sistema nikada ne može smanjiti tokom vremena.. Ukupna entropija sistema i njegovog okruženja se povećava i proces je **nepovratan** u termodinamičkom smislu.

Na sledećem primeru prikazana su dva rezervoara koja sadrže grejač (od koga se oduzima određena količina toplote) i hladnjak (kome se predaje količina toplote) i prilikom ovog procesa se vrši **rad**:



Slika 3. Primer [2]

Projektovanje sistema grejanja i hlađenja bilo bi nemoguće bez osnovnog poznavanja **fenomena prenosa toplote** [5]. Razmena toplote između pojedinih tela odvija se na dva, po fizičkom mehanizmu sasvim različita načina: prenos toplote posredstvom materijalnih čestica i prenos zračenjem koje nije vezano za materijalnu sredinu kao posrednika. Prostiranje toplote kroz materijalnu sredinu vrši se takođe na dva donekle različita načina, tako da se najčešće govori o tri mehanizma prenosa toplote, prikazanih na slici 4.

Provođenje toplote [5], odnosno kondukcija se odvija u mikro razmerama sa jednog molekula na drugi. Na primer, kod gasova se prenos toplote vrši sudaranjem molekula iz više zagrejanog sa molekulima u manje zagrejanom sloju gasa, pa prema tome, nezavisno kroz koje se materijalno telo toplota prenosi, mehanizam provođenja je vezan za kretanje elementarnih čestica, **slika 5.**



Slika 4. Ilustrativni prikaz načina prenosa toplote [5]



Slika 5. Ilustrativni prikaz prenosa toplote kondukcijom [5]

Prenos toplote **konvekcijom** (Prenos toplote, bez dat.) definiše se **koeficijentom prenosa toplote α (alfa)** koji povezuje temperatursku razliku površine razmenjivača i fluida i količinu toplote u jedinici vremena koja se prenosi. Pri konvekciji toplota se prostire strujanjem samog fluida (tečnosti ili gasa) i to:

- kroz fluid
- sa fluida na čvrstu površinu ili
- sa čvrste površine na fluid

Veliki broj eksperimenata je izvršen sa strujanjem u okruglim cevima (Marić, 2002), gde je dosta strogo određen momenat kada laminarno kretanje prelazi u turbulentno.

To je momenat kada tzv. **Rejnoldsov broj (Re)** prelazi kritičnu vrednost:

$$Re = \frac{w \times d}{\nu} [-]$$

gde je:

w - brzina strujanja fluida [m/s]

d - prečnik [m]

ν - kinematska viskoznost fluida [m²/s]

Re_k - kritična vrednost Rejnoldsovog broja (Re_k=2300)

U prenosu toplote na granici (površini) unutar fluida, **Nusseltov broj** je bezdimenzijska vrednost koja predstavlja odnos konvektivnog i konduktivnog prenosa toplote preko te granice.

$$Nu = \frac{\alpha \times L}{\lambda} [-]$$

gde je:

α - konvektivni koeficijent prenosa toplote [W/m²K]

L - karakteristična dužina [m]

λ - toplotna provodljivost fluida [W/mK]

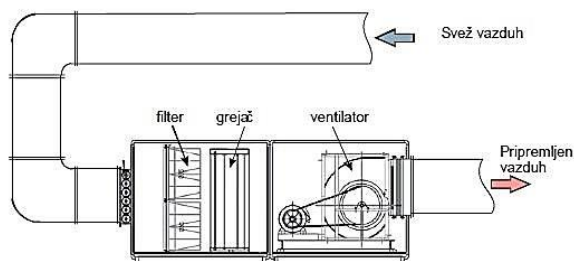
Za transport toplotne energije (energija koju telu donose elektromagnetni talasi i koja se pri nailasku na telo pretvara u njegovu unutrašnju energiju, odnosno menja toplotno stanje tela) od skoro isključivog uticaja je tzv.

temperatursko zračenje. Ukupna energija koju nose zraci od sva tri izvora zove se **sjaj** tog tela [4]:

$$r + a + d = 1$$

Kao nosilac toplote (radni fluid) u vazдушnim sistemima javlja se **vazduh**. Vazduh se zagreva ili hladi u grejaču, vlaži ili suši, filtrira i, pripremljen na odgovarajući način, ubacuje se direktno u prostoriju [7].

U prostoriji se ubačen vazduh meša sa sobnim i na taj način se postiže željena temperatura i vlažnost vazduha u prostoriji.



Slika 6. Ventilaciona komora za rad sa svežim vazduhom [7]

3. RAČUNSKI ZADATAK

Za komercijalni objekat površine **580 m²** proračunat je ventilacioni sistem namenjen za snabdevanje prostorija svežim vazduhom u bilo kom vremenskom periodu.

Za prizemlje, predviđeni protok vazduha po času iznosi **4.970 m³/h**, dok za sprat predviđeni protok je **4.400 m³/h**.

Na prizemlju i na prvom spratu nalaze se prostorije i za njih predviđeni protok vazduha, nabrojane u tabeli 1. i u tabeli 2:

Tabela 1. Prikaz broja prostorija i potrebnog protoka vazduha u prizemlju

R.br.	Naziv prostorije	Broj prostorija	Površina	Potreban protok po prostoriji
1.	Magacin	1	10 m ²	115 m ³ /h
2.	Toalet	2	10 m ²	170 m ³ /h
3.	Kancelarija	1	10 m ²	170 m ³ /h
4.	Kancelarija	1	18 m ²	300 m ³ /h
5.	Kancelarija	2	21 m ²	360 m ³ /h
6.	Menza	1	38 m ²	850 m ³ /h
7.	Sala za sastanke	1	48 m ²	1.075 m ³ /h
8.	Hodnik	1	122 m ²	1.400 m ³ /h
Σ =				4.970 m³/h

U tabeli 3. prikazani su ukupni troškovi repromaterijala ventilacionog sistema i prateće opreme za komercijalni objekat.

Tabela 2. Prikaz broja prostorija i potrebnog protoka vazduha na prvom spratu

R.br.	Naziv prostorije	Broj prostorija	Površina	Potreban protok po prostoriji
1.	Toalet	2	10 m ²	180 m ³ /h
2.	Kancelarija	2	21 m ²	380 m ³ /h
3.	Menza	1	38 m ²	920 m ³ /h
4.	Sala za sastanke	1	48 m ²	1.160 m ³ /h
5.	Hodnik	1	100 m ²	1.200 m ³ /h
Σ =				4.400 m³/h

Tabela 3. Ukupni troškovi repromaterijala

R.br.	Naziv	Cena
1.	Ventilacioni kanali	190.035,37 RSD
2.	Vazdušni terminali	564.318,00 RSD
3.	Regulacioni elementi	1.379.209,60 RSD
4.	Redukcije	393.265,60 RSD
5.	Fitinzi	691.270,21 RSD
6.	Mehanička oprema	1.130.511,20 RSD
Σ=		4.348.609,98 RSD
Σ=		37.104,18 €

4. ZAKLJUČAK

Analizirana je primena BIM metodologije u projektovanju i njena važnost u integraciji različitih disciplina, što rezultira efikasnijim procesom rada, boljom koordinacijom između inženjera i smanjenjem grešaka u fazi izgradnje.

Ovde su identifikovane specifične prednosti u optimizaciji dizajna sistema za ventilaciju, uključujući bolju vizualizaciju, analizu performansi sistema i pristupačnije upravljanje tokom faze izvođenja radova, konstruisanja i održavanja.

Ključni teorijski koncepti poput termodinamike, prenosa toplote i mehanike fluida su takođe predstavljeni i detaljno izvedeni kako bi se istakla važnost razumevanja ovih principa za inženjere koji rade na projektima KGH sistema.

Sprovedena je analiza troškova i repromaterijala u programskom paketu Autodesk Revit u koji prikazuje korisniku detaljan opis svakog sistema i elementa iskorišćenog u projektu, prikazujući time kako se može efikasnije upravljati troškovima i optimizaciji upotrebe materijala. Kao deo istraživanja, izvršen je računski zadatak koji je obuhvatio proračun ventilacionih kanala i prateće opreme, demonstrirajući praktičnu primenu teorijskih principa i primene BIM metodologije u projektovanju.

5. LITERATURA

- [1]. I princip termodinamike, bez dat. (n.d.). *Први принцип термодинамике*. Preuzeto Mart 10, 2023 sa <https://fizis.rs/prvi-princip-termodinamike/>
- [2]. II princip termodinamike, bez dat. (n.d.). *Други принцип термодинамике*. Preuzeto Mart 10, 2023 sa <https://fizis.rs/ii-princip-termodinamike/>
- [3]. Istorija BIM metodologije. (2023). *A history of BIM*. Preuzeto Mart 6, 2023 sa <https://www.letsbuild.com/blog/a-history-of-bim>
- [4]. Marić, M. (2002). *Nauka o toploti (termodinamika, prenos toplote, sagorevanje)*. Novi Sad: Fakultet tehničkih nauka.
- [5]. Prenos toplote, bez dat. (n.d.). *Prenos toplote*. Preuzeto Mart 13, 2023 sa <https://odseknis.akademijanis.edu.rs/wp-content/plugins/vts-predmeti/uploads/Kondukcija.pdf>
- [6]. Primena BIM. (2023). *What are the benefits of BIM?* Preuzeto Mart 6, 2023 sa <https://www.autodesk.com/solutions/aec/bim/benefits-of-bim>
- [7]. Todorović Maja, bez dat. (n.d.). *Energetska efikasnost sistema grejanja i klimatizacije (Mašinski fakultet, Beograd)*. Preuzeto April 27, 2023 sa Departman za energetiku i procesnu tehniku FTN: <https://dept.uns.ac.rs/download/energetska-efikasnost-termoenergetskih-sistema-u-zgradarstvu-a7/?ind=1671178458580&filename=07-Predavanje-Sistemi-klimatizacije.pdf&wpdmdl=11135&refresh=65d3bc54640cf1708375124>

Kratka biografija:



Srđan Terzin rođen je u Somboru, 1994. godine. Master rad iz oblasti mašinstva odbranio je 2024. godine na Fakultetu tehničkih nauka.

Kontakt:
sterzin94@gmail.com



Aleksandar Andelković rođen je u Šapcu 1981. god. Doktorirao je 2015. god. Od 2020. je u zvanju vanrednog profesora. Oblast interesovanja su Nekonvencionalni sistemi grejanja i hlađenja.

**ПОДЛОГЕ ЗА ТЕХНИЧКУ КОНТРОЛУ ЕЛЕКТРИЧНИХ ЛИФТОВА СА
СТАНОВИШТА БЕЗБЕДНОСТИ И ЗАШТИТЕ НА РАДУ****THE BASIS FOR THE TECHNICAL INSPECTION OF ELECTRIC LIFTS FROM THE
PERSPECTIVE OF SAFETY AND OCCUPATIONAL HEALTH**Цвијетин Пејчић, Радомир Ђокић, *Факултет техничких наука, Нови Сад***Област – МЕХАНИЗАЦИЈА И
КОНСТРУКЦИОНО МАШИНСТВО**

Кратак садржај – У уводном делу рада представљен је историјски развој лифтова који представљају значајан вид транспорта људи и терета. Посебна пажња посвећена је лифтовима на електрични погон. Описани су основни елементи и начин њиховог рада са освртом на безбедносне компоненте. Такође су у раду наведене најзначајније инцидентне ситуације са корективним мерама и циљу смањена ризика од незгоде. На крају се приступило прегледу и провери лифта који су од виталног значаја за безбедност корисника и оцавање функционалности система.

Кључне речи: лифт, преглед и провера, основни елементи, инцидентне ситуације

Abstract – The introductory part of the paper presents the historical development of elevators, which represent a significant means of transporting people and goods. Special attention is given to electrically powered elevators. The basic elements and their operation are described, with a focus on safety components. Additionally, the paper lists the most significant incident situations, along with corrective measures aimed at reducing the risk of accidents. Finally, the importance of elevator inspection and testing is emphasized for user safety and system functionality maintenance.

Keywords: lifts, inspection and testing, basic elements, incident situations.

1. УВОД

Овај рад бави се лифтовима на електрични погон. У данашњем урбаном окружењу, где се лифтови користе у зградама различитих намена, од стамбених до пословних и индустријских, важно је разумети техничке захтеве који се односе на редовно одржавање, контролу и безбедно коришћење лифтова.

У датом контексту, овај рад ће истражити стандарде, прописе и праксе које се примењују при техничкој контроли електричних лифтова са циљем очувања њихове функционалности и сигурности.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Радомир Ђокић, ванр. проф.

2. ЛИФТОВИ НА ЕЛЕКТРИЧНИ ПОГОН

Лифтови на електрични погон као извор енергије за подизање и спуштање кабине користе електромотор. Он покреће механичке компоненте како би се кабина и противтег кретали по вертикалним шинама. Контролни систем омогућава корисницима да одаберу жељени спрат или аутоматски управља кретањем кабине. Сигурносни механизми су уграђени ради заштите путника и спратова. Електрични лифтови су широко коришћени због своје ефикасности, поузданости и економичности у употреби.

Постоје различити типови електричних лифтова који се користе у различитим окружењима и наменама. Сваки тип има своје карактеристике и предности прилагођене различитим захтевима и потребама. Неки од основних типова електричних лифтова су: електрични лифтови са и без машинске просторије, теретни и малотеретни, ауто лифтови, лифтови за пословне зграде, панорамски, стамбени, индустријски и кућни лифтови [1].

**2.1. Основни елементи електричних лифтова и
начин њиховог рада**

Основни елементи електричних лифтова укључују делове који функционишу синхронизовано како би обезбедили безбедно и ефикасно кретање путника или терета између спратова зграда. У основне елементе убрајамо:

- Машинска просторија
- Возно окно
- Противтег
- Врата лифта
- Погонска челична ужад
- Кабина
- Вођице
- Хватачки уређаји
- Одбојници кабине
- Забраве на вратима

Срце лифта јесте електрични мотор, који када се активира покреће механизам који одговарајућим добошем или системом премотавања ужади преко ужетњача покреће кабину дуж вођица. Простор у који људи улазе – излазе представља кабина лифта. Врата кабине морају бити опремљена уређајем за блокаду и кинематским механизмом, јер се на тај начин обезбеђује сигурност путника. Битан елемент јесу и вођице чија се улога огледа у немогућности клаћења

кабине и противтега. Возно окно представља простор у коме се кабина и противтег крећу вертикално. Данас нешто мање заступљена, али ништа мање важна компонента лифта јесте машинска просторија, која представља простор где су смештени погонска машина и друга потребна опрема. Ослонци и одбојници ограничавају кретање кабине на доле и налазе се у јами возног окна. Лифт је опремљен и аутоматским системом за укључивање хватачког уређаја активирањем граничника брзине. Граничник брзине се активира уколико се кабина креће повећаном брзином на доле. Уграђује се на бочним странама рама кабине [2].

Основни елементи раде заједно како би омогућили сигурно и ефикасно кретање лифта између различитих спратова.



Слика 1. Основне компоненте електричних лифтова [3]

2.2. Предности и недостаци електричних лифтова

Електрични лифтови доносе низ предности у поређењу са другим лифтовима, али свакако да постоје и одређени неостаци. Неке кључне предности су:

- Брзина и ефикасност: обично пружају бржи и ефикаснији транспорт у поређењу са хидрауличним или другим врстама лифта, што може бити корисно у зградама са великим бројем спратова или високим фреквенцијама коришћења.
- Поузданост: електрични лифтови имају високу поузданост и мање су подложни кваровима и то их чини популарним избором за комерцијалне и стамбене објекте.

- Високи капацитет: електрични лифтови обично имају већи капацитет за носивост у поређењу са хидрауличним лифтовима.
- Ниска потрошња енергије: модерни електрични лифтови су обично дизајнирани са ефикасним моторима и системима управљања енергијом, што доприноси смањењу потрошње енергије.

Са друге стране недостаци електричних лифтова су:

- Потреба за електричном енергијом: ови лифтови захтевају стално снабдевање електричном енергијом како би радили, па квар у напајању може довести до застоја или губитка функционалности.
- Трошкови инсталације: инсталација може бити скупља у поређењу са хидрауличним или другим врстама лифтова, а посебно у случају комплексних или високих инсталација.
- Потребно одржавање: иако су електрични лифтови обично поуздани, они захтевају редовно одржавање како би се осигурало да раде исправно и безбедно, што може додатно повећати трошкове током времена
- Потенцијал за кварове електронике: електрични лифтови садрже комплексне електронске компоненте које су подложне кваровима или оштећењима услед пренапона или других електричних проблема.

Иако електрични лифтови имају многе предности, важно је узети у обзир и њихове недостатке.

3. ПРЕГЛЕД И ПРОВЕРА ЛИФТОВА ЗА ПРЕВОЗ ЛИЦА И ТЕРЕТА НА ЕЛЕКТРИЧНИ ПОГОН

Постоји низ корака и активности које је потребно пратити приликом провере и прегледа лифтова, а све са циљем да се провери њихова сигурност за употребу, да испуњавају све прописе и стандарде безбедности.

Преглед и контролу обављају лица која су запослена у именованом контролном телу од стране надлежног министарства, а која су оспособљена од стране Акредитационог тела.

Контрола лифтова обухвата проверу исправности рада сигурносних уређаја и делова, степена њихове истрошености и безбедности рада постројења, као и статичко и динамичко испитивање постројења, односно усаглашеност предметног постројења са референтним документима.

Да би се квалитетно обавило контролисање лифтова неопходно је да именовано тело за преглед и проверу лифтова поседује одређену мерну опрему.

Разликујемо две основне контроле лифтова:

- Контролисање лифтова пре стављања у употребу
- Контролисање лифтова у употреби

У оквиру прве поделе разликујемо завршну верификацију лифта и појединачну верификацију лифта, док у оквиру друге поделе разликујемо редован и ванредни преглед лифта.

Контролисање лифтова обухвата активности као што су преглед техничке документације, преглед предмета контролисања (лифт) који подразумева проверу сигурности, поузданости и исправности рада, испитивање карактеристика, обраду резултата и оцену усаглашености [4].

3.1. Редован преглед

Редован преглед лифта дефинисан је чланом 6 Правилника о прегледу лифтова у употреби („СЛ Гласник РС 15/17“). Обавља се најмање једном годишње [5].

Власник лифта је у обавези да обезбеди редован преглед, док сам преглед обавља именовано тело, а лице задужено за одржавање лифта присуствује прегледу и по потреби учествује у истом

У поступку редовног прегледа лифта проверава се:

- Исправан рад опреме за безбедност и заштиту
- Исправност друге опреме која би могла да утиче на безбедност
- Да ли су настале промене на лифту које могу да утичу на безбедност
- Да ли су настале промену у окружењу које могу да утичу на безбедност
- Да ли долази до промена код употребе лифта које могу да утичу на безбедност
- Да ли се на лифту налазе све ознаке и упутства за употребу, одржавање и спашавање лица из лифта
- Да ли су у књигу одржавања лифта уписане све промене настале од последњег редовног прегледа
- Да ли су од последњег редовног прегледа уклоњени сви недостаци који су утврђени у извештају о прегледу

После обављеног прегледа саставља се извештај о прегледу. Извештај саставља Именовано тело. Уколико постоје неки недостаци на лифту, извештај о прегледу треба да их садржи уз предложени рок за њихово отклањање.

Поред тога извештај садржи податке о пропису на основу којег је тај преглед извршен (потребно је навести назив прописа и број службеног гласника у коме је тај пропис објављен).

3.2. Формирање извештаја о прегледу и провери

Извештај о прегледу је документ који се саставља након обављеног прегледа и који садржи јасан закључак да ли предмет прегледа задовољава захтеве и техничке нормативе из одговарајућих референтних стандарда.

Извештај садржи преглед протокола конкретно за лифт који се испитује, резултате прегледа, евентуалне идентификоване недостатке, а у том случају и мере за отклањање недостатака.

Извештај се састоји из три основна поглавља:

- Општи подаци о предмету контролисања
- Технички подаци
- Протокол прегледа и провере [6].

Табела 1. Стручни извештај за лифтове на електрични погон

ОПШТИ ПОДАЦИ О ПРЕДМЕТУ ПРЕГЛЕДА			
Број извештаја:			
Предмет прегледа:	ЕЛЕКТРИЧНИ ПУТНИЧКИ ЛИФТ		
Врста прегледа:	РЕДОВНИ ПРЕГЛЕД		
Локација:			
ТЕХНИЧКИ ПОДАЦИ			
Произвођач:		Снага агрегата:	3,9 kW
Типска ознака:	R 40	Номинална брзина:	
Фабрички број:		Пречник ужетњаче:	Ø520 mm
ID број лифта:	I009-012.3443	Носећа ужад:	3 x Ø11 mm
Година производње:	/	Врата возног окна:	Полуаутоматска, обртна

Носивост:		Врата кабине:	Ручна двокрилна, обртна
Број станица/прилаза:		Управљање лифтом:	Јединачно
Носилац одржавања:			
Контрола прописане документације:	Сертификат о усаглашености пре пуштања у рад		
Датум прегледа лифта:			

Референтни документи:

- SRPS EN 81-20:2020 – Безбедносна правила за конструкцију и уградњу лифтова – лифтови за превоз лица и терета – Део 20: Лифтови за превоз лица и терета са пратиоцем
- Правилник о прегледима лифтова у употреби („Службени Гласник РС“ бр. 15/17)
- Правилник о безбедности лифтова („Службени Гласник РС“ бр. 15/17 и 21/20)
- *Правилник о техничким нормативима за електричне инсталације ниског напона („Службени лист СРГ“ број 28/95)
- *Правилник о техничким нормативима за заштиту објеката од атмосферског пражњења („Службени лист СРГ“ број 11/96)

**Обележени правилници нису из обима акредитације*

Опрема која се користи за преглед лифта:

1. **Дигитално ласерско уређај** са сталком, до 70,0 m, произвођач: „BOSCH“ / Germany, тип: „DLE 70“, Фабрички број: 009728987, година производње: 2011.
2. **Дигитални уређај силе**, до 100,0 kg, произвођач: „PROBUS GROUP“ / Lutron, Taiwan, тип: „PRO FG – 5100“, Фабрички број: 1 82269, година производње: 2010.
3. **Дигитални уређај осветљења**, од 40 – 400.000 lux, произвођач: „PROBUS GROUP“ / Lutron, Taiwan, тип: „PRO LX – 1108“, Фабрички број: Q442931, година производње: 2010.
4. **Дигитални уређај звука (буке)**, од 30 – 130 db, произвођач: „PROBUS GROUP“ / Lutron, Taiwan, тип: „PRO DT – 805“, Фабрички број: 08080261, година производње: 2010.
5. **Динамометар за мерење затезне силе у сајлама и ужадима (тензиометар)**, од 0 – 5 kN, произвођач: „TRC PRO“ / Србија, тип: „TRC-MMMU-5kN-10MM“, Фабрички број: 160829, година производње: 2016.
6. **Универзално мерило за испитивање заштите електричних инсталација**, произвођач: „METREL“ / Словенија, тип: „EUROTEST 61557“, Фабрички број: 14010066, година производње: 2014.
7. **Контактни и бесконтактни дигитални обртмер (тахометар)**, произвођач: „GAIN EXPRESS“ / Кина, тип: „DT – 6236B“, Фабрички број: T 430410, година производње: 2013.

8. Кљунасто помично уређај (шублер), од 0 – 200 mm, произвођач: „TOPEX“ / Пољска, Фабрички број: 12.0412, Година производње: 2017.
9. Микрометар за спољашње мерење, од 0 – 25 mm, произвођач: „WOMAX“ / Germany, Фабрички број: 0.572.591, Година производње: 2009.
10. Хронометар (штоперница), произвођач: „TFA“ Dostmann Wertheim / Germany, Фабрички број: 38.201602, Година производње: 2010.
11. Гарнитура тегова за статичка и динамичка испитивања, од 20.8 kg, произвођач: „Ливница Чока“, Чока / Србија, Фабрички број: 001 – 025, година производње: 2010.

ПРОТОКОЛ О ПРЕГЛЕДУ И ПРОВЕРИ				
ред. број	Протокол о прегледу и провери	Захтеви за лифтове стављене на тражиште Пре 1. Јануара 2014.	Провер а/ Мерењ а	Налаз Не/одговара
1.	Прилаз, диспозиција опреме и машинског простора, радне зоне; заштитна ограда око отвора у поду и покрета	Слободан и осветљен прилаз, прилаз агрегата, постојање канта, ГПП апарата и ед. изолационе, просторне ограда	П	Не одговара
2.	Врата машинског простора, вентилација и температура у машинском простору	Мogućност закључавања, отварање ван поља просторије, натпис на вратима	П/М	одговара
3.	Главни прекидач лифта, положај, функција	Функционалност, положај и обележавање	П	одговара
4.	Поговски агрегат, смештај, стање, погонска и помоћна ужетњача, издржљивост ужетљава, однос, пречника погонске ужетњаче и ужете	Приступачност, слободан простор, стање ужади и ужетљава, однос, капацитета пречника најмање 40	П/М	Не одговара
5.	Рад у хитним случајевима, возни или електрични уређај за померање кабине	За возну силу мању од 400 N – глатки точак без падова, ознака смера	П/М	одговара
6.	Заштита од обртних делова, заштитним направама или фарбањем у жуту боју	Заштита опасних приступачних, обртних делова лифта, обележавање жутом бојом	П	одговара

7.	Електромеханичка кочица – најмање две кочице безбедне папуче, стање облога, исправност направе за ручно откочивање	Могућност заустављања кабине са називним оптерећењем при кретању на доле са 125 % називне брзине	П/М	одговара
8.	Граничник брзине, приступачност, пломбирање и ознака последњег подешавања, стање ујета и жљебова, ујетаче	Граничник брзине доступан за преглед и одржавање, исправност погонског ујета и ујетаче, стање затезне ујетаче	П/М	Не одговара
17.	Врата кабине, зорови око врата, адекватни ревициони отвори код ручних врата	Контрола затворености врата, зорови око врата до 10 mm; Фотозавеса – без врата	П/М	одговара
18.	Рукохват, огласало, осветљење – опште и нужно, ознака положаја кабине у станици	Постојаност, функционалност: Бар два осветљена тела, осветљење изнад 50 lux	П/М	не одговара
30.	Заштита од индиректног додира је изведена аутоматским искључењем напајања TN развољним системом заштите	Одговарајућа вредност према номиналној струји осигурача: Отпор петље кvara у прописаним границама	П/М	не одговара
31.	Вега елемената лифта са громобранском инсталацијом	Сви елементи металне конструкције повезани са громобранском инсталацијом	П/М	Одговара
32.	Промене на лифту, у окружењу и код употребе лифта које могу да утичу на безбедност	Да ли су настале промене, оцена ризика према упутству за контролисање лифтова	П	Одговара
33.	Контрола вођења књиге одржавања лифта	Књига оформљена и редовно попуњавана	П	Одговара
Ред. Број	Протокол о прегледу и провери (чл. 6 и 17. Правилника о прегледима лифтова у употреби)	Посебни захтеви за преглед постојећих лифтова	Провера/Мерење	Налаз Не/одговара
34.	Постојаност адекватних врата кабине или оптичке детекције присуности лица	Испуњени битни захтеви безбедности за кабине постојећих лифтова	П	одговара
35.	Постојаност адекватних показивача или ознака положаја кабине у кабин. лифта	Функционалан показивач положаја кабине у станици или други вид ознаке	П	одговара
36.	Постојеће зајаве врата возног окна у складу са захтевима за стављање лифта и компоненти на трајнице	Испуњени сви битни захтеви за зајаве као безбедносне компоненте лифта	П	одговара
37.	Врста и стање управљачких команди лифта, тачност заустављања кабине у станицама	Управљачке команде лифта, приступ. Висока тачност заустављања кабине	П/М	одговара
38.	Опште стање граничника брзине	Граничник подешен, пломбиран	П	Не одговара

НЕУСАГЛАШЕНОСТИ	
- У кабин. лифта недостаје рукохват и противпанична расвета; - Поједини осигурачи у машинској просторији су неправо повезани – линовани; - Потребно је поставити нове осигураче одговарајуће ампераже; - Заменити зотрајале кулдуи: гуме на споју између редуктора и погонског агрегата; - Заменити зотрајале граничник брзине; - Око отвора у поду машинске просторије потребно је издати упозниче – цокле у висини од најмање 50 мм у циљу спречавања ненамерног пада предмета у возно окно; - Прилком рада лифта, примећен је појаван звук мотора: Потребно је установити узрок и по потреби, издати сервис.	
НАПОМЕНА: Ради усаглашавања лифта са важећим прописима неопходно је отклонити наведене неусаглашености у року од годину дана од дана издавања Извештаја о прегледу.	
ЗАКЉУЧАК	
На основу извршеног прегледа предметног лифта на локацији, тим је мишљења да је лифт НЕУСАГЛАШЕН са важећим референтним документима.	

4. ЗАКЉУЧАК

У данашњем урбаном окружењу примена лифта је све већа. Због тога безбедност постаје важна тачка разматрања у овом домену. Она захтева детаљне увиде у различите аспекте, укључујући техничке спецификације, прописе, стандарде, одржавања и управљања ризицима.

Да би коришћење лифтова било сигурно и дугорочно одрживо неопходно је спроводити редовне и ванредне прегледе, одржавања, као и придржавати се безбедносних процедура које су веома битне уколико желимо да обезбедимо поузданост и сигурност лифтова.

Преглед и провера лифтова није ни мало једноставна активност. То је сложен процес који захтева детаљну проверу функционалности свих компоненти, електричних система, сигурносних уређаја, али и проверу усаглашености са прописаним стандардима и прописима. У току прегледа и провере посебна се пажња посвећује безбедносним компонентама чија је сврха спречавање повреда у случају кvara и спасавање људских живота неких од кључних елемената. Важно је истаћи да због комплексности процеса прегледа, ове активности не може обављати било ко. Вршење прегледа и провере обављају стручна лица која су углавном из различитих кадрова као што су инжењери из области машинства и електро струке.

5. ЛИТЕРАТУРА

- [1] http://www.vmps.rs/elektricni_liftovi_sa_masinskom_prostorijom.htm (приступљено 07.12.2023.године)
- [2] Тошић С.: Лифтови, Машински факултет Универзитет у Београду, Београд, 2004.
- [3] <https://ba.aolida-lift.com/> (приступљено 14.12.2023.)
- [4] Жигић Н.: Q3.01.08.: Упутство за контролисање лифтова за превоз лица и терета на електрични погон, ВСС ЦОНТРОЛ ДОО, Нови Сад, 2017
- [5] Правилник о прегледима лифтова у употреби, „Сл. Гласник РС“, бр. 15/17
- [6] ВСС ЦОНТРОЛ ДОО, Q3.01.08-03: Извештај о прегледу лифта на електрични погон за вертикални превоз лица и терета, Нови Сад, 2017

Кратка биографија:



Цвијетин Пејчић рођен је у Бијељини 1996. год. Мастер рад на Факултету техничких наука из области Механизације и конструкционо машинство одбранио је 2024.год.
контакт: cilepejcic96@hotmail.com

VEZA IZMEĐU TVRDOĆE I STEPENA DEFORMACIJE KOD ZAPREMINSKOG DEFORMISANJA CILINDRIČNIM ALATOM**HARDNESS – STRAIN RELATION IN A BULK FORMING WITH A CYLINDRICAL TOOL**

Boško Vuković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – PROIZVODNO MAŠINSTVO

Kratak sadržaj – Delovi dobijeni hladnim deformisanjem metala odlikuju se povećanjem tvrdoće i čvrstoće, što je posledica deformacionog ojačavanja. U radu analizirana je promena tvrdoće po zapremini dela dobijenog hladnim deformisanjem cilindričnim alatom.

Ključne reči: Hladno deformisanje, Tvrdoća, Deformacija

Abstract – The main characteristic of cold formed metal parts is strain hardening which results in an increase of hardness and strenght of material. In the paper hardness distribution within a part formed by a cylindrical tool is presented.

Keywords: Cold forming, Hardness, Strain

1. UVOD

Glavna karakteristika delova koji se dobijaju hladnim deformisanjem metala jeste deformaciono ojačavanje, što znači da su čvrstoća i tvrdoća materijala na kraju procesa deformisanja povećane u odnosu na čvrstoću i tvrdoću pre procesa deformisanja. Povećanje čvrstoće i tvrdoće metala nakon plastične deformacije objašnjava se na sledeći način. Glavni mehanizam plastičnog deformisanja predstavlja kretanje dislokacija koje međusobno reaguju, obrazuju nepokretne dislokacije ili grupe dislokacija koje predstavljaju prepreke za njihovo dalje kretanje. Kako se povećava stepen deformacije povećava se broj prepreka i gustina dislokacija. To za rezultat ima povećanje opterećenja potrebnog za nastavak procesa deformisanja, koje u stvari predstavlja deformaciono ojačavanje [1,2].

Merenje tvrdoće materijala predstavlja jedan od najčešće primenjivanih testova za ispitivanje materijala. Tvrdoća se može okarakterisati i kao otpor materijala prema plastičnoj deformaciji [3]. Tokom testa utiskivanja ispod utiskivača formira se plastična zona u kojoj se postepeno smanjuje veličina plastične deformacije [4,5].

U radu su analizirane vrednosti tvrdoće po zapremini dela koji je dobijen u eksperimentu sabijanja cilindričnim alatom. Metoda je izvedena u hladnom stanju tako da je u procesu deformisanja došlo do deformacionog ojačavanja što je uticalo i na povećanje vrednosti tvrdoće u deformisanom delu.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Miroslav Dramićanin, docent.

2. EKSPERIMENTALNI DEO

U eksperimentalnom delu rada izvršena je provera zavisnosti između ostvarenog stepena deformacije i tvrdoće materijala. Deformisanje je ostvareno sabijanjem cilindričnim alatom, nakon čega je merena promena tvrdoće u zoni ispod delovanja žiga.

Za materijal je odabran konstrukcioni čelik S235. Na slici 1. prikazan je izgled uzorka pre sabijanja.



Slika 1. Izgled uzorka pre sabijanja

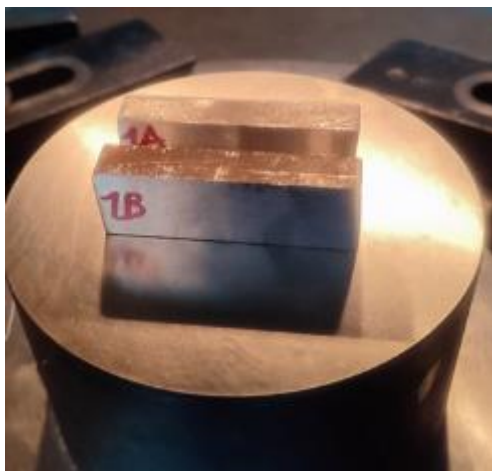
Mere uzoraka koji su ispitivani na sabijanje prikazane su u tabeli 1. Uzorci su sabijajni za različite stepene deformacije, odnosno za različite ostvarene hodove alata čije vrednosti su takođe prikazane u tabeli 1.

Tabela 1. Dimenzije uzoraka

Hod [mm]	Uzorak	Širina a [mm]	Visina b [mm]	Dužina l [mm]
2	1A 1B	11,35	20,04	59,20
		11,35	19,97	60,68
4	2A 2B	11,35	20,00	59,43
		11,35	20,05	58,46
6	3A 3B	11,35	20,01	55,75
		11,35	19,98	55,66
8	4A 4B	11,35	20,03	59,14
		11,35	19,97	56,85
10	5A 5B	11,35	20,04	50,12
		11,35	19,99	50,51

2.1. Eksperiment sabijanja

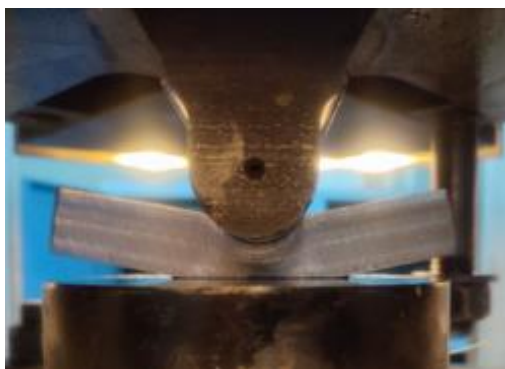
Alat korišten u eksperimentu sastojao se iz dva dela, donje ravne ploče (slika 2) i gornjeg cilindričnog žiga koji se utiskivao u materijal bočnom stranom cilindra (slika 3). Uzorci su sabijani za različite vrednosti hoda alata koji su prikazani u tabeli 1. Sam proces sabijanja (slika 4) je tekao na sledeći način. Po dva uzorka su postavljana u alat i sabijana za predviđenu vrednost hoda, pri čemu se sila očitavala na svakih 0,5 mm hoda.



Slika 2. Donja ploča alata



Slika 3. Utiskivanje žiga u materijal



Slika 4. Proces sabijanja uzorka

Nakon sabijanja svi uzorci imali su dodatnu deformaciju po širini usled tečenja materijala. Kako bi se izvršilo merenje promene tvrdoće po dubini uzorka, bilo je neophodno da se proširenje otkloni. Stoga se pristupilo obradi glodanjem da bi se uklonili krajevi koji su istisnuti,

a potom brušenju uzoraka, na mašini za brušenje, kako bi uzorci bili pripremljeni za merenje tvrdoće.

2.2. Metalografska ispitivanja

Metalografska priprema uzoraka ima tri osnovne faze: brušenje, poliranje i nagrivanje. Brušenje je izvedeno sa silicijum karbidnim brusnim papirima na uređaju za brušenje uzoraka sa dva diska. Korišteni su brusni papiri različite granulacije, od grublje ka finijoj. Prilikom brušenja uzorci su se konstantno hladili vodom kako bi se sprečila eventualna promena hemijskog sastava usled trenja između brusnog papira i površine metalnog uzorka i kako bi se odstranile nečistoće i produkti brušenja.

Nakon brušenja usledio je proces poliranja. Poliranje se izvodi tako što se na tkanine nanosi dijamantska suspenzija različitih granulacija, od veće ka manjoj. Uzorci su polirani sa svrhom uklanjanja ogrebotina nastalih brušenjem sve dok nije postignuta uglaćana površina uzoraka. Svaka faza brušenja i poliranja kontrolisana je pomoću svetlosnog mikroskopa.

Nakon što su uzorci ispolirani, isprani su vodom, očišćeni alkoholom i osušeni toplim vazduhom. Ispolirani i očišćeni uzorci prikazani su na slici 5.



Slika 5. Ispolirani i očišćeni uzorci

Nakon postupaka brušenja i poliranja, poslednja faza pripreme uzoraka je nagrivanje. Nagrivanje predstavlja razvijanje mikrostrukture materijala određenim hemijskim sredstvima kako bi mikrokonstituenti postali vidljivi. Uzorci su nagriveni nitalom (3% rastvor azotne kiseline, HNO_3 , u alkoholu). Na taj način uklonjeni su slojevi oksida ili nečistoća sa površine uzoraka i zrno je postalo vidljivo za analizu na mikroskopu.

Na pripremljenim uzorcima zabeležena je njihova mikrostruktura pomoću svetlosnog mikroskopa.

2.3. Merenje mikrotvrdoće

Na uređaju za ispitivanje mikrotvrdoće „Wilson Tukon 1102” vršeno je merenje tvrdoće po metodi Vikers. Opterećenje prilikom merenja je HV 1 kg, a utiskivač kod ovog postupka je četvorostrana dijamantska piramida sa uglom vrha od 136° .

Površina uzorka koji se ispituje treba biti ravna i glatka, kako bi se dijagonale otiska mogle tačno izmeriti. Nakon utiskivanja pritiskivača na površini uzorka ostaje kvadratni otisak čije se dijagonale mere pomoću uređaja, kao što je prikazano na slici 6. Sam uređaj na osnovu izmerenih dijagonala izračunava površinu utiskivanja.



Slika 6. Proces merenja tvrdoće i merenje dijagonala otiska

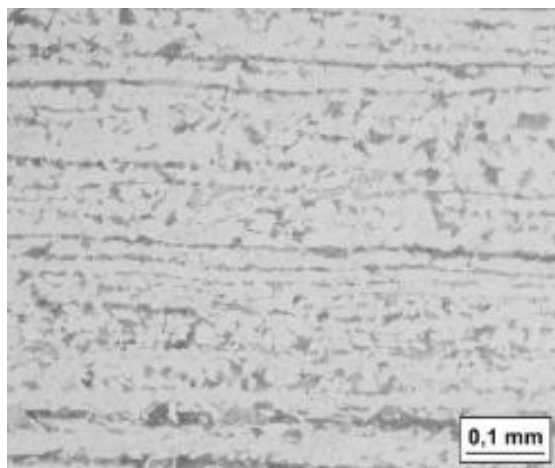


Slika 7. Šema merenja mikrotvrdoće

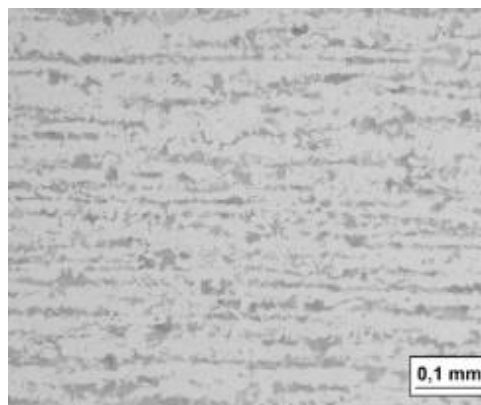
3. REZULTATI I DISKUSIJA

Analizom poliranih uzoraka ustanovljen je mali broj uključaka koji je dozvoljen u materijalu. Nakon nagrizanja poliranih uzoraka razvijena je mikrostruktura. Na slikama 8 i 9 prikazana je mikrostruktura u uzdužnom i poprečnom preseku uzorka, redom.

Uočava se trakasta struktura usled postupka dobijanja materijala. Suštinska razlika je u smeru valjanja, na slici 8 trake su duže i bez prekida, dok su na slici 9 trake češće isprekidane. Trakavost je dozvoljena u materijalu.



Slika 8. Uključci u uzdužnom preseku uzorka



Slika 9. Uključci u poprečnom preseku uzorka

U eksperimentu sabijanja registrovana je promena sile kod deformisanja svakog uzorka. Vrednosti koje su očitane sa prese, tačnije hod i sila prese za uzorak 2 prikazane su u tabeli 2.

Tabela 2. Promena sile u zavisnosti od hoda – uzorak 2

Hod [mm]	0,6	1	1,5	2	2,5	3	3,5	4
Sila [kN]	160	200	250	290	320	350	370	400

Na osnovu izmerenih visina uzoraka na mestu najmanje debljine nakon sabijanja pristupljeno je računanju logaritamskog stepena deformacije, φ . Logaritamski stepen deformacije izračunat je za svaki sabijeni uzorak pomoću izraza (1):

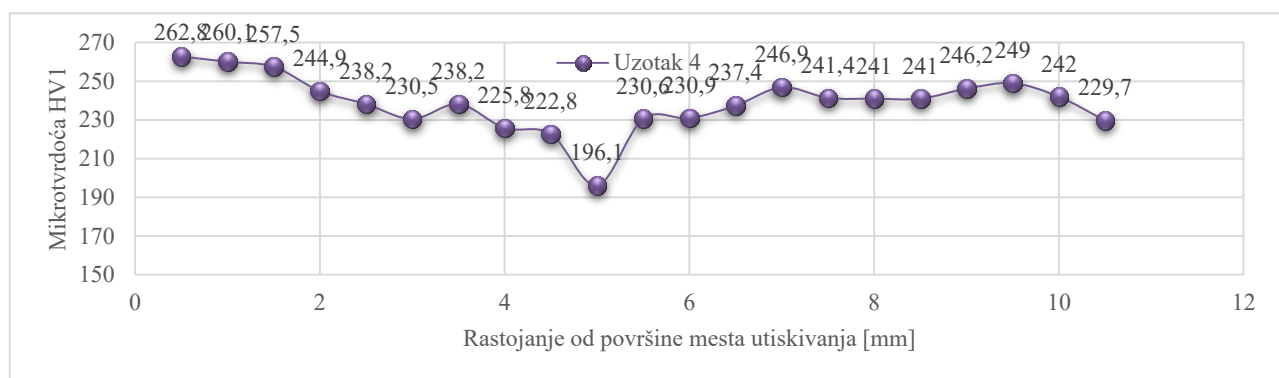
$$\varphi = \ln \frac{b_0}{b} \quad (1)$$

Dimezije uzoraka pre i posle deformisanja, kao i ostvareni stepeni deformacije prikazani su u tabeli 3.

Tabela 3. Stepni deformacije u eksperimentu sabijanja

Hod [mm]	Uzorak	Početna dimenzija uzorka [mm]	uzorka nakon sabijanja [mm]	Logaritamski stepen deformacije φ
2	1A	20,04	18,26	0,09
	1B	19,97		0,09
4	2A	20	16,10	0,22
	2B	20,05		0,22
6	3A	20,01	14,25	0,34
	3B	19,98		0,34
8	4A	20,03	12,15	0,50
	4B	19,97		0,50
10	5A	20,04	10,41	0,65
	5B	19,99		0,65

Ispitivanje mikrotvrdoće sprovedeno je metodom po Vickersu (opterećenje od 1 kg u vremenu od 10 sekundi). Na slici 10 prikazana su promene mikrotvrdoće po preseku uzorka merene na mestu najmanje debljine za uzorak. Srednja vrednost izmerene mikrotvrdoće osnovnog materijala pre ispitivanja iznosi 180 HV1.



Slika 10. Promena mikrotvrdoće po preseku uzorka merene na mestu najmanje debljine za uzorak 2

S obzirom da je proces deformisanja izveden u hladnom stanju materijal je deformaciono ojačao. To znači da je kod uzoraka sa većim stepenom deformacije očekivana veća čvrstoća i tvrdoća materijala. Takođe, u delu materijala koji je bliže alatu stepeni deformacije su veći pa samim tim i tvrdoća materijala. Rezultati merenja mikrotvrdoće za uzorke 1, 2, 4 i 5 se u najvećoj meri slažu sa očekivanim vrednostima. Odstupanje postoji kod uzorka broj 3 jer je kod njega dobijena manja mikrotvrdoća nego kod uzorka broj 2, kod koga je bio ostvaren manji stepen deformacije.

U tabeli 4 prikazan je odnos stepena deformacije i procenat porasta srednje mikrotvrdoće ispod mesta utiskivanja u odnosu na početnu vrednost.

Tabela 4. Odnos stepena deformacije i procenta porasta mikrotvrdoće ispod mesta utiskivanja u odnosu na početnu vrednost

Stepen deformacije	0,09	0,22	0,34	0,50	0,65
Porast mikrotvrdoće [%]	0,18	0,34	0,19	0,46	0,44

Jedan od razloga zašto mikrotvrdoća varira 0,5 mm ispod površine utiskivanja jeste u samom materijalu, odnosno za proračun je korišćena srednja vrednost tvrdoće osnovnog materijala koja je merena u sredini uzorka, dok zbog samog načina dobijanja čelične ploče mikrotvrdoća uz površinu materijala može varirati. Pre ispitivanja nije za svaki uzorak detaljno provereno kolika je mikrotvrdoća sa koje strane ploče. Kod uzorka 5 vidljiv je porast mikrotvrdoće od mesta utiskivanja do podloge, odnosno najveća mikrotvrdoća je izmerena 0,5 mm od same podloge, gde je takođe došlo do značajnih deformacija, što potvrđuje da je najverovatnije uz površinu ploče bila drugačija mikrotvrdoća.

4. ZAKLJUČAK

Jedna od bitnih karakteristika procesa plastičnog deformisanja koji se odvijaju u hladnom stanju jeste deformaciono ojačavanje materijala. Deformaciono ojačavanje rezultira povećanjem čvrstoće i tvrdoće oblikovanog dela, kao i smanjenjem plastičnosti i žilavosti.

U radu analiziran je uticaj deformacionog ojačavanja materijala na promenu mikrotvrdoće po zapremini deformisanog dela. Proces hladnog deformisanja izveden je na alatu koje se sastojao od donje ravne ploče i gornjeg

cilindričnog žiga koji se utiskivao u materijal bočnom stranom cilindra. Materijal korišten u eksperimentu bio je čelik S235.

Promena mikrotvrdoće merena je na poprečnom preseku uzorka na mestu njegove najmanje debljine. Mikrotvrdoće su merene na uzorcima koji su deformisani za različite stepene deformacije. Dobijeni rezultati u velikoj meri se slažu sa teorijskom pretpostavkom da će tvrdoća materijala rasti sa stepenom deformacije. Na osnovu međusobnog poređenja rezultata mikrotvrdoće, dobijene vrednosti su očekivane i u skladu sa deformacionim ojačavanjem materijala. Uočljiv je porast mikrotvrdoće koji odgovara porastu stepena deformacije.

Zbog uočenih problema u samom materijalu, odnosno blažeg rasipanja mikrotvrdoće uz površinu čeličnih ploča u odnosu na sredinu, pre budućih ispitivanja potrebno je uraditi meko žarenje materijala kako bi se te male razlike u mikrotvrdoći još više umanjile.

5. LITERATURA

- [1] L. Šidanić, "Mašinski materijali SV 2", FTN Novi Sad, 2014.
- [2] W. D. Callister, D. G. Rethwisch, "Materials science and engineering, An introduction", Wiley USA, 2014.
- [3] E. Broitman, "Indentation Hardness Measurements at Macro-, Micro-, and Nanoscale: A Critical Overview", *Tribology Letters*, Vol. 65, No. 1, 2017.
- [4] G. Sundararajan, Y. Tirupataiah, "The hardness-flow stress correlation in metallic materials", *Bulletin of Materials Science*, Vol. 17, No. 6, pp. 747–770.
- [5] D. Tabor, "The hardness of metals", Oxford at the Clarendon press, 1951.

Kratka biografija:



Boško Vuković rođen je u Trebinju 1996. godine. Diplomirao je na Fakultetu tehničkih nauka na smeru Proizvodno mašinstva, gde je nastavio školovanje na master studijama. Master rad je odbranio 2024. godine.
kontakt:boskovukovic0@gmail.com

ИСТРАЖИВАЊЕ УТИЦАЈА МОДЕЛА ТРЕЊА НА КАРАКТЕРИСТИКЕ ПРОЦЕСА КОМБИНОВАНОГ ИСТИСКИВАЊА ПУТЕМ НУМЕРИЧКИХ СИМУЛАЦИЈА**RESEARCH ON THE IMPACT OF FRICTION MODELS ON THE CHARACTERISTICS OF COMBINED EXTRUSION PROCESSES THROUGH NUMERICAL SIMULATIONS**

Балша Дачевић, Факултет техничких наука, Нови Сад

Област – МАШИНСТВО

Кратак садржај – У раду је испитиван утицај модела трења на процес комбинованог истискивања путем нумеричких симулација. Симулације су моделоване по узор на експеримент и њихови резултати су упоређени. Испитиван је утицај трења на геометријске карактеристике готовог дела и дијаграм сила ход.

Кључне речи: Контактно трење, модели трења, комбиновано истискивање, Метода коначних елемената

Abstract – In this paper characteristics of combined extrusion processes were researched through numerical simulations. Simulations were modelled after experimental setup and later their results were compared. Characteristics that were investigated are geometrical characteristics of finished part and Force-stroke diagram.

Keywords: Friction, friction models, combined extrusion, Finite element method

1. УВОД

Контактно трење представља један од најважнијих фактора процеса обраде деформисањем који у великој мери утиче на тачност и квалитет финалног дела, као и радни век алата. Како би се оптимално пројектовао процес обраде и правилно конструисао алат, његово познавање и квантитативно одређивање је од круцијалног значаја [1,2,3]. На основу бројних истраживања, развијени су различити теоријски модели који описују контактне услове на граничној површини алат-обрадак [4,5].

У овом раду је испитиван утицај модела трења на карактеристике процеса комбинованог истискивања. Нумеричким симулацијама, методом коначних елемената, добијени су резултати који су упоређени са експерименталним.

2. МОДЕЛИ КОНТАКТНОГ ТРЕЊА

Као последица релативног померања материјала у односу на површину алата контактнo трење има за последицу стварање тангенцијалних напона у приграничним слојевима материјала. Како би

се анализирано напонско стање у току процеса деформисања, веома је значајно питање квантитативне оцене и изражавања величине тангенцијалног напона услед контактнoг трења, односно обухватање утицаја трења на промену напонског стања [5].

У процесима обраде деформисањем се користе три закона, теорије, односно модела за математичко описивање контактнoг трења [5]:

- Кулонов модел,
- модел константног трења,
- адхезионе теорије.

У случају Кулоновог модела, тангенцијални напон се рачуна преко коефицијента трења μ , који представља однос тангенцијалног и нормалног напона [5].

Аналитички се изражава преко релације која је још позната и као Кулонов закон [1]:

$$\tau_k = \mu \cdot \sigma_n \quad (1)$$

где τ_k представља тангенцијални напон на контактнoј површини услед контактнoг трења а σ_n одговарајући нормални напон.

Модел константног трења се односи на изражавање тангенцијалног напона преко константног фактора смицања и максималног тангенцијалног напона. Релација којом се аналитички изражава овај модел јесте [5]:

$$\tau_k = m \cdot \tau_{max} \quad (2)$$

где τ_k представља тангенцијални напон на контактнoј површини услед контактнoг трења, m фактор смицања, односно трења и τ_{max} максимални тангенцијални напон.

Претходна два закона немају генералну вредност, и не “покривају” области средњих нормалних притисака. Управо зато су развијене адхезионе теорије које представљају групу модела који узимају у обзир реалну хрпавост контактних површина и покривају све области притисака [4].

Wanheim је развио универзални модел који покрива све области притисака који се иначе назива и општи модел. Аналитички се изражава [4]:

$$\tau = f \cdot \alpha \cdot k \quad (3)$$

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Младомир Милутиновић, ред. проф.

где је τ тангенцијални напон на контактної површини услед контактног трења, f фактор трења, $0 \leq f \leq 1$, α однос реалне зоне контакта и рачунске површине контакта, k напон течења.

3. МЕТОДА КОНАЧНИХ ЕЛЕМЕНАТА

С обзиром да експерименталне методе испитивања генерално изискују пуно времена, нумеричке симулације представљају веома користан и ефикасан алат за предвиђање проблема у индустријској производњи уз смањење времена и трошкова развоја. Основна предност нумеричких симулација јесте могућност извођења “Шта ако” симулација, где се без додатних трошкова могу испитати различити сценарији и пробати алтернативна пројектна решења [4].

Основна идеја која лежи иза методе коначних елемената јесте проналажење решења компликованих проблема заменом за једноставнији. Како се уводе овакве измене проналажење егзактног решења није могуће, већ добијени резултат представља апроксимирано решење [6].

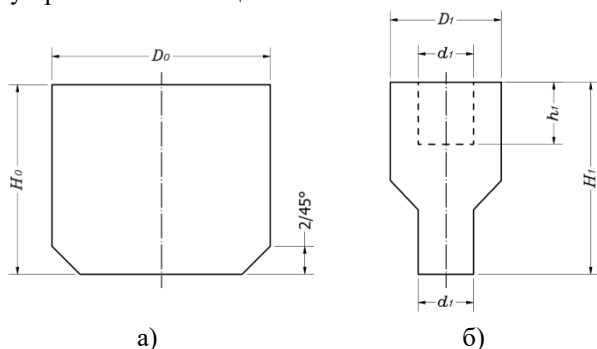
Ова метода спада у методе дискретне анализе и заснива се на физичкој дискретизацији посматраног домена. Принцип на коме се базира се састоји од раздвајања посматраног континуума (са бесконачним бројем степени слободе) на ограничен број међусобно повезаних делова (са ограниченим бројем степени слободе), који се називају коначни елементи. [4, 7].

Све нумеричке симулације у овом раду су урађене у програмском пакету Simufact Forming 12 који се користи за анализу процеса пластичног деформисања методом коначних елемената.

4. ПРЕГЛЕД РЕЗУЛТАТА ЕКСПЕРИМЕНТА

Експериментални део рада је одрађен у оквиру Мастер рада [8]. За истискивање узорака је коришћена хидраулична преса Sack&Kiesselbach номиналне силе 6,3 MN и две матрице различите геометрије једне са углом конуса од 30° и друге са углом конуса од 90°. Помоћу сваке матрице истиснута су четири узорка са различитим ходовима алата: 9mm, 12mm, 15mm и 18mm.

Карактеристичне димензије припремка и готовог дела су приказани на слици 1.



Слика 1. а) Припремак б) готов деоа [8]

Резултати промене деформационе силе у зависности од хода алата су добијени у (*.xls) формату и они су

приказани у компарацији са резултатима добијеним путем нумеричких симулацијама

5. ПРИПРЕМА НУМЕРИЧКЕ АНАЛИЗЕ

Први корак у припреми нумеричких симулација јесте дефинисање врсте обраде, након чега је извршен унос геометрије алата у програм за вршење симулација.

С обзиром да се ради о алатима са конусним површинама њихово моделирање је урађено у CAD програму SolidEdge 2022. Конверзија (*.par) формата модела у (*.stl) формат је извршена у бинарном облику са фином толеранцијом конверзије од 0,001mm.

Матрице и жиг су моделовани као апсолутно крута тела услед чега није потребно извршити избор материјала. Материјал узорка је Ч1531 (C45E) чије је дефинисање за потребе нумеричких симулација извршено мануелно.

Крива течења [8] је добијена рачунском методом и њен облик је приказан испод:

$$K = 289,67 + 597,58 \cdot \varphi^{0,333} \quad (4)$$

Остале неопходне карактеристике материјала су приказане у табели 1.

Табела 1. Карактеристике материјала

Модул еластичности [MPa]	210000
Густина материјала [kg/m³]	7800
Poisson-ов коефицијент	0,3

Коефицијенти и фактори трења који су коришћени у симулацијама су приказани у табели 2.

Табела 2. Модели трења и вредности коефицијента/фактора трења

Кулонов модел трења, μ	0,08; 0,11; 0,2
Модел константног трења, m	0,14; 0,19; 0,35
Комбиновани модел трења, μ	0,08 + 0,14; 0,11 + 0,19; 0,08 + 0,14

Топлотни параметри су приказани у табели 3. док је температура околине подешена на 20°C.

Табела 3. Топлотни параметри

Топлотни параметар	Алати	Обрадак
Почетна температура [°C]	20	20
Коефицијент преноса топлоте на околину [W/m²·K]	50	50
Коефицијент преноса топлоте на обрадак [W/m²·K]	20000	/
Емисиона способност	0,25	0,25

Дискретизација модела узорка је урађена са коначним елементом типа Quad(10), односно квадратним коначним елементом са десет чворова. Формирање мреже је извршено на основу величине коначног елемента која је износила 0,2 mm па је број елемената након дискретизације износио 6673 елемента

Последњи корак пре почетка симулација јесте дефинисање хода алата који износе 9mm, 12mm, 15mm и 18mm.

6. ПРЕГЛЕД РЕЗУЛТАТА НУМЕРИЧКИХ СИМУЛАЦИЈА

6.1. Праћење геометријских промена

У табелама 4 и 5 су приказани добијени резултати за димензију h_1 заједно са експерименталним за оба типа матрице.

У случају матрице са углом конуса од 30° , видимо да се при ходу алата од 9mm јављају знатна одступања у односу на експериментално добијени резултат, где су се ниже вредности коефицијента и фактора трења показале тачније резултате.

За вредности хода од 12, 15 и 18mm најприближнији резултати су добијени коришћењем модела константног трења за вредност фактора $m=0,14$ и коришћењем комбинованог модела за вредности $\mu=0,08$ и $m=0,14$.

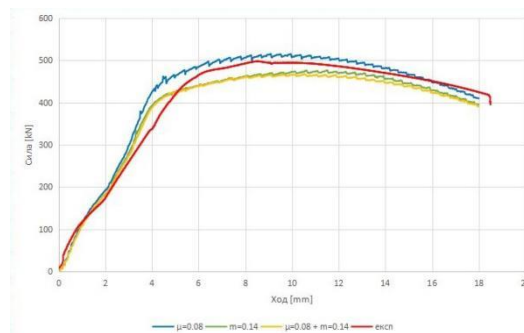
Највећа одступања су се јавила у случају коришћења Кулоновог модела трења, где су добијене вредности у свим случајевима биле више у односу на друга два модела. Може се закључити да у случају матрице са углом од 30° стварним контактним условима боље одговарају ниже вредности трења.

У случају матрице са углом конуса 90° , ситуација је знатно другачија. При ходовима алата 9mm, 12mm и 15mm експериментално измерена вредност је виша од оних добијених симулацијом и приближније су јој оне којима одговарају веће вредности трења.

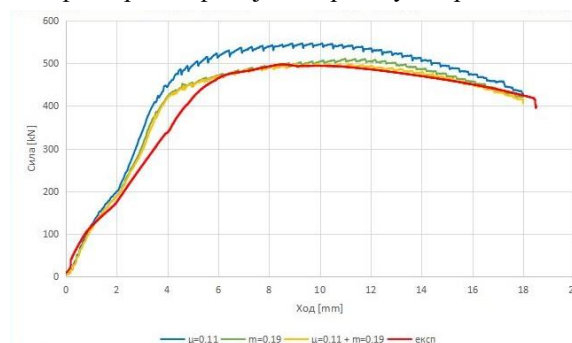
У процесу са ходом алата од 18mm најприближнији резултати су добијени коришћењем најнижих вредности коефицијента и фактора, где је модел константног трења за вредност $m=0,14$ пружио најтачнији резултат.

6.2. Дијаграм сила ход

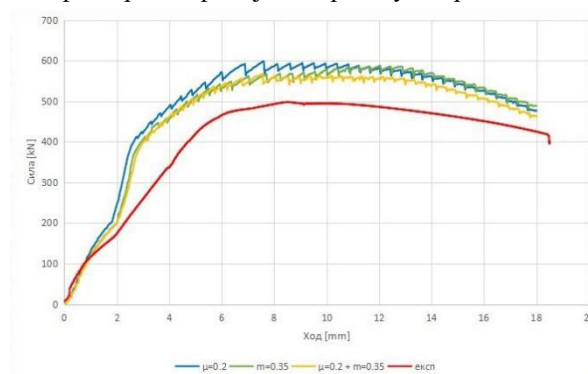
На сликама 2, 3 и 4 приказани су дијаграми сила ход за различите моделе трења при ходу алата од 18mm у матрици са углом од 30° .



Слика 2. Сила ход дијаграм за најниже вредности фактора/коефицијента трења у матрици од 30°



Слика 3. Сила ход дијаграм за средње вредности фактора/коефицијента трења у матрици од 30°



Слика 4. Сила ход дијаграм за највише вредности фактора/коефицијента трења у матрици од 30°

Табела 4. Димензија h_1 при употреби матрице са углом конуса од 30°

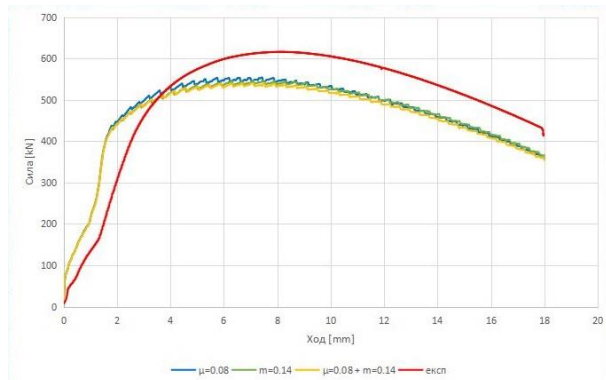
	h_1									
Ход	$\mu=0.08$	$\mu=0.11$	$\mu=0.2$	$m=0.14$	$m=0.19$	$m=0.35$	$\mu=0.08 + m=0.14$	$\mu=0.11 + m=0.19$	$\mu=0.2 + m=0.35$	Експ.
9mm	4.47	5.512	7.093	3.182	4.153	6.576	3.138	4.098	6.524	1.57
12mm	6.757	7.796	9.904	5.318	6.435	9.181	5.241	6.361	8.99	4.73
15mm	9.084	10.12	12.407	7.629	8.763	11.692	7.54	8.702	11.354	7.23
18mm	11.393	12.451	14.76	9.973	11.085	14.006	9.877	11.03	13.696	8.46

Табела 5. Димензија h_1 при употреби матрице са углом конуса од 90°

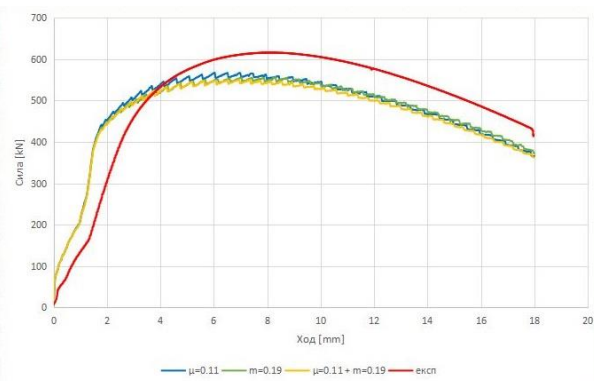
	h_1									
Ход	$\mu=0.08$	$\mu=0.11$	$\mu=0.2$	$m=0.14$	$m=0.19$	$m=0.35$	$\mu=0.08 + m=0.14$	$\mu=0.11 + m=0.19$	$\mu=0.2 + m=0.35$	Експ.
9mm	7.717	8.002	8.304	7.562	7.888	8.286	7.496	7.8	8.246	9.39
12mm	10.336	10.682	11.144	10.156	10.542	11.104	10.068	10.434	11.018	12.14
15mm	12.95	13.341	13.892	12.796	13.228	13.952	12.673	13.074	13.73	14.07
18mm	15.575	15.96	16.648	15.447	15.903	16.82	15.299	15.71	16.465	15.46

Најприближнији резултати сила ход дијаграма су добијени коришћењем Кулоновог модела за вредност коефицијента трења $\mu=0,08$, модела константног трења за вредност фактора трења $m=0,19$ и коришћењем комбинованог модела за вредности $\mu=0,11$ и $m=0,19$. Уз то да је приликом употребе константног и комбинованог модела дошло до скоро идентичних криви.

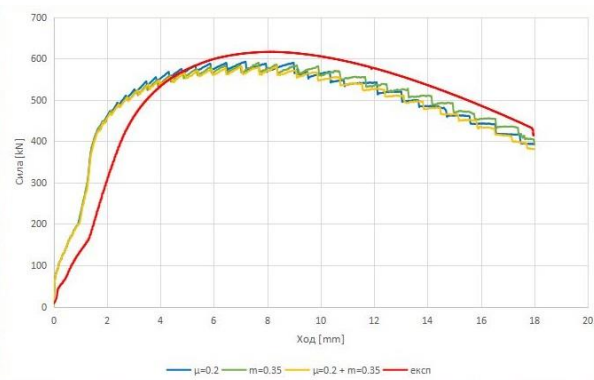
На сликама 5, 6 и 7 су приказани дијаграми сила ход за различите моделе трења при ходу алата од 18mm у матрици са углом од 90°.



Слика 5. Сила ход дијаграм за најниже вредности фактора/коефицијента трења у матрици од 90°



Слика 6. Сила ход дијаграм за средње вредности фактора/коефицијента трења у матрици од 90°



Слика 7. Сила ход дијаграм за највише вредности фактора/коефицијента трења у матрици од 90°

Код матрице са углом конуса од 90° видимо да се приликом коришћења еквивалентних коефицијената и фактора трења добијају веома слични дијаграми сила-ход.

Најприближније вредности су добијене коришћењем највиших вредности коефицијента/фактора трења.

7. ЗАКЉУЧАК

Модели су се при еквивалентним вредностима фактора и коефицијента генерално понашали уједначено. Уочена већа одступања могу бити услед одступања модела од реалних услова експеримента.

Што се тиче димензионих карактеристика, приликом испитивања је уочено да најрелевантније резултате пружа комбиновани модел. Модел константног трења је дао нешто мању тачност у односу на њега, док су коришћењем Кулоновог модела уочена највећа одступања. Најбоља поклапања су добијена приликом коришћења комбинованог модела трења за вредности фактора и коефицијента трења $\mu=0,08 + m=0,14$.

Током испитивања дијаграма сила-ход видимо да су се у матрици од 30° тачније показале ниже вредности коефицијента односно фактора трења, док је код матрице од 90° ситуација супротна.

8. ЛИТЕРАТУРА

- [1] M. Plančak, D. Vilotić, “*Tehnologija plastičnog deformisanja*”, Novi Sad, Fakultet tehničkih nauka, 2012.
- [2] M. Plančak, Z. Car, M. Kršulja, D. Vilotić, I. Kačmarčik, D. Movrin, “*Possibilities to Measure Contact Friction in Bulk Metal Forming*”, Technical Gazette, 19(4), pp.727–734, 2012.
- [3] Đ. Čupković, M. Plančak, D. Vilotić, K. Kuzman, “*Uticaј kontaktnog trenja na proces dvostrukog suprotnosmernog istiskivanja.*”, Kragujevac Srbija i Crna gora, YUTRIB’05 9ta jugoslovenska konferencija o tribologiji, 2005.
- [4] V. Mandić, “*Fizičko i numeričko modeliranje procesa obrade deformisanjem*”, Kragujevac, Fakultet inženjerskih nauka, 2012.
- [5] V. Vujović, “*Obrada deformisanjem u mašinstvu: I deo*”, Novi Sad, Fakultet tehničkih nauka, 1990.
- [6] S:S. Rao, “*The Finite Element Method in Engineering*”, Elsevier, 2011.
- [7] F. Klocke, “*Manufacturing Processes 4 Forming*”, Berlin, Heidelberg Springer, 2013.
- [8] D. Čizmić, “*Istraživanje toka materijala pri kombinovanom istiskivanju u funkciji geometrije matrice*”, Novi Sad, Fakultet tehničkih nauka, Master rad 2014.

Кратка биографија:



Балша Дачевић рођен је у Новом Саду 1996. год. Основне студије завршио је на Факултету техничких наука 2019. године. Мастер студије уписује исте године на Факултету техничких наука у Новом Саду на департману за производно машинство, смер Савремене технологије обликовања материјала. Тренутно је запослен у Hillmeg Engineering-у на позицији конструктора кочионих система за лучке кранове.
контакт: bdacevic96@gmail.com

ИЗБОР И ПРОРАЧУН МАШИНСКЕ ОПРЕМЕ ЦРПНЕ СТАНИЦЕ ЗА ДРЕНАЖУ ЕНЕРГЕТСКЕ ГАЛЕРИЈЕ УЗВОДНЕ ГЛАВЕ БРОДСКЕ ПРЕВОДНИЦЕ ХЕ ЂЕРДАП 1
SELECTION AND CALCULATION OF MECHANICAL EQUIPMENT FOR THE PUMP STATION FOR DRAINAGE OF THE ENERGY ROOM OF THE UPSTREAM HEAD OF THE SHIP LOCK OF HPP DJERDAP 1

Петар Ивановић, Факултет техничких наука, Нови Сад

Област – МАШИНСТВО

Кратак садржај – У раду је приказан начин избора и хидраулички прорачун машинске опреме црпне станице за дренажу енергетске галерије узводне главе бродске преводнице ХЕ Ђердап 1. Ова црпна станица штити од потапања електро-хидрауличке инсталације сервомотора врата и затварача узводне главе преводнице. У оквиру рада укратко су дате теоријске основе рада пумпи, извршен је хидраулички прорачун, димензионисана је машинска опрема црпне станице и дефинисан је начин аутоматског управљања радом постројења.

Кључне речи: *Потапајуће пумпе, хидраулички прорачун, бродска преводница.*

Abstract – The paper shows the method of selection and the hydraulic calculation of the mechanical equipment of the pump station for the drainage of the energy room of the upstream head of the ship lock HPP Djerdap 1. The pump station prevents flooding of the electro-hydraulic installations of the servo-drive of the upstream gate and the valve. Within the paper are given brief theoretical background of pump operation, hydraulic calculation, selection of mechanical equipment of the pump station and automatic control of the pump station is specified.

Keywords: *Submersible pumps, hydraulic calculation, ship lock.*

1. УВОД

Хидроенергетски и пловидбени систем „Ђердап 1“ је комплексан и вишенамјенски објекат изграђен на 943. километру Дунава од ушћа у Црно море. То је највећа хидротехничка грађевина на Дунаву, укупне дужине 1.278 m. Потпуно је симетрична и пројектована тако да свака земља (Србија и Румунија) располаже са истим дјеловима главног објекта: по једном хидроелектраном и бродском преводницом и по 7 преливних поља. Обе хидроелектране су међусобно повезане тако да у случају потребе агрегати електране на српској страни могу да испоручују електричну енергију у мрежу на румунској страни и обратно [1]. Основни технички подаци ХЕ Ђердап 1 (српска страна):

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Слободан Ташин, ванр. проф.

- рачунски нето пад - 25,8 m,
- инсталирани проток – 5040 m³/s,
- укупна активна снага електране – 1140 MW,
- просечна годишња производња - 5.636 GW/h.

Сем хидроелектране хидроенергетски систем Ђердап 1 обухвата и читав низ црпних станица чија је функција да сниже ниво подземних вода узводно од електране. Наиме, изградњом ХЕ Ђердап 1 створен је успор који је потпуно промјенио природан режим подземних вода на приобалном подручју. Због тога је уз ријеку Дунав, све до Новог Сада, изграђен читав низ сервисних бунара са пумпама чија је једина намјена да црпе подземну воду и препумпавају је у Дунав, чиме се обезбјеђује снижавање нивоа подземних вода. Контролом нивоа подземних вода обезбјеђује се пољопривредна производња у приобалним подручјима који би били поплавлени без ових црпних станица.

У склопу објекта ХЕ Ђердап 1 изграђене су две бродске преводнице којима је регулисан пловни саобраћај из акумулације у реку и обратно. Грађење преводница значило је савлађивање висинске разлике од 21 m до 34 m коју одређује доток Дунава и висина бране. Преводнице обе стране бране су двостепене са коморама дужине од по 310 m и ширине 34 m. Преводнице функционишу на принципу изједначавања нивоа у коморама и горње и доње воде. Бочне стране комора чине масивни бетонски зидови, а попречне двокрилне гвоздене капије, од којих је једна увијек затворена. У току једног циклуса преводница може да пропусти (преведе) бродски конвој максималне дужине 290 m и ширине 33 m за 90 минута. По својој величини преводнице на ХЕ Ђердап 1 спадају међу највеће на свјету, веће чак и од преводница на Панамском каналу [5].

2. ТЕХНИЧКО РЈЕШЕЊЕ ЦРПНЕ СТАНИЦЕ ЗА ДРЕНАЖУ ЕНЕРГЕТСКЕ ГАЛЕРИЈЕ

Предмет рада је црпна станица која служи за дренажу процурјеле воде и заштиту од потапања енергетске галерије - коначна експлоатација и МНУ просторије на узводној глави бродске преводнице. Опрема црпне станице смјештена је у одвојеној просторији у склопу просторије МНУ у којој је смјештена електро и хидрауличка инсталација обалних сервомотора врата и затварача узводне главе бродске преводнице. Постојећа црпна станица дренирала је галерију на коти 40,80 m „енергетска галерија – привремена експлоатација“. Ова галерија се више не користи, потопљена је, тако да ће

се новом црпном станицом штити виша „енергетска галерија - на коти 54,00 m“. Постојећа концепција црпне станице, заснована на потапајућим пумпама са моторима на сувом и дугим вратилима смештеним у потисним цјевоводима пумпи – напушта се. Постојеће пумпе се демонтирају, а уместо њих уграђују се нови пумпни агрегати који треба да остваре пројектовани капацитет од по 12 L/s. Постојећи цјевоводи пумпи демонтирају се све до прикључне прирубнице (кота 67,30 m) у плафону цијевне галерије на коти 65,00 m. У реконструисаној црпној станици уграђују се двије пумпе типа ABS JC 54 HD 50 Hz. Пумпе погоне електромотори снаге од по 5,8 kW, номиналне брзине обртања 2850 min⁻¹. Ради се о мобилним, потапајућим пумпним агрегатима намјењеним за дренажу шахтова и других објеката у којима се налазе прљаве и муљевите воде. Нове пумпе су центрифугалне, са полуотвореним радним колом. Процесни прикључак пумпних агрегата је спољашњи цијевни навој 3". На овај прикључак, потребно је надоградити прирубничку везу са прирубницом DN 80, PN 10, према EN 1092-1.

Нове пумпе димензионисане су за заштиту од плавења енергетске галерије на коти 54,00 m.

3. НАЧИН РАДА ЦРПНЕ СТАНИЦЕ И ХИДРАУЛИЧКИ ПРОРАЧУНИ

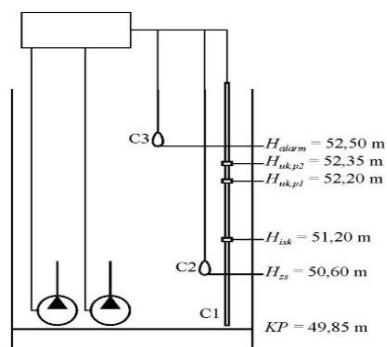
3.1 Референтни нивои црпљења

У црпној станици инсталисане су двије пумпе. Пумпе раде наизмјенично, једна се користи као радна а друга као резервна пумпа. Током експлоатације радна пумпа мијења статус у резервну и тако се успоставља равнотежа радних сати обе пумпе.

По потреби, у екстремним случајевима када радна пумпа нема довољан капацитет да евакуише воду из енергетске галерије обе пумпе могу да раде истовремено у паралелној спрези.

Обе пумпе раде у аутоматском режиму у зависности од нивоа воде у црпилишту али се по потреби могу активирати и деактивирати ручно. Када се систем налази у ручном режиму рада, процедуре за заштиту пумпи од рада на суво остају активне као и прекострујне заштите погонских мотора.

Референтни нивои воде у црпилишту црпне станице, по којима се аутоматски управља радом пумпи дати су на слици 1.



Слика 1. Референтни нивои воде у црпилишту црпне станице

Референтни нивои су:

$H_{uk.p1} = 52,20$ m – ниво стартовања радне пумпе,

$H_{uk.p2} = 52,35$ m – ниво стартовања резервне пумпе,
 $H_{alarm} = 52,50$ m – алармни ниво воде на којем се укључују обе пумпе без обзира на тренутни статус,
 $H_{isk} = 51,20$ m - ниво заустављања обе пумпе,
 $H_{isk.pr} = 40,70$ m - ниво заустављања обе пумпе, привремена експлоатација,
 $H_{zs} = 50,60$ m - ниво воде на којем реагује заштита пумпе од рада на суво,
 $Z_{C0} = 70,20$ m – кота излива потисног цјевовода,
 $KP = 49,85$ m – кота пода галерије на којем су постављене пумпе.

Потребне величине које је неопходно мјерити или регистровати да би се обезбједио аутоматски рад црпне станице су:

1. ниво воде у црпилишту,
 - континуални сигнал о нивоу воде (сонда C1 на сл. 1),
 - сигнал о достизању нивоа заштите пумпи од рада на суво (сонда C2)
 - сигнал о достизању алармног нивоа воде (сонда C3)
2. вријеме рада сваке од пумпи.

3.2 Алгоритам рада црпне станице

У аутоматском режиму рада пумпе у црпној станици се управљају по нивоима воде у црпилишту према следећим процедурама /поступцима

1. Поступак стартовања радне пумпе

Овај се поступак покреће кад ниво воде у црпилишту достигне коту $H_{uk.p1}$ која се детектује преко сонде C1. Достизањем ове коте стартује се радна пумпа.

2. Поступак нормалног заустављања радне пумпе

Овај се поступак покреће кад ниво воде у црпилишту достигне коту H_{isk} која се детектује преко сонде C1. Достизањем ове коте зауставља се радна пумпа односно обе пумпе ако су у погону.

3. Поступак интервентног заустављања радне пумпе

Интервентно заустављање радне пумпе покреће се у случају појаве ниског нивоа воде у црпилишту (заштита од рада на суво) и у случају прегријевања погонских електромотора пумпи..

Заштита од рада на суво се покреће када ниво воде у црпилишту опадне на коту H_{zs} која се детектује преко сонде C2. Радне пумпе се зауставља и може поново да стартује тек када сонда C1 детектује да је ниво воде у црпилишту достигао коту $H_{uk.p1}$ односно када сонда C3 детектује да је достигнут алармни ниво воде H_{alarm} .

Заштита погонског електромотора од прегријевања је решена као прекострујна заштита. Радна пумпа се зауставља и брише са листе радних све док се не изврши интервенција односно не отклони узрок прегријевања мотора.

4. Покретање друге пумпе

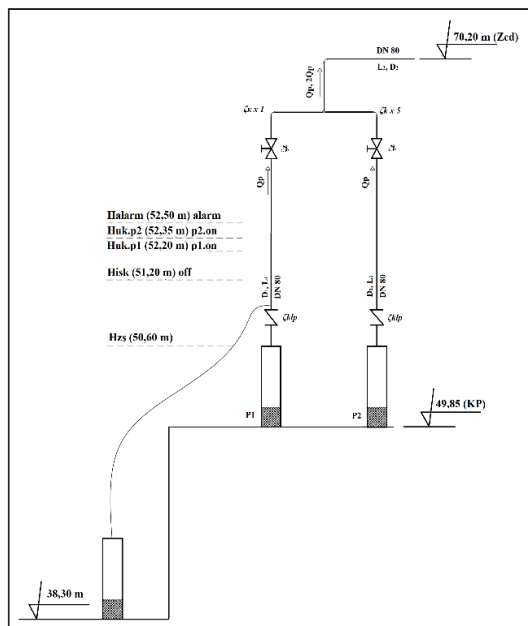
Овај се поступак покреће кад ниво воде у црпилишту достигне коту $H_{uk.p2}$ која се детектује преко сонде C1. Достизањем ове коте покрећу се обе пумпе, без обзира на тренутни статус.

5. Поступак интервентног стартовања пумпи - аларм

Уколико ниво воде у црпилишту достигне коту H_{alarm} , коју региструје сонда С3 (сонда С1 у квару) одмах се у погон укључују обе пумпе.

3.3 Хидраулички прорачун

Хидраулички прорачун за избор пумпи у црпној станици за дренажу енергетске галерије разматран је за случајеве када ради једна пумпа и када раде обе пумпе на галерији на коти $KP = 49,85$ m, као и случај привремене експлоатације када ради једна пумпа на коти 38,30 m (слика 2).



Слика 2. Шематски приказ црпне станице

Потребан напор пумпних агрегата добија се примјеном енергијске једначине за случај рада једне пумпе (1) и паралелне спреге двије пумпе (2):

$$H_{c.1p} = (Z_C - H_i) + \frac{8Q_1^2}{g\pi^2 d_1^5} \left(\lambda_1 \frac{L_1}{d_1} + 5\zeta_k + \zeta_{kl} + \zeta_z \right) + \frac{8Q_2^2}{g\pi^2 d_2^5} \left(\lambda_2 \frac{L_2}{d_2} + \zeta_k + 1 \right) \quad (1)$$

$$H_{c.2p} = (Z_{Cd} - H_i) + \frac{8Q_p^2}{g\pi^2 d_1^5} \left(\lambda_1 \frac{L_1}{d_1} + 5\zeta_k + \zeta_{kl} + \zeta_z \right) + \frac{8(2Q_p)^2}{g\pi^2 d_2^5} \left(\lambda_2 \frac{L_2}{d_2} + \zeta_k + 1 \right) \quad (2)$$

где H_i узима одговарајуће вредности нивоа воде у црпилишту на којима се пумпе укључују односно искључују.

Коефицијенти локалних отпора у пумпним цјевоводима имају следеће вриједности [2], [3], [4]:

- кољено DN 80, $R/D = 1,5$ – $\zeta_k = 0,4$,
- неповратни кугла вентил DN80 – $\zeta_{kl} = 1,5$,
- засун DN 80 – $\zeta_z = 0,3$.

Дужине, пречници, храпавости цјевовода и срачунати коефицијенти тријења при задатом прорачунском протоку од 12 L/s дати су у табели 1.

Потребан напор пумпе при задатом протоку од 12 L/s добија се из једначине (1) за ниво воде H_{isk} на којем се пумпа искључује и износи:

Табела 1. Прорачун линијских губитака у цјевоводима

	L (m)	OD (mm)	Q (L/s)	v (m/s)	e (mm)	λ
Пумпни цјевовод	17	88,9	12	2,12	0,2	0,026
Заједнички цјевовод	15	88,9	12	4,24	0,2	0,025
Допунски цјевовод	10	75	12	2,72	0,2	0,026

$$H_p = 22,4 \text{ m}$$

На основу овако срачунатог потребног напора пумпе усвојена је пумпа „ABS JC 54 HD“ следећих техничких карактеристика.

- проток 12 L/s,
- напор 24,5 m,
- снага мотора 5,8 kW,
- дозвољени број покретања 30/h.

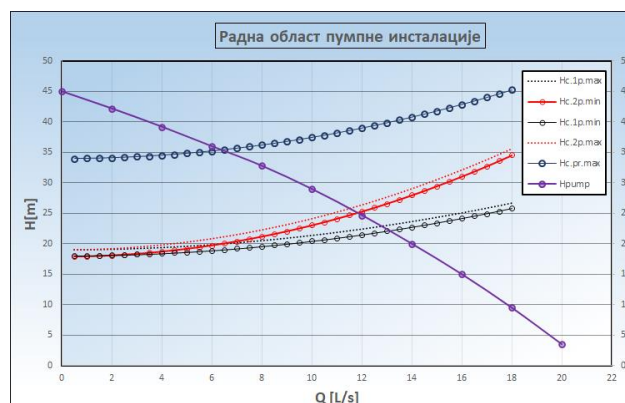
Ради се о мобилним потапајућим пумпним агрегатима који су намијењени за транспорт процурјелих вода, вода које садрже мање механичке нечистоће и муљ, отпадних вода и других које садрже мање количине суспендованих чврстих материјал.

Пумпе су центрифугалне са полуотвореним радним колом. Процесни прикључак пумпе остварује се са спољашњим навојем 3". На овај прикључак се спаја прирубница DN 80 са навојем.

Препоручљиво је да се због саме намјене пумпе врши периодична провјера уписних решетки пумпи једном у 3 мјесеца. Услијед нагомилавања нечистоћа, алги и других материјала на уписној решетки, може доћи до смањења протока те угрожавања саме намјене црпне станице.

На слици 3, приказане су карактеристике цјевовода за све предвиђене експлоатационе радне режиме и радна Q-H карактеристика изабране пумпе.

Пресејци карактеристика цјевовода и Q-H карактеристике пумпе дефинишу радну област изабраних пумпи, односно експлоатационе капацитете црпне станице за дренажу енергетске галерије.



Слика 3. Радна област пумпне инсталације за једну пумпу у погону, двије пумпе у паралелној спрузи и једну пумпу у режиму привремене експлоатације

Са слике 3. може се закључити да ће једна изабрана пумпа у самосталном раду у нормалној експлоатацији остваривати капацитет од око 13 L/s, што је нешто више од захтијевањег капацитета црпне станице.

Двије пумпе у паралелној спрези оствариваће капацитет од $2 \times 11,5 \text{ L/s} = 23 \text{ L/s}$, док ће једна пумпа у режиму привремене експлоатације остваривати проток од минимум 6 L/s (у тренутку искључења).

3.4 Број укључења пумпи на сат

За црпне станице које су предвиђене за црпљење воде из црпилишта дефинисане геометрије, у којима пумпе раде у аутоматском режиму према задатим нивоима воде у црпилишту, потребно је да се изврши прорачун броја укључења пумпе на сат, како би се избегло прегријевање погонског мотора и испад пумпног агрегата из погона.

У наставку се даје прорачун броја укључења изабраних пумпних агрегата у нормалној експлоатацији, када ради једна пумпа и када су на располагању обе пумпе. За референтну запремину црпљења узима се површина црпилишта на мјесту постављања пумпних агрегата и референтна висина црпљења.

- површина црпилишта:

$$A_C = 1,4 \text{ m} \times 1,6 \text{ m} = 2,24 \text{ m}^2 \quad (3)$$

- референтна висина црпљења (разлика нивоа воде у црпилишту од стартовања до заустављања пумпе):

$$\Delta Z_C = H_{uk,p1} - H_{isk} = 1 \text{ m} \quad (4)$$

- референтна запремина црпљења:

$$V_C = A_C \Delta Z_C = 2,24 \text{ m}^3. \quad (5)$$

Укупно трајање једног циклуса између два укључења пумпе једнако је збиру вријемеа потребног да пумпа у погону испразни запремину црпљења и вријемеа потребног да се та запремина поново напуни дотоком q :

$$T(q) = t_1(q) + t_2(q) \quad (6)$$

Вријеме прањњења запремене црпљења се рачуна као

$$t_1(q) = \frac{V_C}{Q_C - q}. \quad (7)$$

а вријеме пуњења запремене црпљења као:

$$t_2(q) = \frac{V_C}{q}. \quad (8)$$

гдје q представља рачунски доток воде у запремину црпљења који је дат у интервалу од 0 до 12 L/s .

Број укључења пумпи на сат зависи од броја расположивих пумпи. У конкретном случају на располагању може бити само једна пумпа (друга у квару) или обе које се наизмјенично укључују и искључују.

Према томе, максималан број укључења пумпи на сат је (једна пумпа на располагању):

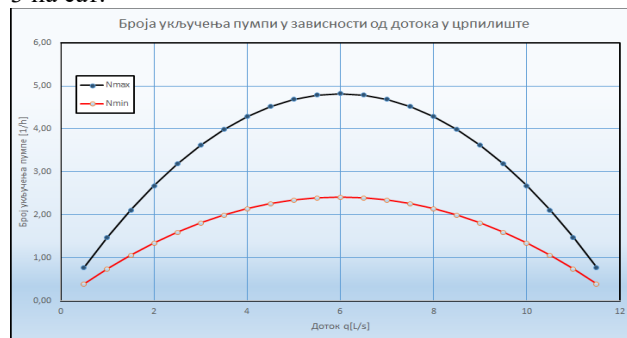
$$N_{Cmax}(q) = \frac{1}{T(q)}. \quad (9)$$

а минималан (две пумпе у цикличном раду):

$$N_{Cmin}(q) = \frac{1}{2T(q)}. \quad (10)$$

Резултати прорачуна броја укључења пумпи на сат у функцији дотока воде у црпилиште црпне станице дати су на слици 4. Као што се може закључити,

највећи број укључења пумпи на сат може се очекивати у случају да је доток воде у црпилиште једнак отприлике половини капацитета једне пумпе у погону. Тако када је на располагању само једна пумпа, број укључења на сат не прелази 5, а у случају да су на располагању обе пумпе број укључења је мањи од 3 на сат.



Слика 4. Број укључења пумпи на сат

4. ЗАКЉУЧАК

Предмет рада био је избор и прорачун машинске опреме црпне станице за заштиту од плављења енергетске галерије узводне главе бродске преводнице ХЕ Ђердап 1.

На бази дефинисаних нивоа воде у црпилишту црпне станице, коте изливног цјевовода и потребног капацитета црпне станице извршено је најпре димензионисање пумпних цјевовода а затим и хидраулички прорачун за избор пумпних агрегата. Изабране пумпе остварују нешто већи капацитет од потребног, што је на страни сигурности постројења.

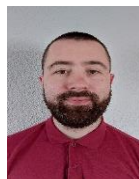
Коначно, за дефинисану геометрију црпилишта извршен је прорачун броја укључења пумпних агрегата на сат у циљу провјере опасности од прегријевања погонских мотора.

На основу извршених прорачуна може се закључити да изабрани пумпни агрегати задовољавају све техничке захтеве за поуздану заштиту енергетске галерије узводне главе бродске преводнице ХЕ Ђердап 1.

5. ЛИТЕРАТУРА

- [1] https://www.eps.rs/lat/djerdap/Stranice/he_djerdap1.aspx
- [2] Idelchik, I. E.: Handbook of Hydraulic Resistance, Sprigner-Verlag, Berlin, 1976.
- [3] Miller, D. S.: Internal Flow Systems, BHRA Fluid Engineering, 1986.
- [4] Каталог произвођача.
- [5] www.eps.rs/lat/djerdap/Stranice/brodske_prevodnice.aspx

Кратка биографија:



Петар Ивановић рођен је у Брчком 1997. Мастер рад на Факултету Техничких Наука из области Енергетике и процесна техника, модул - гасна и нафтна техника, одбранио је 2024. године. контакт: kinez.p.i@gmail.com

**DECENTRALIZOVANO USAGLAŠAVANJE SISTEMSKIH SATOVA U ČVOROVIMA
AD-HOK BEŽIČNE MREŽE****CONSENSUS CLOCK SYNCHRONIZATION IN DECENTRALIZED WIRELESS
NETWORKS**

Saša Taloši, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – Bežične ad-hok mreže danas imaju veoma široku primenu. Zbog prirode upotrebe, sinhronizacija vremena je jedan od bitnih problema sa kojim se suočavaju. U ovom radu ćemo prikazati jedan postojeći algoritam sinhronizacije vremena koji se bazira na usaglašavanju sistemskih satova čvorova.

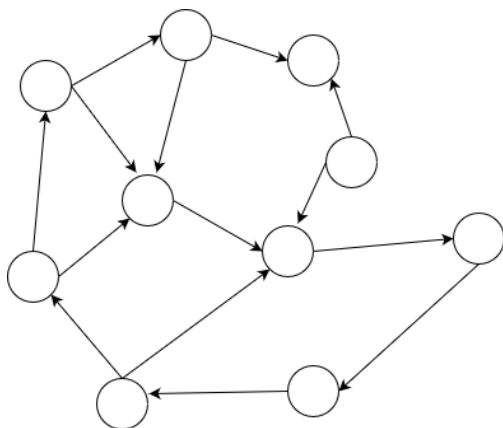
Ključne reči: Sinhronizacija satova, usaglašavanje satova, decentralizovane mreže, bežične mreže

Abstract – Decentralized low-powered wireless networks have a wide application in science and industry. Because of their distributed nature and differences in environment time synchronization is an important issue that needs to be addressed. In this paper, we will present one algorithm that solves this problem and is based on consensus clock synchronization.

Keywords: Clock synchronization, consensus synchronization, decentralized network, wireless network

1. UVOD

Kada je bežična mreža decentralizovana nazivamo je ad-hok bežičnom mrežom. U mreži ne postoje čvorovi koji služe za usmeravanje poruka već svaki čvor ravnopravno učestvuje u rutiranju podataka ka drugim čvorovima. Putanja podataka se određuje na dinamički način i zavisi od povezanosti same mreže i algoritma za rutiranje.



Slika 1. Primer bežične ad-hok mreže

Ovakve mreže predstavljaju dobro rešenje za prikupljanje podataka sa velikog broja uređaja. Neki od primera gde bi se ad-hok mreža mogla koristiti su strujna brojila, ulična rasveta, semafori, mreže senzora toplote, vlage, itd. Sem prikupljanja podataka, svaki čvor ima i sposobnost da obradi podatke i da upravlja aktuatorima.

Svaki čvor se sastoji od mikrokontrolera, oscilatora, mrežnog priključka, izvora struje, itd. Izbor mikrokontrolera i njihova primena raste iz godine u godinu, kao što propratno rastu i njihove sposobnosti. Da bi mikroprocesor radio, na čvoru se nalazi i oscilator (interni ili eksterni) koji određuje takt mikrokontrolera. Čvor, takođe, ima i način da komunicira sa mrežom. Komunikacija može biti putem bakra ili radio talasa. Kod bežičnih mreža nam je najbitnije na kojoj frekvenciji se vrši komunikacija. Neko pravilo kojim možemo da se vodimo je da što je veća frekvencija, to je prenos podataka brži, ali mreža postaje manje otporna na smetnje. Neki od primera poznatih tehnologija bežičnih mreža su: Wi-Fi [1], Bluetooth [2], Zigbee [3], LoRaWAN [4] i sada široko popularne bežične mreže tipa 5G i njeni prethodnici 1G, 2G, 3G i 4G.

Najčešće se koriste kristalni (kvarc) oscilatori zato što su jeftini, mali, mala im je potrošnja i relativno dobro drže frekvenciju u prisustvu smetnji. Međutim, temperatura i vlaga mogu promeniti takt oscilatora u dovoljnoj meri da uvedu grešku u sat mikrokontrolera.

1.1. Pregled problema

Vremenska sinhronizacija je veoma bitna za razne primene gde čvorovi moraju da rade istovremeno. Jedan takav primer je prikupljanje podataka. Merne stanice za struju, gas, vlažnost povremeno očitavaju i šalju podatke na jedno centralno mesto gde se obrađuju. Podaci mogu da se obrađuju i na samom čvoru u određenim primenama, ali je i dalje bitno kada je podatak očitao.

Drugi primer gde je bitna sinhronizacija vremena je pri tehnologijama koje se oslanjaju na usaglašenost (sinhronizam) kao ključan faktor. Na primer, slanje paket tokom precizno određenih vremenskih okvira kada su kanali slobodni.

Sinhronizacija čvorova se može iskoristiti i za uštedu energije. Čvorovi koji rade na baterije često se nalaze u uspavanom režimu rada i ako se ne bi probudili na vreme, propustili bi pakete koji se šalju ili bi kasno poslali svoj paket. Ovo bi bilo rešivo ako bi povećali vreme koje je čvor budan, ali bi to skratilo životni vek baterije.

NAPOMENA:

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

U ovom radu ćemo rešavati dva problema kod unutrašnjih satova u čvorovima. Prvi problem je inicijalna loša postavka sata koja uvodi smicanje u odnosu na realno vreme. Drugi problem je promena brzine oscilatora koja tokom vremena uvećava smicanje sata.

Greška smicanja je razlika u vremenu između vremena čvora i realnog vremena. Najčešće se pojavljuje pri samom postavljanju ili pri lošoj inicijalizaciji memorije čvora. Kada se čvor postavlja, nije moguće postići prikladnu preciznost i uključiti čvor u pravom momentu i ovde nastaje greška. U slučaju da inicijalizacija unutrašnjeg sata zavisi od predefinisane memorije, moguće je da je memorija loše postavljena i nastaje druga vrsta greške.

Kao što smo već naveli, temperatura i vlaga mogu kako ubrzati, tako i usporiti rad oscilatora. Ovo dovodi do toga da sat čvora napreduje ili kasni za realnim vremenom i evidentnije je na lokacijama na kojima su te promene veće.

Algoritam usaglašavanja [5], koji je primenjen u ovom radu, rešava oba prethodno predstavljena problema. Svi čvorovi se usaglašavaju ne na realno, već novo usaglašeno vreme. Postupak usaglašavanja takođe pokušava da koriguje i greške u brzini promene pojedinačnih oscilatora. Algoritam je potrebno periodično izvršavati kako bi se korigovale greške nastale od poslednjeg pokretanja, za šta je poželjno da svi čvorovi u svom dometu imaju što više čvorova koji sa njima komuniciraju.

1.2. Organizacija ostatka rada

U poglavlju 2. se prikazuju najznačajniji faktori od uticaja na sinhronizaciju vremena i navode se neki primeri postojećih algoritama. U poglavlju III je uveden matematički model sata, a u poglavlju IV je objašnjena matematička osnova algoritma. Poglavlje V daje opis organizacije simulacije i analizira rezultate simulacije. Nakon toga u poglavlju VI sledi zaključak i diskusija.

2. PREGLED LITERATURE

2.1. Faktori od uticaja na sinhronizaciju

Prilikom odabira algoritma za sinhronizaciju potrebno je uzeti u obzir nekoliko faktora [1]:

- Vreme pripreme - vreme koje je potrebno da pošiljalac pripremi paket za slanje.
- Vreme pristupa - vreme čekanja na slobodan kanal za slanje.
- Vreme slanja/primanja - vreme koje je potrebno da se poruka pošalje i primi i često je jednake dužine.
- Vreme prenosa - dovoljno malo da možemo da ga zanemarimo.
- Vreme obrade - vreme koje je potrebno da primalac obradi paket.
- Energetska efikasnost - bitan faktor ako je čvor na bateriju.
- Brojnost - bitna ako je algoritam namenjen za male ili velike mreže.
- Potrebna preciznost - preciznost koja se postiže algoritmom.
- Robusnost - opisuje koliko je neki algoritam otporan na to da se aktivni čvorovi mogu menjati kroz vreme.

- Životni vek sinhronizacije - da li sinhronizacija mora da se pokreće više puta ili je dovoljno samo jednom.
- Domet - bitan ako može sinhronizacija da se radi i lokalno, ne uvek na celoj mreži.
- Cena - da li je potrebno dodatnih hardverskih modula da bi algoritam radio.

2.2. Postojeći algoritmi

Globalni pozicioni sistemi (GPS) se mogu koristiti za sinhronizaciju sa realnim vremenom i pružaju visok stepen preciznosti, čak od 1 do 10 nanosekundi greške [6]. Ovakvi sistemi zahtevaju dodatan hardver koji će komunicirati sa satelitom.

Precision Time Protocol (PTP) je algoritam koji sinhronizuje vreme u računarskim mrežama. Na lokalnim mrežama on postiže i preciznost ispod 1 milisekunde [7]. Koristi se gde GPS nije dostupan, finansijskim sistemima i drugim vremenski kritičnim mrežama.

Network Time Protocol (NTP) je jedan od najstarijih algoritama za sinhronizaciju računarskih mreža. Dizajniran je da sinhronizuje sve povezane računare na preciznost od oko 1 milisekunde [8].

The Timing-sync Protocol for Sensor Networks (TPSN) [9] je algoritam koji je proširenje NTP algoritma i koristi se u bežičnim mrežama. Tokom postavljanja mreže čvorovi pronalaze jedni druge i uspostavljaju međusobne odnose nadređenog i podređenog. Ovako se stvara stablo gde je glavni čvor u korenu. Sinhronizacija kreće od korena i obavlja se niz grane dok svi čvorovi ne budu sinhronizovani.

The Reference Broadcast Synchronization [10] (RBS) je algoritam koji pruža sinhronizaciju samo čvorovima koji su u dometu glavnog čvora. On može ili ne mora da ima GPS prijemnik i povremeno pošalje paket sinhronizacije koji ostali čvorovi iskoriste da se sinhronizuju.

U algoritmu *Asynchronous Diffusion* [11] čvorovi potezno šalju sinhronizacioni paket svojim komšijskim čvorovima koji odgovaraju sa svojim vremenom. Prvi čvor usrednji vrednost svih dobijenih vremena i dobijenu vrednost pošalje svima koji su u dometu i svi prihvate poslatu vrednost kao novo vreme. Ovo se ponavlja u celoj mreži dok se ne postigne sinhronizacija.

Average TimeSync [12] (ATS) je mešavina dva algoritma usaglašavanja. U prvom koraku, čvorovi šalju svoje lokalno vreme u cilju da izračunaju grešku brzine u odnosu jedan sa drugim. Nakon toga, čvorovi šalju svoju pretpostavku greške brzine i ovo koriste da pretpostave grešku brzine usaglašenog sata. Isti koraci se ponove i za grešku smicanja.

3. MODEL SATA

Sat možemo da prikazemo kao funkciju u zavisnosti od realnog vremena

$$S(t) = \alpha t + \beta. \quad (1)$$

Prvi izvod sata $dS/dt = \alpha$ predstavlja brzinu sata i određuje koliko će se sat uvećati ili umanjiti za određeni period vremena, a β je smicanje sata u odnosu na realno vreme. Idealna brzina sata je 1, a idealno smicanje 0.

Koliko često treba raditi sinhronizaciju satova zavisi od promene brzine sata. Što je ta promena veća,

sinhronizaciju treba raditi češće i obrnuto. Da bi se postigla sinhronizacija u mreži, potrebno je da svi čvorovi $i = 1 \dots N$ savršeno kompenzuju parametre α_i i β_i , tako da je smicanje 0, i svi satovi kucaju istim taktom.

$$\widehat{S}_i(t) = \frac{\alpha_i}{\widehat{\alpha}_i} t + \beta_i - \widehat{\beta}_i = t, i = 1 \dots N \quad (2)$$

$\widehat{S}_i(t)$ je kompenzovani sat čvora i , a $\widehat{\alpha}_i$ i $\widehat{\beta}_i$ su kompenzacija brzine i kompenzacija smicanja čvora i . Da bi se zadovoljila jednačina (2), kompenzacioni parametri svakog čvora moraju biti jednaki njegovim parametrima sata, $\widehat{\alpha}_i = \alpha_i$ i $\widehat{\beta}_i = \beta_i$.

4. ALGORITAM USAGLAŠAVANJA

Umesto da pokušamo i sinhronizujemo vreme na realno vreme, ovaj algoritam pokušava da postigne usaglašeno vreme unutar mreže. Sa svakom sinhronizacijom, kompenzacioni parametri se ažuriraju i s vremenom se postiže usaglašavanje, tako da je

$$\lim_{t \rightarrow \infty} \widehat{S}_i(t) = S_u(t), \quad (3)$$

gde je

$$S_u(t) = \alpha_u t + \beta_u \quad (4)$$

usaglašeni sat. Proširivanjem (3) pomoću (2) i (4) dobijamo

$$\lim_{t \rightarrow \infty} \frac{\alpha_i}{\widehat{\alpha}_i} t + \beta_i - \widehat{\beta}_i = \alpha_u t + \beta_u, \quad (5)$$

odakle neposredno sledi

$$\lim_{t \rightarrow \infty} \widehat{\alpha}_i = \frac{\alpha_i}{\alpha_u}, \quad (6)$$

$$\lim_{t \rightarrow \infty} \widehat{\beta}_i = \beta_i - \beta_u. \quad (7)$$

Da bi se postigli kompenzacioni parametri čvorovi moraju da ponavljaju sinhronizaciju koja se sastoji iz dva dela, kompenzacija smicanja i kompenzacija brzine. U svakoj sinhronizaciji, čvorovi razmenjuju svoje lokalne satove da bi postigli jedno zajedničko vreme i poništili grešku smicanja, a u isto vreme upoređuju novo vreme sa vremenom iz prošle sinhronizacije i uklanjaju grešku brzine. Sinhronizacija se vrši u intervalu t_{sync} , koji je podesiv i može se menjati po potrebi.

4.1. Uklanjanje greške smicanja

U svakoj sinhronizaciji, čvorovi treba da izvrše sledeće korake:

1. Svaki čvor podesi svoj faktor samopouzdanja na
2. Svaki čvor pošalji paket sinhronizacije koji sadrži njegovo trenutno vreme i faktor samopouzdanja.
3. Svaki put kada prime sinhronizacioni paket izračunaju (8) i povećaju svoj faktor samopouzdanja (9).

$$\widehat{S}_i^+(t_j) = \frac{\gamma_i \widehat{S}_i(t_j) + \gamma_j \widehat{S}_j(t_j)}{\gamma_i + \gamma_j} \quad (8)$$

$$\gamma_i^+ = \gamma_i + 1, i = 1 \dots N, \forall j \in R_i \quad (9)$$

N je broj čvorova u mreži, a R_i su čvorovi koji su u dometu čvora i . γ_i i γ_j su faktori samopouzdanja čvorova i i j , a t_j je vreme kada je čvor j poslao svoj sinhronizacioni paket. Jednačina (9) takođe može da se izrazi kao (10) što je lakše mikrokontrolerima da izračunaju jer ima manje operacija množenja i deljenja.

$$\widehat{S}_i^+(t_j) = \widehat{S}_i(t_j) + \frac{\gamma_j}{\gamma_i + \gamma_j} (\widehat{S}_j(t_j) - \widehat{S}_i(t_j)) \quad (10)$$

Faktor samopouzdanja daje veći značaj čvorovima koji su primili sinhronizacione pakete jer su oni bliži usaglašenom vremenu.

4.1. Uklanjanje greške brzine

Nakon što se sinhronizaciona runda završila, čvor i ima bolju estimaciju usaglašenog vremena, $\widehat{S}_i^+(t_n + \tau)$. t_n je vreme početka sinhronizacije, a τ je trajanje sinhronizacije. Ako izračunamo parametar kompenzacije brzine, $\widehat{\alpha}_i^+$, kao rezultat ćemo sledeću sinhronizaciju započeti sa preciznijim satom.

$\widehat{\alpha}_i^+$ se može izračunati bez da čvor zna svoju grešku brzine. Vreme čvora u momentu t_n se može izraziti kao funkcija prethodnog vremena plus posmatrano vreme koje je prošlo (11).

$$\widehat{S}_i(t_n) = \widehat{S}_i(t_{n-1} + \tau) + \frac{\alpha_i}{\widehat{\alpha}_i} (t_n - t_{n-1} - \tau) \quad (11)$$

Isto važi i za vreme nakon sinhronizacije (12).

$$\begin{aligned} \widehat{S}_i(t_n + \tau) &= \widehat{S}_i(t_{n-1} + \tau) \\ &+ \frac{\alpha_i}{\widehat{\alpha}_i^+} (t_n + \tau - t_{n-1} - \tau) \end{aligned} \quad (12)$$

Apsolutno vreme koje je prošlo između dve sinhronizacije može da se izrazi kao (13).

$$t_n - t_{n-1} - \tau = \frac{\widehat{\alpha}_i}{\widehat{\alpha}_i} (\widehat{S}_i(t_n) - \widehat{S}_i(t_{n-1} + \tau)) \quad (13)$$

Ako ubacimo (13) u (12), dobijemo

$$\begin{aligned} \widehat{S}_i(t_n + \tau) &= \widehat{S}_i(t_{n-1} + \tau) \\ &+ \frac{\widehat{\alpha}_i}{\widehat{\alpha}_i^+} (\widehat{S}_i(t_n) - \widehat{S}_i(t_{n-1} + \tau) + \tau) \end{aligned} \quad (14)$$

pri čemu je

$$\widehat{\alpha}_i^+ = \frac{\widehat{\alpha}_i (\widehat{S}_i(t_n) - \widehat{S}_i(t_{n-1})) + \alpha_i \tau}{\widehat{S}_i(t_n + \tau) - \widehat{S}_i(t_{n-1} + \tau)}. \quad (15)$$

S obzirom da ne znamo α_i i τ nismo u mogućnosti da računamo $\widehat{\alpha}_i^+$ na način opisan jednačinom (15). Međutim, ako $\widehat{\alpha}_i^+$ računamo svaki put kada primimo paket, možemo zanemariti τ jer je ono prvenstveno suma vremena čekanja na slanje paketa, a vreme slanja se pretpostavlja da je zanemarljivo. Na taj način dobijamo

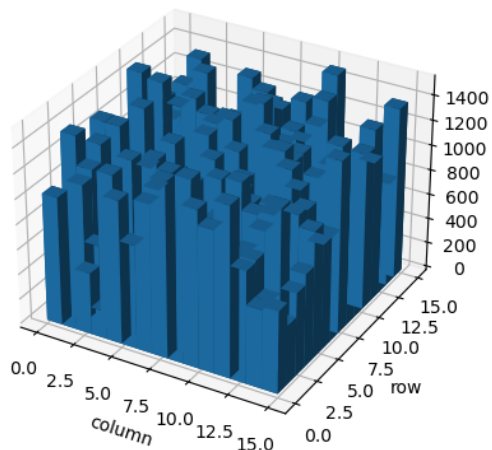
$$\widehat{\alpha}_i^+(t_{n,j}) = \widehat{\alpha}_i(t_{n,j}) \frac{\widehat{S}_i(t_{n,j}) - \widehat{S}_i(t_{n-1,j})}{\widehat{S}_i^+(t_{n,j}) - \widehat{S}_i(t_{n-1,j})}. \quad (16)$$

Vrednost $\widehat{S}_i^+(t_{n,j})$ predstavlja ažurirani sat u trenutku $t_{n,j}$ izračunato koristeći (10), dok je $\widehat{S}_i(t_{n-1,j})$ je vreme nakon poslednje sinhronizacije. Podsećamo da je $\widehat{\alpha}_i$ nije poznato pre prve sinhronizacije te u prvoj sinhronizaciji on iznosi $\widehat{\alpha}_i = 1$. radova.

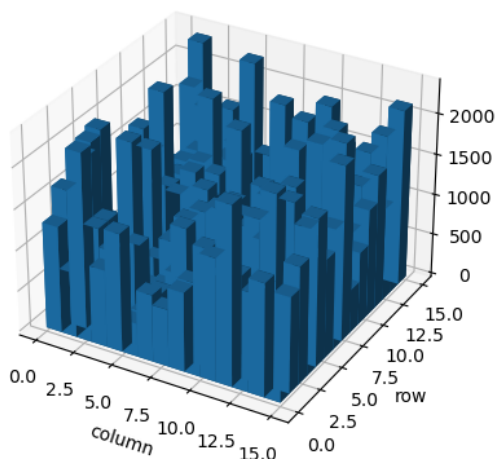
5. REZULTATI SIMULACIJE

Za simulaciju se koristila mreža od 255 čvorova, dimenzija 15 puta 15 i svaki čvor je imao domet 3. Ako je par čvorova van dometa, nisu u mogućnosti da se čuju. Parametri satova su nasumično postavljeni na vrednosti u rasponima $0.2 < \alpha < 1.2$ i $-1500 < \beta < 1500$, gde je β izraženo u milisekundama. Vreme u simulaciji je izraženo u milisekundama. Slika 2 je prikaz apsolutne greške svakog čvora. Nakon toga je u simulaciji vreme napredovalo 10 minuta. Slika 3 je prikaz apsolutne greške nakon 10 minuta simulacije, bez sinhronizacije. Slika 4 je

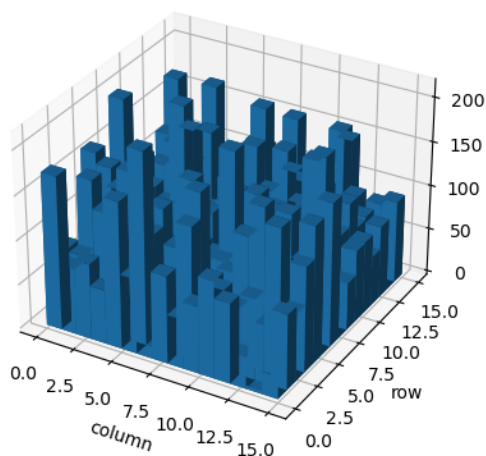
prikaz nakon sinhronizacije. Srednja apsolutna greška na početku iznosila je 749.6 milisekundi, nakon 10 minuta je iznosila 932.7 milisekundi i posle sinhronizacije iznosi 86.95 milisekundi. U toku simulacije primećeno je da vreme klizi naspram realnog vremena. Ovo bi moglo da se reši tako što bi se u mrežu dodao čvor sa realnim vremenom koji bi imao visok faktor samopouzdanja.



Slika 2. Apsolutne greške čvorova na početku simulacije



Slika 3. Apsolutne greške čvorova posle 10 minuta, bez sinhronizacije



Slika 4. Apsolutne greške čvorova posle 10 minuta, posle jedne sinhronizacije

6. ZAKLJUČAK

Prikazan je algoritam koji dosta zavisi od vremena pripreme, pristupa, slanja/primanja i obrade. Oslanja se na to da su sva vremena zbirno što manja da bi algoritam postigao bolju preciznost. Takođe je primećeno da preciznost raste sa porastom dometa između čvorova. Energetska efikasnost zavisi od toga koliko često se pokreće algoritam, što u velikoj meri zavisi od promenljivosti same mreže. Algoritam nije ograničen brojem čvorova i može da se modifikuje da se pokreće na lokalnom domenu. Izdrživ je i ne zavisi od promena čvorova. S obzirom da je u pitanju programsko rešenje, cena je mala jer ne zahteva nikakvu dodatnu elektroniku.

Pošto je u toku simulacije primećeno da usaglašeno vreme krene da se udaljava od realnog vremena, predlaže se da pojedini čvorovi budu prikazani sa GPS modulima koji bi davali precizno realno vreme i ovim čvorovima dati visok faktor samopouzdanja.

7. LITERATURA

- [1] IEEE 802.11TM WIRELESS LOCAL AREA NETWORKS, IEEE, [Online]. <https://www.ieee802.org/11/>
- [2] Bluetooth Specification, Bluetooth, [Online]. <https://www.bluetooth.com/specifications/specs>
- [3] 802.15.4 vs ZigBee, David Gascon, [Online]. <http://sensor-networks.org/index.php?page=08231231506>. 2012.
- [4] What Is LoRa?, Semtech, [Online]. <https://www.semtech.com/lora/what-is-lora>
- [5] Michael Kevin Maggs, Steven G O'Keefe, David Victor Thiel, Consensus Clock Synchronization for Wireless Sensor Networks, IEEE Sensors Journal, 6. 6. 2012.
- [6] GPS Time Transfer, National Institute of Standards and Technology, [Online]. <https://tf.nist.gov/time/commonviewgps.htm>
- [7] IEEE-1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, NIST, [Online]. <https://www.nist.gov/system/files/documents/el/isd/ieee/tutorial-basic.pdf>
- [8] David L. Mills, Computer Network Time Synchronization: The Network Time Protocol, CRC Press, 24. 3. 2006.
- [9] S. Ganeriwal, R. Kumar, and M. Srivastava, Timing-sync protocol for sensor networks, in Proc. 1st ACM Conf. Embed. Netw. Sensor Syst., Nov. 2003.
- [10] J. Elson, L. Girod, and D. Estrin, Finegrained network time synchronization using reference broadcasts, in Proc. 5th Symp. Operat. Syst. Design Implement., vol. 36. Dec. 2002.
- [11] Q. Li and D. Rus, Global clock synchronization in sensor networks, in Proc. IEEE Conf. Comput. Commun., vol. 1. Hong Kong, China, Mar. 2004.
- [12] L. Schenato and G. Gamba, A distributed consensus protocol for clock synchronization in wireless sensor network, in Proc. 46th IEEE Conf. Decis. Control, New Orleans, LA, Dec. 2007.

Kratka biografija:



Saša Taloši rođen je 1993. u Novom Sadu. Završio osnove akademske studije na smeru Računarstvo i automatika na katedri za Računarsku tehniku i računarske komunikacije.

KONCEPT NAPAJANJA SLOŽENIH TEHNOLOŠKIH CELINA NA PRIMERU GASNE KOMPRESORSKE STANICE – VELIKA PLANA**CONCEPT OF POWER SUPPLYING AN COMPLEX TECHNOLOGICAL PROCESS ON EXAMPLE OF A COMPRESSOR STATION – VELIKA PLANA**

Marija Čamparević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu razmatran je princip odabira opreme za napajanje objekta kompresorske stanice – Velika Plana uz kratak osvrt na princip rada same kompresorske stanice i opis trase gasovoda na kojoj se ona nalazi. Razmatra se odabir energetskih transformatora, kao i njihove zaštite. Obraden je i sistem brespekidnog napajanja koji je implementiran na samom objektu. Pored ovih sistema, određuje se i ATS uređaj koji služi kao sigurnost za napajanje potrošača u slučaju prestanka napajanja jednog od sistema sabirnica.

Ključne reči: Odabir energetskih transformatora, ATS, odabir zaštite transformatora, UPS.

Abstract – In this work, the principle of selecting equipment for the power supply of the compressor station facility - Velika Plana is discussed, with a brief review of the principle of operation of the compressor station itself and a description of the gas pipeline route on which it is located. The selection of energy transformers, as well as their protection, is considered. The uninterruptible power supply system, which was implemented on the building itself, was also processed. In addition to these systems, an ATS device is also specified, which serves as security for the supply of consumers in the event of a power failure of one of the bus systems.

Keywords: Selection of power transformers, ATS, selection of transformer protection, UPS.

1. UVOD

Konstantan rast populacije, tehnološki i ekonomski napredak društva iziskuje sve veću potrebu za energijom. Politikom cena sve više će se destimulisati korišćenje električne energije za zagrevanje, pa će se na taj način eliminisati kao energent. Strategija dugoročnog razvoja energetike Srbije u periodu do 2020. godine, sa vizijom do 2050. godine, ističe gas kao energetski izvor 21. veka i ekološki najprihvatljivije konvencionalno gorivo. Prirodni gas uz ugalj i ostala čvrsta goriva, jedini je primarni oblik energije koji se, uz elementarnu pripremu, može neposredno upotrebljavati.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dejan Jerkan, vanr.prof.

Neposrednom upotrebom prirodnog gasa može se u domaćinstvima zadovoljiti od 80-85% energetskih potreba (grejanje prostorija, kuvanje, priprema tople vode). Upotrebom gasa u pojedinačnim ložištima za grejanje, postiže se veća efikasnost nego upotrebom vrele vode iz toplana i kotlarnica. Prirodni gas ne sadrži zagađivače, pa se njegovim sagorevanjem dobija samo ugljen-dioksid i vodena para.

2. GASOVOD

Gasovod je dug 930 kilometara sa kapacitetom od 31,5 milijardi kubnih metara gasa godišnje. Prvi krak prolazi kroz Tursku, dok drugi krak ide ka Bugarskoj, Srbiji i Mađarskoj. Srbija i Bugarska taj deo gasovoda nazivaju "Balkanski tok". Kapacitet svakog kraka je 15,75 milijardi kubnih metara gasa. Ova količina energije dovoljna je da zadovolji potrebe 15 miliona srednjih domaćinstava. "Turskim tokom" transportuje se gas od ruske kompresorske stanice blizu Anape u ruskoj krasnodarskoj oblasti preko Crnog mora do prijemnog terminala u turskom mestu Kijikej. Od Kijikeja krak gasovoda nastavlja se ka distributivnom centru Luleburgas. Sledeći segment dužine 145 kilometra ide ka centru u Ipsali na tursko-grčkoj granici, koji se tu grana ka tursko-bugarskoj granici.

2.1. Kompresorska stanica

Kompresorska stanica je podeljena na više funkcionalnih celina (prostora):

Prostor za filtere, Prostor za kompresore, Prostor za vazdušne hladnjake, Prostor za turbinski pogonski gas, Prostor za pripremu tople vode, Prostor zatvorene drenaže, Prostor za ispuštanje gasa, Prostor za izdvajanje azota, Prostor za dizel, Prostor za servisnu vodu i vodu za piće, Prostor za prečišćavanje otpadnih voda...



Slika 2.2 Kompresorska stanica – Velika Plana

N_g : Prosečna godišnja gustina pražnjenja

$T_d = 33$: Prosečan broj dana sa grmljavinom na području Velike Plane i opštine Žabari.

7. SISTEM BESPREKIDNOG NAPAJANJA

Sistem za generisanje AC i DC sigurnosnog napona koncipiran je tako da su izvori AC i DC napona međusobno redundantni, a razvodi za distribuciju napona ka potrošačima su tako izvedeni da je obezbeđeno sigurno napajanje potrošača bez prekida. Sistem se sastoji iz dva poddela.

Prvi je podsistem za generisanje, distribuciju i skladištenje energije na jednosmernom naponu 220 V DC i sastoji se od:

- dva ispravljača 220 V DC, 250 A DC;
- dve baterije kapaciteta 800 Ah, 220 V DC, 108 ćelija;
- razvoda DC napona sa spojnim poljem i 20 izvoda za napajanje potrošača.

Drugi je podsistem za generisanje i distribuciju naizmeničnog napona i sastoji se od:

- dva invertora snage 60 kVA, 230 V AC;
- trokanalne statičke preklopke snage 60 kVA sa ručnim prekidačem za izbor mrežnog dovoda;
- razvoda AC napona.

7.1. Opis ispravljača i DC razvoda

7.1.1. Osnovne karakteristike

Ispravljač je namenjen za punjenje po određenom režimu i održavanje u napunjenom stanju akumulatorske baterije i samo u paralelnoj vezi sa akumulatorskom baterijom ispravljač predstavlja izvor jednosmernog napajanja koji je regulisan, dok postoji napajanje iz mreže, a održavan skokovito, u dozvoljenim granicama, u slučaju nestanka napona iz mreže.

7.1.2. Upravljačka elektronika

Osnovu upravljačke elektronike čine tri 16-bitna mikrokontrolera iz familije PIC kontrolera proizvođača MICROCHIP.

U ovom slučaju se radi o SISD (single instruction single data) arhitekturi, korišćenjem tri kontrolera omogućava se kako multitasking tako i redudansa na nivou hardvera. Pomenuta familija kontrolera izabrana je na osnovu dokazanih performansi i pouzdanosti u radu.

7.1.3. Režimi rada

Da bi napon na potrošačima bio u dozvoljenim granicama pri svim režimima punjenja i pražnjenja, baterija je podeljena u dve grane: osnovnu i dodatnu.

U normalnom radu pri pražnjenju akumulatorske baterije, potrošači su priključeni na napon osnovne grane, dok se pri dužem pražnjenju baterije, uključivanjem kontaktora automatski prebacuju na celokupnu bateriju.

7.1.4. Monitoring izlaznih veličina i zaštita od neispravnog funkcionisanja

Ispravljač je konstruisan tako da neprekidno vrši nadzor nad izlaznim veličinama i u slučaju problema u radu preduzima odgovarajuću akciju. Zaštita funkcioniše tako da u slučaju poremećaja dolazi do paljenja signala greške i/ili zaustavljanja rada.

7.1.5. Razvod jednosmernog napona

Razvod jednosmernog napona 220VDC u sebi ima dva dovodna prekidača sa ispravljača i prekidač spojnog polja (molded case circuit breakers MCCB) odgovarajuće prekidne moći i odgovarajuće nazivne struje. Kao izvodni zaštitni elementi se koriste dvopolni automatski prekidači (miniature circuit breakers MCB) odgovarajuće prekidne moći i odgovarajuće nazivne struje.

7.2. Opis statičke preklopke i AC razvoda

Statička preklopka realizovana je u vidu slobodnostojećeg ormana sa četiri ulaza (inverter 1, inverter 2, mreža 1 i mreža 2) i podržava rad u "on-line" ili "off-line" režimu. U "on-line" režimu vodeći izvor su invertori (rade u paraleli i dele opterećenje), dok je u "off-line" režimu vodeći izvor mreža.

7.2.1. Opis AC razvoda besprekidnog napajanja

Razvod besprekidnog napajanja se sastoji od potrebnog broja dvopolnih automatskih prekidača, odgovarajuće nazivne struje, prekidne moći i krive okidanja.

7.3. Opis invertora

Invertor je uređaj zasnovan na primeni komponenata energetske elektronike, koristi jednosmerni napon spoljašnje baterije 220 Vdc i pomoću savremene poluprovodničke tehnologije ga pretvara u monofazni nazimenični sinusni napon stabilne amplitude i frekvencije.

Invertor je upravljao pomoću najsavremenije mikrokontrolerske tehnologije (na bazi 32-bitnog ARM mikrokontrolera) sa primenom metode strujne kontrolne petlje koja obezbeđuje izvanredne dinamičke osobine invertora pri naglom potpunom opterećenju ili rasterećenju invertora, kao i bezbedan rad pri metalnom (tvrdom) kratkom spoju na izlazu uređaja.

7.4. Proračun izbora baterija

Minimalni napon baterije U_{minb} se računa prema minimalno dozvoljenom naponu sistema (187V) i maksimalno procenjenom padu napona u kablju do potrošača (2%):

$$U_{minb} = 187 + 4,4 = 191,4 \text{ V.}$$

Minimalni napon ćelije tada iznosi:

$$U_{minc} = \frac{191,4}{108} = 1,77 \text{ V.}$$

Zadatak ispravljača jeste da napaja potrošače u DC razvodu i istovremeno dopunjava bateriju. Nominalni DC napon 220V je u granicama 220V $\pm$ 10%, što znači da se napon na DC potrošačima mora kretati od 198V do 242V da bi se obezbedilo njihovo sigurno i bezbedno napajanje.

8. SN KABLOVI

Izbor kablova vrši se u skladu sa:

- Trajno dozvoljenom strujom pri različitim tipovima razvoda;
- Dozvoljenim padom napona;
- Dozvoljenim padom napona prilikom starta motora;
- Koordinacijom sa zaštitnim uređajima;
- Zaštitom u slučaju indirektnog dodira;
- Udarnom strujom kratkog spoja (samo za SN kablove).

9. ZAŠTITA

9.1. Izbor strujnog transformatora

Karakteristike strujnog transformatora treba odabrati tako da se, u zavisnosti od namene strujnog transformatora i uslova na mestu ugradnje, obezbedi zahtevani nivo tačnosti i sigurnosti u radu.

9.2. Izbor zaštitnih uređaja

U elektroenergetskim sistemima, IEEE numeracija uređaja identifikuje karakteristike zaštitnih uređaja kao što su releji ili prekidači.

- **IEEE broj 50** - Trenutni prekostrujni element
- **IEEE broj 51** - Vremenski-zavisani prekostrujni element u naizmeničnim strujnim kolima
- **IEEE broj 52** - Prekidač snage u naizmeničnim strujnim kolima

9.2.1. Kriterijumi za evaluaciju sistema relejne zaštite

Kriterijumi za evaluaciju sistema relejne zaštite 35 kV mreže napajanja postrojenja Kompresorske stanice Velika Plana, su definisani kroz sledeće četiri tačke:

- Minimiziranje trajanja kvara, tj. maksimalna moguća sigurna osetljivost trenutnih zaštitnih elemenata;
- Izolovanje mesta kvara od strane rasklopne opreme (jednog ili više prekidača snage) koja odgovara elementu koji je pod kvarom;
- U analizama, usvojen je minimalni koordinacioni interval od 300 ms za normalno stanje, odnosno, 200 ms za izuzetne situacije;
- Minimizacija ispada elektroenergetskih elemenata pri otkazu prekidača snage ili kvara odgovarajućeg zaštitnog uređaja.

Princip podešavanja :

Predlog je da se fazna prekostrujna zaštita realizuje sa tri stepena. Prvi stepen služi za zaštitu transformatora od preopterećenja, dok je uloga drugog i trećeg stepena da štite transformator od međufaznih kratkih spojeva.

10. ZAKLJUČAK

Tehnoloski i ekonomski razvoj čovečanstva doneo je mnoge pogodnosti i rešenja, ali je takođe i iznedrio mnoge probleme vezane za energetske resurse, koji moraju ispratiti ovaj razvoj. Uz svaki problem eksploatacije energije, javlja se i problem energetske efikasnosti tog rešenja, kao i zaštite životne sredine. Kroz ovaj rad je detaljno opisano i prikazano postrojenje za kompresiju gasa, kao i funkcionalnost istog. Objekat ovog tipa je jedinstven na teritoriji Srbije. Izgradnjom gasovoda, Srbija se približila evropskoj direktivi za povećanje energetske efikasnosti, na planu od 2020. do 2050. godine, primenom gasa kao izvora toplotne energije.

Rad je imao za cilj da upozna čitaoca sa funkcionisanjem ovakvog sistema, ali i da pokaže kako se vrši usklađivanje opreme spram potrošača, objekta i infrastrukture. Oprema mora omogućiti bezbedan rad u svim zahtevanim režimima, kao i bezbedno zaustavljanje samog sistema. Tehničko rešenje napajanja, u kom će se pogon nalaziti većinu svog vremena, i rešenje havarijskog stanja, značajno se razlikuju. Zadatak oba rešenja je isti, moraju se zadovoljiti potrebe tehnologije, kako bi se sam proces kompresije gasa nastavio.

6. LITERATURA

- [1] Srbijagas, zvaničan sajt. <https://www.srbijagas.com>
- [2] Jovan Nahman, *Razvodna postrojenja*, Elektrotehnički fakultet Beograd 2005.
- [3] *Technical description of the uninterrupted power supply*, Electrical engineering institute "Nikola Tesla", Belgrade, 2020.
- [4] Glavni projekat transformatorske stanice 35/0,4kV (u objektu kompresorske stanice), MAGISTRALNI GASOVOD (INTERKONEKTOR) GRANICA BUGARSKE – GRANICA MAĐARSKE – Deonica 5 – Kompresorska stanica i ostali objekti u funkciji kompresorske stanice.
- [5] Milenko Đurić, *Relejna zaštita*, Beograd, 2003.
- [6] IEEE Std C37.2-2008: *IEEE Standard for Electrical Power System Device Function Numbers, Acronyms and Contact Designations* - IEEE Power and Energy Society, Sponsored by the Substations Committee and the Power Systems Relaying Committee, 2008.

Kratka biografija:

Marija Čamparević rođena je u Kragujevcu 1996. god. Bachelor rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Energetika, elektronika i telekomunikacije odbranila je 2020.god.

RAZVOJ NPM PAKETA ZA EKSTRAKCIJU NAJFREKVENTNIJIH BOJA SA SLIKE DEVELOPMENT OF NPM PACKAGE FOR EXTRACTING MOST FREQUENT COLORS

Jelena Bojanić, Milan Vidaković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Tema ovog rada jeste razvoj *npm* (*node package manager*) paketa (biblioteke) za rad sa bojama i izdvajanje najfrekventnijih boja sa slike.

Ključne reči: *npm* paket, obrada slike, *TypeScript*, *JavaScript*, *HSL*, *RGB*

Abstract – *This paper deals with the development of an npm package for working with colors and extracting the most frequent colors from an image.*

Keywords: *npm* package, image processing, *TypeScript*, *JavaScript*, *HSL*, *RGB*

1. UVOD

Kako bi se boje tačno izdvojile sa slike potrebno je izvršiti detaljnu analizu svih piksela, i na adekvatan način obraditi prikupljene informacije. Korišćenje gotovih rešenja u tom slučaju izuzetno je korisno, jer razvoj takvih rešenja zahteva visok nivo domenskog znanja (autor takvog rešenja mora dobro poznavati boje, razlike između boja i njihovih nijansi). Ovaj rad se bavi kreiranjem *npm* (*node package manager*) paketa *color-catcher* koji će omogućiti obradu slike i ekstrakciju najfrekventnijih boja koje se na toj slici nalaze. Posmatrani paket pruža i dodatne funkcionalnosti za rad sa bojama, obezbeđujući mogućnost dobijanja palete boja koje su u harmoniji sa predstavljenom bojom, na osnovu korisničkog unosa. Harmonije boja koje paket podržava su komplementarna, monohromatska, triadska i tetradska harmonija.

2. PREGLED SLIČNIH SISTEMA

Pre detaljnijeg navođenja funkcionalnosti i implementacije opisanog sistema, biće izvršena analiza sličnih sistema. Sva posmatrana rešenja biće isključivo biblioteke koje su dostupne kao paketi u *npm* registru. Da bi se neko rešenje smatralo sličnim, potrebno je da ono vrši analizu slike i identifikaciju boja koje se na toj slici nalaze. Jedan od takvih sistema je *Extract Colors* [1]. Dato rešenje predstavlja *TypeScript* biblioteku otvorenog koda, koju je razvila *node.js* zajednica. Može se koristiti u klijentskom i serverskom delu veb aplikacije.

Pruža opciju podešavanja velikog broja parametara, a neki od njih su količina piksela, razlika između sličnih boja i razlika u zasićenosti. Ono što je izdvaja od drugih, jeste i

činjenica da se za svaku pronađenu boju na slici takođe dobija informacija o zastupljenosti date boje ili njoj sličnih boja.

Jedna od poznatijih biblioteka za obradu slika u *npm* zajednici jeste *Color Thief* [2]. Napisana u *JavaScript* programskom jeziku, a obradu boja sa slike vrši putem procesa kvantizacije boja. Kvantizacija boja je proces koji uključuje prevođenje velikog broja boja u manji skup boja. Zbog upotrebe ove tehnike, posmatrano rešenje sadrži određen stepen greške, koji u zavisnosti od upotrebe može biti od manjeg ili većeg značaja. Nudi funkciju za izdvajanje dominantne boje sa slike, kao i funkciju za izdvajanje palete boja sa slike. Funkcionalnost koja bi učinila ovo rešenje boljim bi svakako bila informacija o tome koliko je posmatrana boja zastupljena na slici.

3. OPIS KORIŠĆENIH TEHNOLOGIJA

Rešenje je napisano u *TypeScript* [3] programskom jeziku. *TypeScript* je nadskup *JavaScript*-a [4], što znači da je svaki važeći *JavaScript* kod takođe i važeći *TypeScript* kod. Paket je objavljen pomoću *npm* centralnog repozitorijuma za distribuciju *JavaScript* paketa. Koristi *Canvas* API [5] koji obezbeđuje manipulaciju i rad sa grafikom unutar veb pregledača. Za rad sa bojama, koristi *RGB* [6] i *HSL* [7] modele boja. Za detekciju razlika između boja, upotrebljena je *color-diff* [8] biblioteka koja računa razliku između dve boje upotrebom *CIEDE2000* algoritma.

3.1 TypeScript

TypeScript je programski jezik otvorenog koda, koji razvija i održava *Microsoft*. Predstavlja nadskup *JavaScript*-a i dodaje jeziku opciono statičko i dinamičko tipiziranje. Omogućava definisanje tipova podataka i pruža napredne funkcionalnosti kao što su interfejsi, generički tipovi i enumeracija. Olakšava održavanje i povećava bezbednost tokom razvoja. Integracijom sa popularnim alatima za razvoj, postaje snažan izbor za izgradnju skalabilnih i održivih aplikacija.

3.2 .NPM (Node Package Manager)

Node Package Manager je zvanični centralni repozitorijum *node* modula, kao i softver za upravljanje bibliotekama od kojih aplikacija zavisi. Koristi se prvenstveno za upravljanje zavisnostima u *node.js* projektima, ali se takođe koristi za upravljanje paketima u *JavaScript* projektnom okruženju. Sastoji se od konzole i baze podataka javnih i privatnih paketa, koji se još nazivaju i *npm* registri. *Npm* pruža i opciju kreiranja i objavljivanja vlastitih paketa odnosno biblioteka. Samo-

NAPOMENA:

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

objavljivanje paketa na npm registru omogućava širenje znanja, efikasnost u razvoju, brzu integraciju gotovih rešenja, održavanje standarda, što doprinosi dinamičnom i inkluzivnom okruženju za razvoj softvera.

3.3. Canvas API

Canvas API je deo HTML5 [9] standarda i predstavlja tehnologiju koja omogućava dinamičko crtanje i rad sa grafikom koristeći JavaScript. Između ostalog, može se koristiti za manipulaciju pikselima u realnom vremenu. Od posebnog značaja za posmatrano rešenje jeste canvas element i canvas kontekst.

Element canvas-a je stvarni DOM [10] čvor koji je ugrađen u HTML stranicu. Kontekst canvas-a je objekat sa svojstvima i metodama koje se mogu koristiti za prikazivanje grafike unutar canvas elementa. Kontekst može biti 2D ili 3D. Svaki element koji je tipa canvas, može imati samo jedan kontekst.

3.4. Color-diff

Color-diff je JavaScript biblioteka koja, upotrebom CIEDE2000 [11] algoritma, pruža funkcije za rad sa bojama i njihovo poređenje.

International Commission on Illumination (CIE) [11] je organizacija koja se bavi standardizacijom u oblastima kolorimetrije, fotometrije i drugih srodnih disciplina. Formirana je početkom 1900-ih godina kako bi razvila standarde i preporuke za date oblasti i kako bi se omogućila konzistentnost i međunarodno razumevanje u proučavanju svetlosti, boja i njihovih efekata. Odigrala je ključnu ulogu u oblasti poređenja boja, predstavljajući svetu koncept Delta E.

Delta E je mera promene u vizuelnom opažanju dve date boje. Ova metrika se koristi za kvalifikaciju razlike u boji na onaj način kako je percipira ljudsko oko. Izraz delta potiče iz matematike i označava promenu. Dodatak E odnosi se na nemačku reč *Empfindung*, što se prevodi kao senzacija.

Delta E vrednost se računa na osnovu razlika u parametrima boje kao što su nijanse, zasićenje i svetlina. Na tipičnoj skali, vrednost se kreće između 0 i 100:

- ≤ 1.0 ne vidi se ljudskim okom,
- 1-2 vidi se detaljnim posmatranjem,
- 2-10 vidi se na prvi pogled
- 11-49 boje su više slične nego suprotne i
- 50-100 boje su suprotne.

4. SPECIFIKACIJA

Zadatak obuhvata izradu npm paketa, čija glavna funkcionalnost predstavlja izdvajanje najfrekventnijih boja sa slike.

4.1. Funkcionalni zahtevi

Prilikom razvoja npm paketa, funkcionalni zahtevi obuhvataju definisanje jasnog API-ja i parametara funkcije, izradu dokumentacije, obradu grešaka, kao i osiguravanje konzistentnosti u radu paketa. Budući da postoje brojni slučajevi korišćenja, postoji nekoliko parametara koje je moguće definisati prilikom pozivanja funkcije za ekstrakciju boja sa slike. Korisnicima je

omogućeno da utiču na preciznost i brzinu procesa promenom *blurFactor* parametra, koji određuje koliko se susednih piksela "smešta" u rezultujući piksel.

Broj najfrekventnijih boja koje korisnik želi da dobije zavisi od specifičnosti upotrebe, stoga je važno definisati parametar koji omogućava preciziranje željenog broja boja. Jedan od važnih funkcionalnih zahteva jeste dobro definisan API, što se izražava u dobro definisanim ulaznim i izlaznim parametrima funkcije. Ulazni parametri su tipizirani, a biblioteka je implementirana na način da adekvatno upravlja greškama. Prilikom unosa neispravnih tipova podataka, korisnik dobija poruku sa opisom detektovane greške.

Izlazna vrednost funkcije predstavlja niz boja, koje su sortirane u opadajućem redosledu prema procentu zastupljenosti svake boje na slici. Svaki element u rezultujućem nizu sadrži vrednost boje u RGB i HEX formatu, kao i zastupljenost date boje u procentima.

Idealno je da ovakav vid rešenja obuhvati što veći spektar slučajeva upotrebe, a poželjna je i pismena dokumentacija, koja je jednako bitna kao i sama izrada i implementacija. Dokumentaciji posmatrane biblioteke moguće je pristupiti putem npm veb sajta, gde su jasno definisane sve njene funkcionalnosti, način instalacije, način korišćenja kao i tipovi ulaznih i izlaznih parametara.

4.2. Nefunkcionalni zahtevi

Sa obzirom na primarnu upotrebu ovakvih biblioteka u klijentskim aplikacijama, neophodno je postići minimalno vreme odziva i brzu instalaciju. Ovakvi aspekti su važni radi postizanja optimalne efikasnosti u radu biblioteke. Prosečno vreme odziva posmatranog rešenja je 100ms, što ujedno predstavlja i minimalno prihvatljivo vreme koje se očekuje na klijentskoj strani.

5. IMPLEMENTACIJA

Opisana biblioteka sastoji se od pet funkcija. Četiri funkcije obezbeđuju rad sa harmonijama boja dok se ključna funkcionalnost biblioteke bavi izdvajanjem najfrekventnijih boja obradom slike. Biblioteka je napisana u TypeScript programskom jeziku.

5.1 Ekstrakcija najfrekventnijih boja

Predstavlja ključnu funkcionalnost posmatrane biblioteke. Ulazni parametar je objekat koji sadrži polje *imageURL*, odnosno putanju do učitane slike koju treba obraditi, polje sa nazivom *n* odnosno dužinu rezultujućeg niza kao i *blurFactor* koji označava koliko će se susednih piksela sjediniti u jedan prilikom zamućenja ulazne slike. Struktura funkcije može se podeliti u tri celine:

- obrada ulazne slike,
- klasifikacija i
- analiza boja.

Obrada slike podrazumeva učitavanje ulazne slike, proporcionalno smanjenje njenih dimenzija i iscertavanje na canvas element. Canvas API omogućava pristup piksel podacima slike. Dužina niza koji predstavlja piksel podatke direktno zavisi od dimenzija slike. Slike visoke rezolucije mogu sadržati i milione piksel podataka koji ih

opisuju. Zbog toga je korak proporcionalnog smanjenja dimenzija slike od izuzetne važnosti, kako bi se umanjio ukupan broj piksela koji je potrebno obraditi bez gubitka informacija.

Iz konteksta *canvas* elementa pristupa se nizu koji predstavlja piksel podatke date slike, čime se započinje sledeća celina, odnosno klasifikacija boja. Klasifikacija boja je proces koji obuhvata iteraciju kroz piksele slike, identifikaciju boje na svakom pikselu i kategorizaciju te boje u odgovarajuću klasu na osnovu definisanih karakteristika.

Upotrebom HSL modela, boje mogu klasifikovati u grupe. Pogodan kriterijum klasifikacije, jeste nijansa odnosno ugao na kom se boja unutar kruga boja nalazi. Iz tog razloga, u posmatranom rešenju definisano je sedam klase, gde je svaka klasa opisana nazivom, početnom vrednošću ugla od kog boje te klase počinju kao i vrednošću ugla sa kojim se boje te klase završavaju.

Nakon što je klasa za datu boju poznata, postoje tri slučaja:

- boja je već evidentirana unutar klase,
- postoji dovoljno slična boja datoj boji koja je već evidentirana unutar klase i
- boja nije evidentirana unutar klase i ne postoji dovoljno slična boja unutar iste klase koja je evidentirana.

U situaciji gde je data boja već zabeležena unutar određene klase, povećava se broj njenih ponavljanja. Ukoliko data boja nije prethodno detektovana unutar klase, neophodno je proveriti da li postoji dovoljno slična prethodno detektovana boja unutar iste klase. Za poređenje dve boje koristi se funkcija *diff* iz biblioteke *color-diff*. Proces poređenja boja se oslanja na iteraciju kroz sve boje unutar jedne klase i potragu za razlikama u odnosu na trenutnu boju.

Granična vrednost razlike između boja u ovom radu iznosi 17. Ukoliko rezultat poređenja dve boje iznosi manje od 17, smatra se da je razlika između njih toliko mala da ljudsko oko to ne uočava.

Ukoliko nijedna prethodno evidentirana boja sa sličnošću manjom od 17 nije prisutna, zaključuje se da se radi o prvom pojavljivanju posmatrane boje i da ne postoji dovoljno slična boja, već zabeležena u sistemu. Data boja se evidentira unutar detektovane klase.

Finalni korak uključuje preuzimanje boja iz klasa gde postoji barem jedna detektovana boja na slici i njihovo sortiranje prema broju ponavljanja. Za svaku prepoznatu boju izdvajaju se informacije o njenoj RGB vrednosti, HEX kodu, broju ponavljanja i procentu koji zauzima na slici. Procenat učešća boje na slici se računa kao odnos broja ponavljanja boje i ukupnog broja piksela koji su bili obrađeni.

6. OBJAVLJIVANJE BIBLIOTEKE

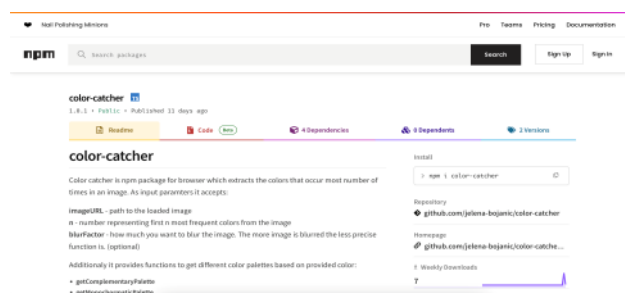
Npm pruža i opciju kreiranja i objavljivanja vlastitih paketa odnosno biblioteka. Samo-objavljivanje paketa na npm registru omogućava širenje znanja, efikasnost u razvoju, brzu integraciju gotovih rešenja, održavanje

standarda, što doprinosi dinamičnom i inkluzivnom okruženju za razvoj softvera.

Kako bi se paket kreirao potrebno je uspostaviti nalog na npmjs.com. Inicijalizacija paketa vrši se komandom *npm init*, nakon koje sledi nekoliko upita o imenu paketa, verziji paketa kao i ulaznom fajlu. Ime paketa mora biti jedinstveno i može sadržati samo mala slova.

Nakon implementacije rešenja, testiranje paketa moguće je vršiti upotrebom komande *npm link <naziv paketa>*. Ova komanda kreira *symbolic link*, odnosno vezu između lokalnog razvojnog paketa sa globalnom npm instalacijom.

Kada je rad na paketu završen, potrebno je prijaviti se na npm registar sa prethodno kreiranim nalogom putem komande *npm login*. Objavljivanje paketa realizuje se komandom *npm publish*. Svaki objavljen paket ima dodeljenu veb stranicu kao što je prikazano na slici 1. Svako naredno objavljivanje zahteva povećanje verzije paketa, a preporučuje se korišćenje semantičkog verzioniranja.



Slika 1. veb stranica objavljenog npm paketa

7. ZAKLJUČAK

Primarni zadatak ovog rada bio je razvoj npm paketa za rad sa bojama, fokusiranog na ekstrakciju najfrekventnijih boja sa slike. Osnovna ideja bila je stvaranje gotovog rešenja, usmerenog na rešavanje izazova gde je značajnija frekventnost boja nego njihova dominantnost.

Centralna funkcionalnost biblioteke jeste izdvajanje boja sa najvećim brojem ponavljanja na slici. Takođe pruža funkcije za generisanje harmonija boja na osnovu zadate boje.

Biblioteka je razvijena u trenutno savremenim tehnologijama koje se proširuju i napreduju, te ima dobru osnovu za dalji razvoj. Dodavanje opcije obrade video snimaka i primena biblioteke unutar serverskog dela aplikacije samo su neke od ideja koje bi ovu biblioteku unapredile i napravile ozbiljnu konkurenciju postojećim rešenjima.

8. LITERATURA

- [1] <https://www.npmjs.com/package/extract-colors> (pristupljeno u decembru 2023.)
- [2] <https://www.npmjs.com/package/color-thief> (pristupljeno u decembru 2023.)
- [3] <https://www.typescriptlang.org/> (pristupljeno u decembru 2023.)
- [4] <https://en.wikipedia.org/wiki/JavaScript> (pristupljeno u decembru 2023.)

- [5] https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API (pristupljeno u decembru 2023.)
- [6] https://en.wikipedia.org/wiki/RGB_color_model (pristupljeno u decembru 2023.)
- [7] https://en.wikipedia.org/wiki/HSL_and_HSV (pristupljeno u decembru 2023.)
- [8] <https://www.npmjs.com/package/color-diff> (pristupljeno u decembru 2023.)
- [9] <https://developer.mozilla.org/en-US/docs/Glossary/HTML5> (pristupljeno u decembru 2023.)
- [10] https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model (pristupljeno u decembru 2023.)
- [11] <http://cie.co.at/> CIE (pristupljeno u decembru 2023.)

Kratka biografija:



Jelena Bojanić rođena je u Novom Sadu 1997. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika, odbranila je 2023.god.
kontakt:
jelena.bojanic97@gmail.com



Milan Vidaković je završio doktorske studije 2003. godine na Fakultetu tehničkih nauka u Novom Sadu. Na istom fakultetu je 2014. godine izabran za redovnog profesora iz oblasti Primenjene računarske nauke i informatika.

**ИМПЛЕМЕНТАЦИЈА МЕХАНИЗМА АРБИТРАЖНОГ ТРГОВАЊА
КРИПТОВАЛУТАМА У iOS МОБИЛНОЈ АПЛИКАЦИЈИ****IMPLEMENTATION OF A CRYPTOCURRENCY ARBITRAGE MECHANISM IN AN
iOS MOBILE APPLICATION**

Лука Ступар, Факултет техничких наука, Нови Сад

Oblast – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У овом раду је изложено једно решење механизма за извођење арбитражне трговине криптовалута засновано на технологијама за развој iOS мобилних апликација и развијено помоћу окружења Xcode. Апликација служи за праћење цена и извршавање трговине на различитим мењачницама. Након описа апликације дискутовани су резултати тестирања и изведени одговарајући закључци.

Кључне речи: мобилне апликације, iOS, блокчејн, криптовалуте, трговање

Abstract – This paper describes an implementation of mechanism for arbitrage trading of cryptocurrencies as an iOS mobile application developed using Xcode IDE. The application is intended for cryptocurrency price tracking and automatic trade execution on different exchanges. After the application description, test results are discussed, and appropriate conclusions are drawn.

Keywords: mobile applications, iOS, blockchain, cryptocurrencies, trading, arbitrage

1. УВОД

Арбитражно трговање је стратегија која се примењује на великом броју финансијских тржишта са циљем остварења профита. Овај вид трговине заснива се на скоро истовременој куповини и продаји финансијских инструмената на различитим тржиштима како би се уновчила разлика у ценама. Чест исход оваквог начина трговања је, поред прихода, изједначавање цена робе или финансијског инструмента на различитим тржиштима. Арбитражно трговање настало је још у античким временима, а попримило данашњи облик почетком XXI века [1]. Применом блокчејна, знатно се повећао број тржишта на којима је могуће уочити прилике за арбитражну трговину. Иако се арбитража сматра ниско ризичним начином трговања, вероватноћа настанка финансијских губитака постоји услед клизања цена¹, трошкова размене, недовољне брзине, као и грешака.

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Душан Гајић, ванр. проф.

¹ Клизање цена означава разлику између очекиване и остварене цене при трговини.

2. ДИГИТАЛНА ИМОВИНА И БЛОКЧЕЈН

Дигитална имовина је сваки облик дигитално складиштене информације који има препознатљивост и који једном или више појединаца пружа вредност која, између осталог може бити материјална. Примери дигиталне имовине су: фотографије, документи, скупови података, метаподаци, криптовалуте, дигиталне валуте централних банака [2].

Блокчејн је технологија која служи за чување података намењена да отежа и онемогући фалсификовање ускладиштених података. Своју примену је пронашла у имплементацијама дистрибуираних регистара (енгл. DLT), што је омогућило напредне начине чувања дигиталне имовине.

Идеја блокчејна први пут настаје 1991. године, а 2008. године аутор под псеудонимом Сатоши Накамото објављује први предлог функционалне блокчејн мреже која решава проблем двоструке потрошње – Bitcoin, чија имплементација почиње са радом 2009 [3].

2.1 Мењачнице криптовалута

Крипто мењачнице су платформе намењене за куповину, продају и размену криптовалута. Многе мењачнице поред ових услуга нуде и могућност трговања финансијским дериватима, штедњу, позајмице и сличне услуге.

Најзначајнија подела ових платформи је на централизоване и децентрализоване. Док се централизоване мењачнице фокусирају на једноставност употребе, ниске трошкове конверзије криптовалута и корисничку подршку, децентрализоване мењачнице су технологија која користи блокчејн и паметне уговоре како би обезбедиле трговину без учешћа посредника, појачану безбедност, али уз ограничења по питању корисничке подршке [4].

2.2 Врсте арбитражног трговања криптовалута

Арбитража између мењачница – (енгл. Cross-exchange arbitrage) је метод трговања који захтева истовремено куповање и продају исте криптовалуте на две различите мењачнице.

Троугаона (или кружна) арбитража – (енгл. Triangular arbitrage) је стратегија трговања која користи разлике у ценама између три (у неким случајевима и више) криптовалуте.

Временска арбитража - (енгл. *Time arbitrage*) захтева праћење цене једне криптовалуте на једној мењачници како би се искористиле краткорочне промене у цени.

Арбитража унутар мењачнице – (енгл. *Inter-exchange arbitrage*) је стратегија која користи разлике у ценама различитих криптовалута на истој мењачници. Захтева идентификацију парова криптовалута који имају корелацију у цени и брзо извршавање трговина када дође до одступања у цени у односу на корелацију [5].

3. МОБИЛНЕ АПЛИКАЦИЈЕ

Мобилне апликације су наменски програми дизајнирани за рад са преносивим уређајима као што су паметни телефони. Појединачно омогућавају ограничен скуп функционалности које корисник може да комбинује и прилагоди својим потребама. [6]

3.1. Развој и испитивање мобилних апликација

Развој наменских апликација је поступак израде програмске подршке за преносиве уређаје узимајући у обзир могућности, ограничења физичке архитектуре и системског софтвера за дату врсту уређаја као што су: верзија оперативног система, количина меморије, процесорска моћ, величина екрана, итд. Процес развоја апликација подразумева дефинисање захтева, употребе и алата неопходних за њену имплементацију, дизајн корисничког интерфејса, дефиницију архитектуре и модела комуникације.

3.2 iOS и iOS мобилне апликације

iOS је оперативни систем намењен за преносиве уређаје који је развила компанија *Apple* и потекao је од *Mac OS X*. iOS подржава ограничен број уређаја у односу на друге мобилне оперативне системе, што му даје бољу прилагођеност физичкој архитектури, безбедност, као и само корисничко искуство [7].

iOS апликација је појам који обухвата широку групу програма израђених искључиво за уређаје које производи компанија *Apple* (*iPod*, *iPhone*, *iPad*). У већини случајева iOS апликације су затвореног кода. Одликује их једноставност употребе Пишу се у *Swift* и *Objective-C* програмским језицима. Велики број апликација је написан уз помоћ *UIKit* радног оквира који омогућава употребу елемената корисничког интерфејса и препознавање корисничких команди, али све је више популаран *SwiftUI* радни оквир.

4. SWIFT ПРОГРАМСКИ ЈЕЗИК

Swift је програмски језик отвореног кода и опште намене објављен од стране компаније *Apple* 2014. године. Главне одлике овог програмског језика су брзина извршавања, висок ниво апстракције као и дескриптивна синтакса приликом писања програма. Поред тога, сигурност типа података (енгл. *Type safety*) чини овај програмски језик погодним за писање мобилних апликација.

Данас се овај програмски језик најчешће користи за израду апликација на *iOS*, *macOS* и осталим платформама компаније *Apple*, али и за развој сервера у рачунарским мрежама [8].

4.1 SwiftUI

SwiftUI је радни оквир намењен за израду корисничког интерфејса који омогућава развој употребом декларативне синтаксе и напредних алата за дизајн. *SwiftUI* је направљен тако да функционише упоредо са старијим радним оквирима као што су *UIKit* и *AppKit*, што додатно олакшава његову употребу како у новим, тако и у постојећим пројектима [9].

4.2. CoreData

Core Data је граф објеката и радни оквир намењен за обраду и стално складиштење података на *macOS* и *iOS* платформама. Користи се за привремено и трајно чување података на, синхронизацију података између више уређаја и сл. Као базу података користи *SQLite*, а имплементација овог радног оквира омогућава једноставну манипулацију објектима без директним управљањем базама података [10].

5. ДИЗАЈН И РАЗВОЈ КОРИСНИЧКОГ ИНТЕРФЕЈСА

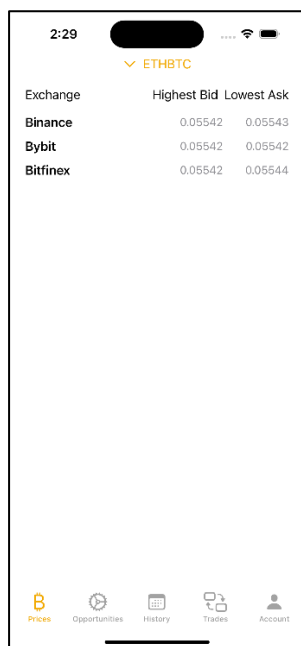
Кориснички интерфејс апликације је подељен на пет главних целина које омогућавају кориснику ефикасан преглед и конфигурацију апликације. Прва целина је приказ цена, односно екран који има задатак да прикаже цену изабраног пара криптовалута на свим подржаним тржиштима. У другој целини корисник има могућност да конфигурише понашање апликације, док трећа и четврта целина представљају прегледе историје цена, односно извршених трговина током арбитражног трговања. Четврта целина садржи додатна подешавања апликације.

5.1. MVVM шаблон

Model-View-ViewModel је структурни шаблон који се користи за раздвајање програмске логики од корисничког интерфејса. *MVVM* је користан приликом организације кода и рашчлањивања програма на модуле како би се поједноставила и убрзала надоградња и поновна употреба постојећег кода. Често се користи приликом израде мобилних и web апликација. На слици 1 може се видети изглед једне компоненте корисничког интерфејса [11].

Програмски код је организован у три компоненте:

- Приказ – (енгл. *View*) је скуп елемената корисничког интерфејса задужен за приказивање и прикупљање корисничких команди.
- Модел – садржи програмску логику и приступа му модел приказа приликом пријема корисничких команди.
- Модел Приказа – компонента која се налази између модела и приказа која садржи контроле за интеракцију са приказом и повезује елементе приказа са подацима који се налазе у моделу.



Слика 1- Приказ Дела Корисничког Интерфејса

6. ОРГАНИЗАЦИЈА ПРОГРАМСКОГ КОДА

Програмски код је подељен на сегменте који омогућавају његову прегледност и проширивост. Посебно су издвојене компоненте које обухватају структуре података неопходне за анализу цена, парова криптовалута, типова понуда итд.

6.1 Singleton образац

Singleton је структурни шаблон који се користи за креирање глобално јединствене инстанце класе или структуре података. Користан је када је потребно обезбедити синхронизован приступ и обраду података током конкурентног извршавања програма [12].

Компоненте које користе овај образац су:

- *KeychainManager* – Класа задужена за безбедно чување, добављање и брисање кључева дигиталних новчаника као и кључева за аутентификацију приликом комуникације са мењачницама.
- *LoginManager* – Класа која омогућава регистрацију, пријаву и одјаву корисничког налога
- *SettingsManager* – Структура података чија јединствена инстанца служи за управљање подешавањима.
- *DatabaseManager* – Ова класа служи за операције над базом података како би се обезбедила функционалност у деловима апликације који захтевају трајно складиште. Ослања се на *CoreData*

6.2. Решење

Решење предложено у овом раду представља апликацију израђену помоћу технологија наведених у претходном поглављу. Имплементација рада дата је у програмском језику Swift. За имплементацију коришћено је развојно окружење *Xcode*, а за само испитивање решења коришћен је *iOS* симулатор.

Тренутни захтев за правилан рад апликације је уређај који подржава *iOS* верзију 16.4 или вишу, или уређај који подржава *macOS Catalist* технологију.

Поред приступачног корисничког интерфејса који пружа, решење преузима понуде криптовалута на подржаним мењачницама, затим анализира куповну и продајну цену проналазећи прилике за трговину.

Тренутна имплементација подржава арбитражу између мењачница и кружну арбитражу на једној мењачници.

По проналаску на довољно велику разлику у ценама, апликација обавештава корисника путем корисничких обавештења *iOS* оперативног система, чува евиденцију о догађају у бази података, и уколико је то дозвољено, шаље захтев за аутоматску објављивање куповине и продаје одабраним мењачницама.

Тренутна изведба апликације подржава следеће мењачнице:

- *Binance*
- *Bybit*
- *Bitfinex*
- Децентрализоване мењачнице на *Hive* блокчејну

Испитивање апликације своди се на аутоматско тестирање појединачних модула програмског кода, и на ручно тестирање корисничког интерфејса на симулатору више различитих уређаја.

Такође, велики изазов за овакву врсту мобилне апликације јесте потреба за константном интернет везом, што није увек могуће због варијација у мрежном сигналу мобилних уређаја.

Поред тога, учестане промене цена и ниска учестаност појављивања прилика за арбитражно трговање са високом стопом профита учинили су време испитивања рада апликације дужим. Понашање апликације је посматрано током различитих стања мреже и самог оперативног система како би се обезбедила њена стабилност и робусност.

Додатни изазов је представљала чињеница да појединци и организације које се баве развојем алата за арбитражно трговање неретко одлучују да резултате свог рада не деле јавно како би задржали предност на тржишту, што као последицу има мањак истраживања и јавних имплементација решења, нарочито у области мобилних апликација.

7. ЗАКЉУЧАК

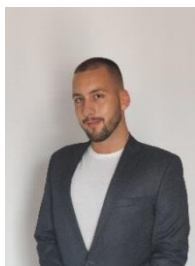
Почетна верзија апликације за арбитражно трговање пружа могућност једноставног аутоматизованог трговања на ограниченом броју тржишта.

Иако услови које арбитражна трговина криптовалутама често захтева испуњавање услова које није увек могуће задовољити на мобилним уређајима, овај рад показује да је могуће извршити аутоматизацију детекције и извршавања трговине такве да се током рада апликације оствари приход.

8. ЛИТЕРАТУРА

- [1] Geoffrey Poitras: *Arbitrage: Historical Perspectives*, Simon Fraser University, 2010
- [2] *Digital Asset: Meaning, Types, and Importance* – investopedia, приступано 19.11.2023. Доступно на: <https://www.investopedia.com/terms/d/digital-asset-framework.asp>
- [3] Сатоши Накамото, „*Bitcoin: A peer-to-peer electronic cash system*“, 2008, Приступано 21.11.2023. Доступно на: <https://bitcoin.org/bitcoin.pdf>
- [4] crypto exchange – PC mag, приступано 20.11.2023. Доступно на: <https://www.pcmag.com/encyclopedia/term/crypto-exchange>
- [5] Crypto Arbitrage Trading - Coindesk, приступано 20.11.2023. Доступно на: <https://www.coindesk.com/learn/crypto-arbitrage-trading-what-is-it-and-how-does-it-work/>
- [6] Лука Ступар, Дипломски рад, Имплементација механизма за логовање на ELK сервер у мобилним апликацијама – iOS, Факултет техничких наука, Универзитет у Новом Саду, 2021.
- [7] Процес развоја мобилних апликација, приступано 19.11.2023. Доступно на: <https://www.invonto.com/insights/mobile-app-development-process/>
- [8] Swift, приступано 21.11.2023. Доступно на: <https://www.swift.org/about/>
- [9] SwiftUI – Apple developer, приступано 21.11.2023. Доступно на: <https://developer.apple.com/xcode/swiftui/>
- [10] CoreData – Apple developer, приступано 21.11.2023. Доступно на: <https://developer.apple.com/documentation/coredata/>
- [11] Model-View-ViewModel (MVVM) – Tectarget, приступано 21.11.2023. Доступно на: <https://www.techtarget.com/whatis/definition/Model-View-ViewModel>
- [12] Singleton Design Pattern, Приступано 21.11.2023. Доступно на: <https://www.geeksforgeeks.org/singleton-design-pattern/>

Кратка биографија:



Лука Ступар рођен је у Новом Саду 1998. год. Дипломски рад на Факултету техничких наука из области Електротехника и рачунарство – Рачунарство и аутоматика одбрао је 2021. године.
контакт: stuparbluka@protonmail.com

**ТРОСЛОЈНА АРХИТЕКТУРА КАО СОФТВЕРСКО РЕШЕЊЕ СИСТЕМА ЗА
ЕВИДЕНЦИЈУ ТРЕНИНГА****THREE-TIER ARCHITECTURE AS A SOFTWARE SOLUTION FOR TRAINING
TRACKING SYSTEMS**Јована Матковић, *Факултет техничких наука, Нови Сад***Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО**

Кратак садржај – У овом раду описано је развијање целокупног система за евиденцију тренинга у спортским центрима и теретанама. Као софтверско решење имплементирани су веб и апликација за мобилне уређаје за тренере где се могу клијентима креирати налози и састављати планови тренинга. За ово решење коришћен је JavaScript програмски језик. Као хардверско решење имплементиран је мали ембедед уређај за клијенте који се може користити током тренинга за праћење вежби и здравствених параметара.

Кључне речи: *Микроконтролер, веб апликација, Android апликација*

Abstract – This project describes the development of the tracking training system for sports centers and gyms. Software solutions include a web platform and a mobile application for trainers, enabling them to create client accounts and design training plans. JavaScript programming language was used for this software solution. As a hardware solution, a small embedded device for clients was implemented, enabling them to track exercises and health parameters during training sessions.

Keywords: *microcontroller, web application, Android application*

1. УВОД

Веб апликације у последње време стичу велику популарност развојем Интернет технологије и веома се често интегришу у разне пројекте због своје приступачности и једноставности. Све што је кориснику у улози клијента потребно за коришћење овог типа апликација јесте веб прегледач. На овај начин се избегава ажурирање и одржавање апликације без потребе за инсталацијом и дистрибуцијом софтвера на различите рачунаре тих корисника што значајно олакшава коришћење без потребе за компликованим процесима инсталације софтвера на њиховим уређајима. Успостављање веб апликација донело је брзину развоја и лако ажурирање у системе који захтевају мало времена за имплементацију измена и исправки.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Предраг Теодоровић, ванр. проф.

Важно је поменути и да, за разлику од старијих метода које се ослањају на инсталацију софтвера на уређајима, код веб апликација за ажурирање на целокупном скупу уређаја је довољно извршити измене само на серверу.

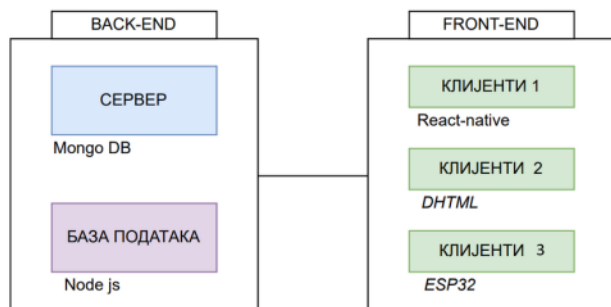
Са друге стране у свакодневној употреби данашњег човека су и ембедед системи чија је имплементација на доста нижем нивоу апстракције због употребе микроконтролера и њихове могућности су специјализоване за одређену примену што доводи до веће ефикасности приликом извршавања жељених функција. Међутим такав дизајн је за крајњег корисника доста ограничавајући јер не оставља превише могућности за персонализацију и интеракцију уопште. Из тог разлога све више се развијају технологије попут IoT (*Internet of Things*) које омогућавају повезивање ових уређаја са веб апликацијама за постизање једноставније контроле и надзора специјализованих ембедед уређаја путем Интернета што крајњем кориснику омогућава већу персонализацију и интеракцију [1].

Идеја овога пројекта јесте повезати поменуте две технологије у једну целину која налази примену у спортским центрима и теретанама где чланови добијају уређаје за евиденцију и праћење тренинга које им тренери задају преко мобилне веб апликације.

2. КОНЦЕПТ РЕШЕЊА

За израду веб апликације користи се трослојна архитектура која се састоји из три слоја - презентационог (кориснички интерфејс), слоја пословне логике и слоја података (база за складиштење података). Овакав систем чија је шема приказана на слици 1. је реализован у два блока од којих један представља серверску (енг. *Back-end*), а други клијентску (енг. *Front-end*) страну. Серверска страна је реализована помоћу Node.js платформе где је имплементирана апликација коришћењем Javascript програмског језика и Express.js back-end web application framework-а за node.js платформу [2]. Што се тиче клијентске стране, ту постоје два типа клијената, од којих се први тип односи на мобилну апликацију реализовану уз помоћ React Native framework-а коришћењем Javascript програмског језика, док други тип клијента подразумева ембедед уређај базиран на ESP32 модулу који је испрограмиран коришћењем C програмског језика. Оба клијента се повезују са сервером како би послали или прикупили релевантне податке о тренингу.

Сервер на почетку свог рада успоставља конекцију са *MongoDB Atlas cloud* базом података. Током даљег рада сервер чита и уписује податке у ту базу на захтеве клијената. Предвиђено је да сервер и клијенти буду повезани на исту локалну мрежу како би се комуникација са сервером успоставила помоћу његове локалне *ip* адресе, мада постоји могућност и да се *web* страница *host*-ује па да се онда она користи као приступна тачка.



Слика 1. Блок шема целокупног система

2.1. Серверска апликација

Серверска апликација се састоји од базе података, модела, рута и контролера и замишљено је да се покреће на рачунару спортског центра који жели да омогући коришћење овог система својим корисницима. Сврха ове апликације јесте извршавање логике целог система и манипулација са подацима које она прима од клијената или их шаље клијентима.

Клијентска апликација за мобилне уређаје користи *HTTP* комуникациони протокол за повезивање са сервером, док се ембедед апликација са сервером повезује преко тзв. *Websocket* протокола [3].

Услов за коришћење серверске апликације јесте да рачунар буде повезан на локалну мрежу и да има приступ интернету. Када је реч о *HTTP* захтевима, клијентски захтеви се даље обрађују помоћу рута креираних за одређену крајњу тачку како би се лакше разазнале врсте захтева и упита. За обраду овог типа захтева користи се *Express.js* који представља слој изграђен поврх *Node.js* и помаже у управљању сервиса и рута. Са друге стране, клијенти који се повезују преко *Websocket* протокола након конекције поруке размењују преко једног те истог канала, тако да нема нових захтева након једном успостављене конекције и за то се користи посебан *ws* пакет. Током комуникације са сервером клијенти могу да шаљу или примају податке. Ови подаци се складиште у *MongoDB Atlas cloud* бази података, а да би се сервер повезао са њом, потребан му је приступ Интернету.

За приступ бази и извршавање упита *Node.js* поседује пакет *Mongoose* са једноставним интерфејсом за објектно моделирање података (*Object Data Modeling - ODM*) [4]. Треба имати у виду да *MongoDB* спада у нерелациону врсту базе података (*NoSQL*) што значи да се подаци не морају складиштити по утврђеним шемама. Главна предност у односу на релационе базе података јесте могућност складиштења података у *JSON* формату где структура може да варира.

JSON представља отворени стандардни формат датотеке и формат за размену података који користи обичан текст за складиштење и пренос објеката који

се састоје од парова атрибут-вредност и низова или других вредности које се могу серијализовати.

Типови података за складиштење у базу дефинишу се у моделима који представљају шеме са атрибутима које један модел поседује. Преко поменутих модела даље контролери шаљу захтеве ка бази или креирају упите са жељеним параметрима претраге.

Разлог зашто је за реализацију серверске апликације коришћена баш платформа *Node.js* јесте могућност коришћења *Javascript* програмског језика за програмирање који је раније био намењен искључиво клијентској страни с обзиром на своју употребу ограничену само на прегледач. На овај начин постиже се једноставност имплементације јер је довољно познавање само *Javascript* програмског језика и за креирање клијентске и за креирање серверске апликације и сви новији веб прегледачи га подржавају.

Треба поменути да је *Javascript* скриптни језик који садржи *API* са рад са текстом, низовима, датумима и регуларним изразима, али не и улазно/излазне функционалности, као што су повезивање, складиштење података или графичке функционалности, за шта се ослања на окружење у коме се извршава.

2.2. Клијентске апликације

У пројекту постоји три типа клијената:

1. Веб апликација
2. Мобилна апликација
3. Ембедед апликација

Клијентска мобилна апликација је реализована помоћу *React Native* који представља *open-source framework* за графички дизајн. Овај *framework* покреће позадински процес који интерпретира *Javascript* директно на уређају крајњег корисника и комуницира са *native* платформом путем серијализованих података коришћењем асинхронних мостова. Користи се за развој *cross-platform* апликација и не захтева познавање *HTML*-а и *CSS*-а. Мобилна апликација намењена је тренерима и самосталним вежбачима који састављају планове тренинга. Корисници ове апликације имају могућност да направе налог који ће се користити на ембедед уређају и да за њега убележе све потребне информације о распореду вежби и броју понављања и серија на једном тренингу. Корисник може да креира, ажурира или обрише тренинг тако што се преко *HTTP* протокола шаље захтев серверу са информацијама које је потребно модификовати у бази. Овај протокол је изворно дизајниран за преузимање *HTML* страница у клијент-сервер окружењу, широко се користи за веб и одликује га то да је сваки захтев од клијента до сервера независан. Клијент је увек тај који шаље захтев, а сервер одговара на тај захтев и размена података се врши у читљивом текстуалном формату.

Свака трансакција је независна, што значи да се веза затвара након што је одговор послат. Захтева много заглавља и метаподатака у својим порукама и обично се користи за претраживање интернета помоћу *REST API* који укључује извршавање *GET*, *POST*, *PUT* и *DELETE* захтева. Треба поменути да је за овај тип клијената потребно реализовати и веб апликацију која

имплементира идентичну функционалност како би корисници могли и преко веб прегледача да приступе истим ресурсима у случају да не желе да користе *Android* или *iOS* уређај.

За креирање веб страница коришћен је *HTML* (*HyperText Markup Language*) који представља описни језик намењен креирању веб садржаја. *HTML* користи тагове да означи различите елементе на веб страници попут наслова, параграфа, слика итд.

Поменути тагови имају структуру која описује како садржај треба да се прикаже и организује на страници тако да се може рећи да *HTML* представља складиште информација које прегледачи читају и преводе у видљиве веб странице за кориснике. Како *HTML* има веома ограничене могућности у погледу дефиниције изгледа, структура и садржај се формирају помоћу *CSS* дескриптивног језика.

У комбинацији са Javascript-ом *HTML* и *CSS* чине *DHTML* (*Dynamic HTML*).

Клијентска апликација која се покреће на ембедед уређају од сервера тражи податке о тренингу како би их исписала кориснику на дисплеј. То омогућава кориснику да провери коју вежбу треба да ради на који начин и колико понављања. Дисплеј који се користи за интеракцију са корисником садржи резистивни панел осетљив на додир. На почетку корићења апликације, потребно је да се вежбач улогује на свој налог како би могао видети свој план тренинга.

Након сваког сета могуће је означити да је вежба одрађена и након тога тајмер одбројава паузу пре него што на дисплеју прикаже наредну серију/вежбу. Поред тога овај ембедед уређај поседује *MAX30102* сензор откуцаја срца и кисеоника у крви, које је по потреби могуће мерити и у реалном времену послати на *cloud*. Као што је већ поменуто ова врста клијента се са сервером повезује преко *Websocket*.

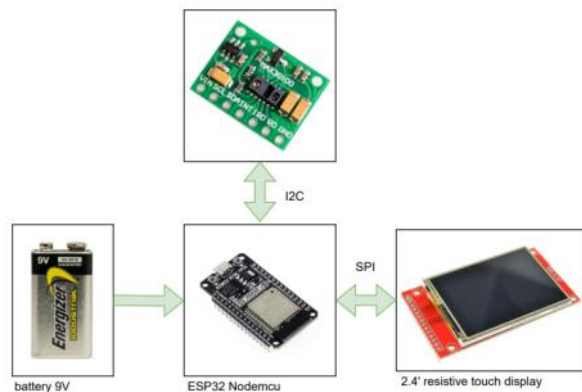
Под појмом *Websocket* се подразумева клијент-сервер комуникациони протокол који се извршава преко *TCP* (*Transmission Control Protocol*). Главна разлика између *WebSocket* и *HTTP* протокола јесте да је *HTTP* конекцијски оријентисан, док *WebSocket* обезбеђује комуникацију преко једне дугометражне *TCP* везе.

Овај протокол, иако мало захтевнији од *HTTP* протокола, омогућава бидирекциону *full-duplex* размену порука између клијента и сервера. Веома је поуздан и ефикаснији је од *HTTP* протокола јер се отвара једна веза која се одржава током целог трајања једне сесије. Погодан је за апликације које захтевају стална ажурирања и размену порука у реалном времену.

3. ХАРДВЕРСКИ ДЕО СИСТЕМА

На слици 2 приказана је блок шема хардверског уређаја са свим деловима који га чине. Хардверски део базира се на *ESP32* модулу који поседује *Xtensa LX6* микроконтролер. Поред тога контролер је повезан и са сензором *MAX30102* и комуникација се одвија преко *I2C* протокола.

Уређај се напаја 9V батеријом и поседује прекидач за укључење напајања.

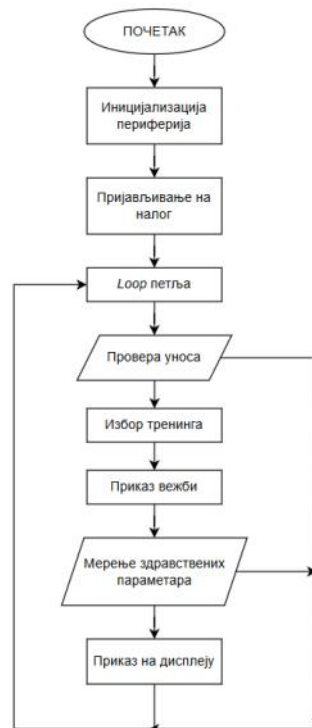


Слика 2. Блок шема хардверског дела система

4. СОФТВЕРСКИ ДЕО СИСТЕМА

4.1. С апликација

На слици 3 приказан је алгоритам рада *main* функције С апликације.



Слика 3. Алгоритам рада *main* функције С апликације

Функција почиње иницијализацијом потребних периферија која је дефинисана у увезеним библиотекама. Након тога улази се *loop* петљу која се стално понавља, и проверава кориснички унос на екрану осетљивом на додир како би се преузеле одређене акције.

Прво је потребно да корисник унесе корисничко име и шифру. Уређај се повезује са сервером како би проверио да ли је унос успешан и ако јесте, уређај од сервера тражи податке о тренингу. Током тренинга се подаци ажурирају у реалном времену, и могуће је помоћу сензора очитати податке о телесној температури, пулсу и нивоу кисеоника у крви.

4.2. Веб апликација - серверски део



Слика 4. Алгоритам рада серверске апликације

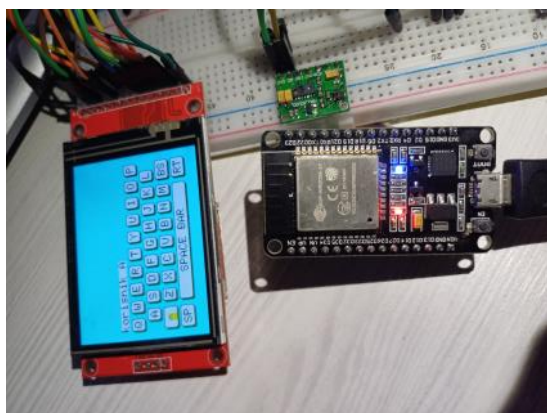
Програмски код главног модула серверске апликације изгледа тако што се прво се учитавају пакети *ws* и *Express.js* који омогућавају комуникацију путем *Websocket* и *HTTP* протокола. *HTTP* захтеви клијената се даље обрађују кроз *MVC (Module, View, Controller)* приступ на порту 2812 [5].

Захтеви се обрађују помоћу дефинисаних рута и организују помоћу модела док је логика за рад са базом података имплементирана у контролерима. На порту 3108 се успоставља *Websocket* конекција и одржава се док год је ембедед уређај активан.

5. РЕЗУЛТАТИ

5.1. Ембедед апликација

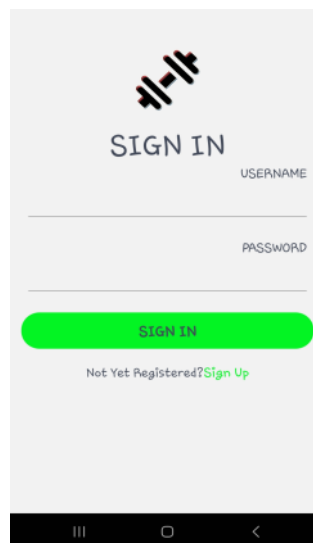
На слици 5. приказан је хардвер из ког се састоји ембедед уређај за праћење тренинга. На дисплеју се види почетни приказ за пријаву корисника.



Слика 5. Хардверско решење

5.2. Android и веб апликација

Android и веб апликација имају исту намену, главна разлика је у платформи за коју су предвиђене. На слици 6. приказана је почетна страница *Android* апликације



Слика 6. Почетна страна *Android* апликације

6. LITERATURA

- [1] Шта је *Internet of Things*, <https://samoobrazovanje.rs/sta-je-internet-of-things-internet-intelligentnih-uredjaja/>, последњи приступ децембар 2023.
- [2] Шта је *Node*? <https://www.skolaprogramiranja.rs/sta-je-to-node/>, poslednji pristup oktobar 2023.
- [3] *WebSocket* vs. *HTTP* <https://sendbird.com/developer/tutorials/websocket-vs-http-communication-protocols> последњи приступ децембар 2023.
- [4] *MongoDB* и *Mongoose* <https://www.mongodb.com/developer/languages/javascript/getting-started-with-mongodb-and-mongoose/>, последњи приступ децембар 2023.
- [5] Шта је *Model-View-Controller* <https://blog.logrocket.com/building-structuring-node-js-mvc-application/> последњи приступ децембар 2023.

Кратка биографија:



Јована Матковић рођена је 28.XII 1999. године. Дипломски рад на Факултету техничких наука из области Мехатронике – Импулсна електроника одбранила је у септембру 2022. године. Има годину дана радног искуства, најпре у компанији *Typhoon-HIL*, а потом у компанији *ARS-Embedded Systems LLC*. Контакт: jovana.matkovic99@gmail.com

ČET-BOTOVI U DOMENU PUTOVANJA I TURIZMA**CHATBOTS IN TRAVEL AND TOURISM**Miloš Živić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu je predstavljen čet-bot koji je sposoban da daje odgovore na pitanja iz domena putovanja i turizma. Pokušano je više pristupa, a dva glavna tipa modela su bazirana na odgovaranju putem izvlačenja i odgovaranju putem generisanja.

Ključne reči: Čet-bot, generisanje, odgovori, putovanja, turizam

Abstract – The paper presents a chatbot capable of answering questions from the field of travel and tourism. Several approaches have been tried, and the two main model types were based on response by retrieval and response by generation.

Keywords: Chatbot, generation, answers, travel, tourism

1. UVOD

Cilj ovog rada je predstavljanje rada čet-botova ali i samih ideja koje su dovele do njihovog nastanka. Prvo bih se zaustavio na prirodi komunikacije između ljudi i čet-botova, a dosta lepo poređenje je dato u radu *Real conversations with artificial intelligence* u kom je upoređena konverzacija između ljudi sa konverzacijom između čoveka i čet-bota. Ovo istraživanje je dovelo do zanimljivih rezultata, a neki od njih su ti da imamo duplo manji broj reči po poruci kod konverzacije sa čet-botom u odnosu na konverzaciju sa čovekom, kao i duplo veći broj poruka.

Ovo ponašanje se može opisati ljudskom tendencijom da razmenjuju manje reči sa osobama koje ne poznaju, a u komunikaciji sa čet-botom se osećaju tako.

Kada pričamo o istoriji nastanka čet-botova važno je pomenuti prevaru zvanu Mehanički turčin. Ova mašina, za koju se kasnije ispostavilo da je pokretana od strane čoveka, naizgled je mogla da igra šah na visokom nivou krajem 18. veka. Iako je sve ovo bila prevara, sama ideja je zaitrigirala razne naučnike i propagirala se do Alana Turinga koji je napisao rad *Computing Machinery and Intelligence*, 1950. godine. U tom radu je predstavljena Igra imitacije koja je kasnije nazvana Turingov test. U igri postoje dva učesnika i sudija koji je čovek. Svaki od učesnika se nalazi u zasebnoj sobi, a na sudiji je zadatak da pogodi da li je u sobi mašina ili čovek. Ako sudija ne uspe da pogodi da se mašina nalazi u sobi to znači da je mašina prošla Turingov test.

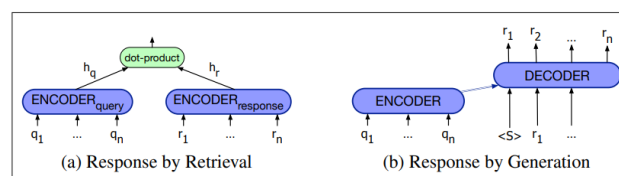
NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Milan Segedinac, vanr.prof.

Celo ovo prenošenje ideja sa generacije na generaciju naučnika dovelo je do prvog pravog čet-bota sposobnog za vođenje konverzacije. ELIZA čet-bot, razvijen 1966. godine spada u prvu od dve veće grupe čet-bototva, a to su čet-botovi bazirani na pravilima, a u ovu grupu takođe spada i PERRY, koji je razvijen šest godina kasnije. Ono što karakteriše čet-botove bazirane na pravilima je relativno mali broj unapred napisanih pravila po kojima ovi čet-botovi treba da se ponašaju. Oni rade tako što iz pitanja korisnika izvlače određene ključne reči, i na osnovu tih ključnih reči oni daju odgovor definisan unapred postavljenim pravilom. Na primer, ako korisnik u pitanju spomene majku ili oca, čet-bot tu reč prepozna kao ključnu i vrati odgovor koji može glasiti „Reci mi više o svojoj porodici“. Ovaj odgovor je dobijen direktnim korišćenjem pravila koje je asociiralo odgovor sa ključnom reči.

U drugu grupu čet-botova spadaju čet-botovi bazirani na korpusu. U ovu grupu spadaju modeli korišćeni u aplikaciji koja je prethodila ovom radu. Ove čet-botove karakteriše veliki skup podataka koji varira od nekoliko stotina hiljada reči, ako govorimo o podskupu čet-botova koji je baziran na izvlačenju, do nekoliko stotina milijardi reči ako govorimo od podskupu koji je baziran na generisanju.

Razliku između ova dva pristupa možemo videti na sl. 1.



Slika 1. Razlika između metode izvlačenja i metode generisanja

Pravljenje ovako velikog skupa podataka definitivno nije lak posao. Neki od načina da se prikupe ovi podaci su upošljavanje velikog broja ljudi da imaju konverzacije na razne teme, često u određenim ulogama ili prikupljanje konverzacija sa društvenih mreža kao što su Facebook, Twitter (sada X), Reddit i druge. Još jedan način je prikupljanje mogućih odgovora sa platformi koje poseduju velike količine informacija i znanja kao što je na primer Wikipedia. Iz ovih podataka model može da nauči da prepriča priču ili vrati određene činjenice.

Nakon ove faze, najčešće sledi faza u kojoj ljudi testiraju ove modele, a konverzacije nastale iz ovog testiranja se kasnije koriste za dotreniranje. Takođe, veoma je bitno da postoje precizne metrike koje mogu nedvosmisleno da nam kažu koji odgovori su dobri, a koji nisu.

2. PODACI

Kao što znamo, kvalitet modela mašinskog učenja zavisi od podataka koje taj model koristi. Što su kvalitetniji podaci to će model bolje raditi. Uzevši ovo u obzir, sastavili smo skup podataka od 5334 pojedinačnih pitanja i odgovora za trening i 60 pitanja i odgovora za testiranje.

Skup podataka za testiranje modela visokog odziva dobijen je prikupljanjem podataka pod nazivom web scraping, a korišćen je alat koji se zove Selenium. Web scraping se odnosi na prikupljanje podataka tako što se programski simulira korisnik i preuzima se sadržaj elemenata na veb stranicama.

Veb sajt koji je korišćen i ovu svrhu jeste Quora, a pristupano je raznim temama vezanim za putovanja i turizam na ovom sajtu.

Test skup podataka kreiran je ručno, biranjem nasumično 20 različitih pitanja i izmenom istih na 3 različita načina: malo (po jedno slovo ili kraća reč), srednje (po duža reč ili fraza) i puno (kompletno parafraziranje pitanja).

Podaci korišćeni za treniranje modela visokog odziva dobijeni su sa Kaggle-a, a tamo se nalaze pod nazivom Quora Question Pair. Skup podataka se sastoji, izuzimajući ID-jeve, od tri kolone: pitanje1, pitanje2 i duplikat. Kolona duplikat je binarna i kao vrednosti ima 0 ili 1, 0 označava da pitanje1 nije duplikat u odnosu na pitanje2, 1 govori da su ova dva pitanja ista, odnosno duplikati.

3. ARHITEKTURA

U ovom poglavlju ćemo obraditi izgled aplikacije i njenu arhitekturu, kao i arhitekturu modela korišćenih za predikciju odgovora na pitanje postavljeno od strane korisnika.

Tehnologije na bekendu i frontendu su iste za sve modele i ta strana aplikacije je fiksna za sve modele. Na bekendu imamo Python-ov radni okvir FastAPI za kreiranje endpointa, dok je na frontendu React.

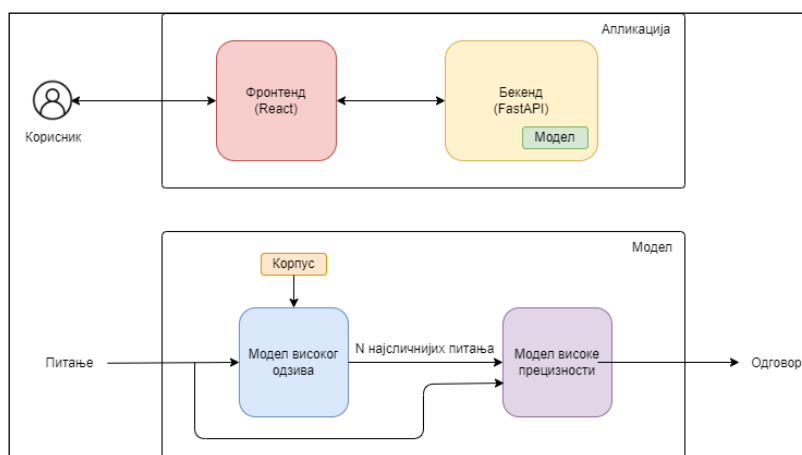
Glavni deo ovog rada je arhitektura samih modela i opis njihovog rada. Imamo dva glavna tipa modela čet-bota: čet-bot baziran na odgovaranju putem izvlačenja i čet-bot baziran na odgovaranju putem generisanja. Dalje, dva dela čet-bota koji daje odgovore uz pomoć izvlačenja su model visokog odziva i model visoke preciznosti. O svemu ovome se više može pročitati u nastavku.

3.1 Čet-bot baziran na odgovaranju putem izvlačenja

Sada ćemo preći na prvi tip modela koji je baziran na odgovaranju putem izvlačenja i sastoji se od modela visokog odziva i modela visoke preciznosti.

U model visokog odziva ulazi pitanje zajedno sa celim korpusom pitanja, a cilj mu je da brzo nadje N najbližijih pitanja. N najbližijih pitanja idu dalje u kompleksniji model visoke preciznosti gde se nalazi najbližije pitanje.

Na slici 2 vidimo kako ova arhitektura izgleda u celosti, a u nastavku ćemo ući u detalje i objasniti kako ona u stvari radi.



Slika 2. Arhitektura aplikacije

3.1.1 Model visokog odziva

Model visokog odziva je deo čet-bota baziranog na odgovaranju putem izvlačenja koji daje N najbližijih pitanja, ne nužno najpreciznije, ali relativno brzo. Cilj je da se najbližije pitanje nađe bilo gde unutar N najbližijih pitanja.

Svi tipovi modela visokog odziva koriste k-NN nakon vektorizacije za izvlačenje N najbližijih pitanja. Pošto odradimo vektorizaciju pitanja treba nam metod kojim bismo odredili koja pitanja su najbližija. Najjednostavniji, ali ne i najbolji pristup je da nakon vektorizacije uporedimo vektor ulaznog pitanja sa svakim vektorom pitanja iz baze pojedinačno. Upoređivanje vektora bi bilo vršeno nekom metrikom sličnosti kao što je euklidska ili kosinusna, i bili bi vraćeni vektori sa sličnošću najbližom jedinici. Ovo bi za velike baze verovatno trajalo predugo. Zato smo u ovoj aplikaciji primenili k-NN algoritam, koji kao izlaz iz algoritma daje najbliže vektore po određenoj metrici.

Da bismo dobili N mogućih kandidata koji bi kasnije ušli u model visoke preciznosti moramo prvo vektorizovati ulazne dokumente. Koristili smo tri različita pristupa: vektorizacija na nivou celog dokumenta, vektorizacija na nivou reči uz pomoć Word2vec algoritma i vektorizacija uz pomoć BERT-a.

Za vektorizaciju na nivou dokumenta smo koristili TF/TF-IDF algoritam. Ovaj algoritam radi tako što gleda koliko često se pojavljuju određene reči u dokumentu, odnosno pitanju. Na kraju dobijemo vektor koji predstavlja broj pojavljivanja svake reči iz vokabulara.

Kada pričamo o vektorizaciji na nivou reči, tu smo probali Word2vec algoritam. Word2vec radi tako što je treniran da predviđi koje reči se nalaze zajedno, to jest u istom kontekstu. Iz ovog treninga dobijamo guste vektore reči, kod kojih se reči koje imaju slično značenje nalaze blizu u vektorskom prostoru.

Poslednji algoritam za vektorizaciju je BERT. BERT je treniran na raznim taskovima iz NLP-ja, tako da nakon ovog treninga sadrži neku predstavu jezika. Ovo mu omogućava da napravi dobre embedinge, odnosno vektore, reči. Verzija BERT-a koja je ovde korišćena se zove TourBERT, a razlika je ta da je ovaj model dotreniran na podacima iz sveta turizma i putovanja, tako da bi trebalo da može da napravi bolju reprezentaciju reči koje su vezane za turizam.

3.1.2 Model visoke preciznosti

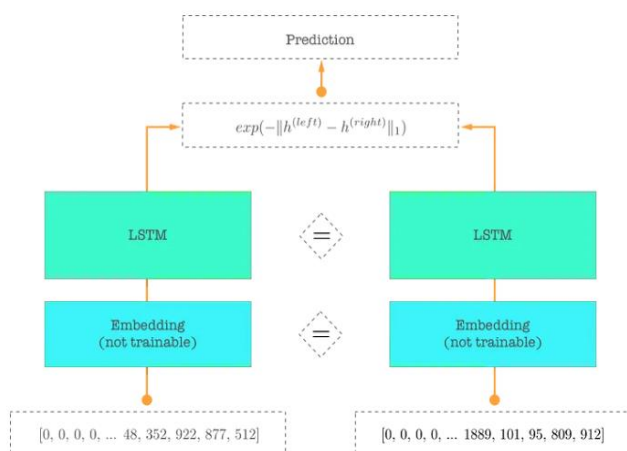
Model visoke preciznosti je drugi deo čet-bota, a služi da od N pitanja dobijenih od modela visokog odziva odluči koje je najbližije ulaznom pitanju postavljenom od strane korisnika. Ovaj model je po pravilu komplikovaniji i radi sporije, ali bi u teoriji bolje mogao da nađe najbližije pitanje.

Model koji je korišćen kao model visoke preciznosti je sijamska LSTM mreža. Ulazi u mrežu su vektori prvog i drugog pitanja, vektorizovani uz pomoć pretreniranog Word2vec modela, a postoji mogućnost korišćenja modela koji je treniran na našim podacima.

Tako vektorizovana pitanja prolaze kroz LSTM mrežu koju možemo zamisliti kao da ima dve putanje, za svako pitanje po jednu, ali realno postoji samo jedna putanja jer LSTM mreža deli težine i upravo zbog toga se i zove sijamska.

Na kraju, uz pomoć Menhetn distance mreža predviđa koliko su ova dva pitanja slična, a labela iz trenning skupa podataka, koja govori da li su ta dva pitanja duplikati se koristi za podešavanje težina mreže.

Na slici 3 možemo videti kako izgleda arhitektura sijamske LSTM neuronske mreže. Kao ulaz imamo vektore koji predstavljaju pitanja, dobijamo embedinge pitanja i šaljem ih u sijamsku LSTM neuronsku mrežu. Dalje se u LSTM mreži proverava da li su ta dva pitanja prafrazirana verzija istog pitanja, tako što na izlazu iz LSTM-a imamo Menhetn distancu koja nam govori da li su ta dva vektora, koji su izlaz iz sijamske LSTM neuronske mreže, dovoljno slični.



Slika 3. Arhitektura sijamske LSTM neuronske mreže

Takođe, treba napomenuti da na slici izgleda kao da imamo dva LSTM-a, ali je to u stvari jedan koji deli težine, kao što je pričano ranije.

3.2 Čet-bot baziran na odgovornju putem generisanja

Do sada smo pričali o modelu koji je pisan od nule, a model je spadao u grupu modela baziranih na odgovaranju putem izvlačenja. U nastavku ćemo pričati o modelima koji su bazirani na generisanju isključivao i koriste se već pretrenirani pozivanjem njihovog API-ja.

Koristili smo generativne modele koji su preuzeti sa Hugging Face platforme. Ova platforma sadrži ogromnu bazu raznih tipova pretreniranih transformer modela i trenutno je najpopularnija u svetu transformera.

Neki od modela koji su korišćeni pretrenirani sa Hugging Face-a su GPT2 i Bloom, ali je korišćen i ChatGPT gađanjem direktno OpenAI API-ja.

4. REZULTATI

Probali smo razne modele visokog odziva na testnom skupu podataka, a u tabeli 1 su navedene performanse nekih od njih. Kao što se na tabeli vidi, najbolje performanse je postigao model koji je koristio Stemming i Stop reči u kombinaciji.

Naravno, performanse ovih modela u velikoj meri zavise od testnog skupa podataka. Ovde je korišćen testni skup od 60 pitanja o kom je bilo priče u poglavlju 2. Podaci. Moguće unapređenje ovog skupa podataka bi bilo njegovo proširenje i pažljiviji izbor pitanja, a ovom aspektu će više reči biti u poglavlju 6. Dalji rad.

Tabela 1. Poređenje performansi različitih modela

Model	Efficiency percent
Stemming	97.8%
Lemmatization	96.4%
N-grams	96.3%
Stemming + Stop words	98.3%
Custom word vectors with IDF	98.0%
Custom word vectors with POS+NER	97.4%
Pretrained word vectors	95.8%
TourBERT word vectors	91.5%

5. ZAKLJUČAK

U aplikaciji je primenjeno više pristupa, kao što sy pravljenje od nule čet-bota baziranog na odgovaranju putem izvlačenja i korišćenje pretreniranih čet-botova koji su bazirani na odgovaranju putem generisanja.

Kada pričamo o prvom tipu čet-bota on radi tako što pokušava da nađe najbližije pitanje iz baze pitanja i odgovora, pitanju koje je postavio korisnik, i da vrati odgovor asociran sa tim pitanjem. Čet-bot se sastoji od dva glavna dela, modela visokog odziva i modela visoke preciznosti. Uloga modela visokog odziva je ta da vrati skup od par desetina ili stotina pitanja za relativno kratko vreme, i time značajno smanji broj pitanja sa kojima model visoke preciznosti treba da radi. Sa druge strane, model visoke preciznosti troši mnogo veće resurse, pa tako i vreme, da bi našao pravo pitanje među onih par desetina ili stotina dobijenih iz modela visokog odziva.

Drugi tip čet-bota koji je korišćen radi na u potpunosti drugačiji način. Treniran je na velikom skupu podataka, i uz pomoć toga je uspeo da stvori neku reprezentaciju znanja, koja je kasnije korišćena za generisanje odgovora.

Oba ova pristupa imaju svoje veoma važne prednosti. Čet-botovi bazirani na odgovaranju putem izvačenja su odlični kada je potrebna preciznost prilikom odgovaranja, ali i kada podaci koji treba da se koriste prilikom odgovaranja na pitanje nisu javno dostupni ili su veoma novi.

U poređenju sa ovim pristupom, čet-botovi bazirani na odgovaranju putem generisanja, daju novu dimenziju odgovaranja. Sposobni su da daju odgovore na pitanja na kojim nisu eksplicitno trenirani gde imaju i određenu dozu kreativnosti i fleksibilnosti koja manjka u prethodnom pristupu.

Naravno, svaki od ovih tipova modela ima i svoje mane. Recimo, kod odgovaranja putem izvačenja postoji ograničenje kod dobijanja odgovora na pitanja koja ne postoje u skupu podataka. Isto tako, kod odgovaranja putem generisanja vrlo lako može da se desi da model da pogrešan odgovor iz pokušaja da odgovori na pitanje na koje ne zna odgovor. Sve ovo gorenavedeno nam govori da ovi modeli nisu savršeni, i imaju svoje prednosti i mane koje je potrebno pravilno adresirati.

6. DALJI RAD

Svakako, postoji dosta prostora kako bi se aplikacija i rad mogli unaprediti. Pre svega, kod modela baziranih na odgovaranju putem izvačenja bismo mogli prikupiti bolje podatke koji bi bili korišćeni prilikom izvačenja odgovora. Takođe, postoji prostor za optimizaciju izvršavanja koda, kao i za optimizaciju samih modela u smislu poboljšavanja pronalaska najbližijeg pitanja. Kao poboljšanje na model koji je baziran na odgovaranju putem generisanja bismo mogli koristiti neki od naprednijih modela za koje su potrebni jači računarski resursi.

Pored ovoga, moglo bi se probati dotreniravanje postojećih modela na manjem skupu podataka, što bi dodatno unapredilo odgovaranje modela na pitanja iz specifičnog domena kao što je turizam. Konačno, poboljšanje bi moglo da ide u smeru kombinacije postojeća dva pristupa, gde bismo koristili RAG.

Treba pomenuti da bi se ovaj čet-bot, kao deo unapređenja, mogao integrisati u aplikaciju razvijanu kao deo diplomskog rada „Sistem za preporučivanje turističkih destinacija“ istog autora i mentora, gde bi doneo dodatnu vrednost i popravio korisnički doživljaj, ali i unapredio samu upotrebljivost obe aplikacije.

7. LITERATURA

- [1] Daniel Jurafsky & James H. Martin, Speech and Language Processing, Chapter 15: Chatbots & Dialogue Systems, Stanford, January 2023
- [2] A. M. TURING, I.—COMPUTING MACHINERY AND INTELLIGENCE, *Mind*, Volume LIX, Issue 236, October 1950, Pages 433–460, <https://doi.org/10.1093/mind/LIX.236.433>
- [3] Joseph Weizenbaum. 1966. ELIZA—a computer program for the study of natural language communication between man and machine. *Commun. ACM* 9, 1 (Jan. 1966), 36–45. <https://doi.org/10.1145/365153.365168>
- [4] Jennifer Hill, W. Randolph Ford, Ingrid G. Farreras, Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations, *Computers in Human Behavior*, Volume 49, 2015, Pages 245–250, ISSN 0747-5632, <https://doi.org/10.1016/j.chb.2015.02.026>
- [5] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, Efficient Estimation of Word Representations in Vector Space, 2013, <https://doi.org/10.48550/arXiv.1301.3781>
- [6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, Attention Is All You Need, 2017, <https://doi.org/10.48550/arXiv.1706.03762>
- [7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018, <https://doi.org/10.48550/arXiv.1810.04805>
- [8] Veronika Arefieva, Roman Egger, TourBERT: A pretrained language model for the tourism industry, 2022, <https://doi.org/10.48550/arXiv.2201.07449>
- [9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (November 15, 1997), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>

Kratka biografija:



Miloš Živić rođen je u Novom Sadu 1999. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva pod nazivom Sistem za preporučivanje turističkih destinacija odbranio je 2022.god.
kontakt: miloszivic99@gmail.com

SISTEM ZA PREGLED I PRETRAGU VESTI BAZIRAN NA GRAPHQL API-U I RAVENDB-U**SYSTEM FOR VIEWING AND SEARCHING NEWS BASED ON GRAPHQL API AND RAVENDB**

Igor Malešević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – PRIMENJENE RAČUNARSKE NAUKE I INFORMATIKA

Kratak sadržaj – Ovaj rad prikazuje pristup razvoju aplikacije za pretraživanje vesti koja se zasniva na GraphQL API-u i RavenDB-u

Ključne reči: GraphQL, RavenDB, Lucene Search Engine

Abstract – This paper proposes an approach in developing an application for viewing and searching news based on GraphQL API and RavenDB.

Keywords: GraphQL, RavenDB, Lucene Search Engine

1. UVOD

U današnjem društvu, pristup informacijama je postao neizostavan deo svakodnevnog života. Razvoj interneta je doprineo ogromnoj količini podataka i znanja koji su nam lako dostupni na samo par klikova, gde god se nalazili. Zahtevi za efikasne metode pretraživanja informacija su postale normativ, kako zbog uštede vremena tako i zbog potrebe za što relevantnijim i tačnijim rezultatima. Jedna od sfera zahvaćena ovim promenama jesu mediji. Novine, kao štampani mediji, televizija i radio postepeno gube korisnike, s obzirom da su sve informacije dostupne na internetu. Iz tog razloga, većina medija su otvorili svoje onlajn veb stranice gde pružaju pregled i pretragu vesti. Na ovakvim stranicama, korisnici očekuju da mogu lako i intuitivno da dođu do informacija koje ih zanimaju uz efikasnu pretragu i filtriranje kako bi suzili skup rezultata na onaj koji je njima najpotrebniji. Još jedan od bitnih aspekata digitalizacije medija jeste proces praćenja i analiza medija, koji je bitan za razumevanje aktuelnih događaja širom sveta.

Ovakvih sistema i veb stranica ima puno, jer veliki broj medija u ovom momentu imaju svoje veb stranice i onlajn portale. Neke od popularnih globalnih veb stranica su Google News, Bing News i News360. Pored globalnih veb stranica, svaka zemlja ima medije koji imaju svoje veb stranice, a primeri popularnih stranih jesu BBC News i CNN, dok domaćih RTS, B92 itd.

U ovom radu, rešavan je problem implementacije jedne veb stranice koja korisnicima omogućava pregled svih

najnovijih vesti. Adresiran je problem velike količine podataka koji mogu dovesti do potencijalnih problema u performansama. Prilikom izrade rada, korišćen je GraphQL kao programski interfejs aplikacije i RavenDB kao baza podataka. Odabir navedenih tehnologija je odgovor na prethodno pomenute potencijalne probleme u performansama.

GraphQL je u prethodnim godinama postao dosta popularniji i zastupljeniji način korišćenja programskog interfejsa aplikacije, te su velike organizacije počele da koriste ovu tehnologiju. Facebook, radeći na svojim aplikacijama, susreo se sa problemom performansi kako su kompleksnost same aplikacije i broj korisnika rasli. Podaci koji se koriste na početnoj strani aplikacije i upiti ka serveru su bili veoma različiti i bilo je neophodno dobiti podatke iz više poziva ka serveru. Ovo je inspirisalo njihove zaposlene da kreiraju GraphQL kao upitni jezik koji im omogućava dobavljanje podataka iz perspektive klijentske aplikacije. Ova tehnologija je zatim javno objavljena 2015. godine [1].

2. PROGRAMSKI INTERFEJS APLIKACIJE

GraphQL [2] je moderan način za kreiranje i pisanje upita za programski interfejs aplikacije (API – Application Programming Interface). Posедуje dvostruku ulogu - sa klijentske strane, to je upitni jezik, dok sa serverske strane predstavlja izvršni okvir (runtime). Klijentska aplikacija koja implementira GraphQL može da definiše i šalje upite na server koji mora da implementira GraphQL servis koji je sposoban da razume upite i odgovori na bilo koji zahtev napisan u GraphQL jeziku. GraphQL karakteristike su:

- Strogo definisani tipovi podataka;
- Tačno definisani podaci;
- Nezavisnost od verzije;
- Hijerarhija;
- Introspekcija.

2.1. Upiti

GraphQL upiti se koriste kada želimo da dobavimo podatke sa servera. Predstavljaju operaciju koja služi isključivo za čitanje podataka, što znači da ne menjaju stanje podataka u bazi. GraphQL upite možemo uporediti sa GET zahtevom u REST API-u. GraphQL upiti nam mogu pomoći u rešavanju takozvanih problema prekomernog (over-fetching) ili nedovoljnog dobavljanja (under-fetching) podataka sa servera.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dragan Ivanović, red. prof.

2.2. Mutacije

GraphQL mutacije predstavljaju mehanizam za izvršavanje promena nad podacima na serverskoj strani. Mutacije omogućavaju klijentima da šalju promenljive ili argumente ka serveru kako bi ažurirali podatke, kreirali nove entitete ili obrisali postojeće. Za razliku od upita, mutacije su destruktivne i menjaju stanje podataka u bazi. *GraphQL* mutacije možemo uporediti sa *POST*, *PUT* i *DELETE* zahtevima u *REST API*-u.

2.3. Pretplate

GraphQL pretplate predstavljaju mehanizam koji omogućava komunikaciju u realnom vremenu između servera i klijenta. Pretplate omogućavaju klijentima da se pretplate na određene događaje ili promene na serveru, a zatim dobiju automatski ažurirane podatke kako se ti događaji dese. To omogućava izgradnju aplikacija kao što su četovi, obaveštenja, novosti itd, gde je komunikacija u realnom vremenu od ključnog značaja. *GraphQL* pretplate koriste *WebSocket* protokol za održavanje dugotrajnih veza između klijenta i servera.

3. SKLADIŠTENJE PODATAKA

Skladištenje podataka je suštinska funkcija baza podataka, gde informacije mogu biti organizovane u tabelama, dokumentima, grafikonima itd. u zavisnosti od vrste baze podataka. Baza podataka je centralizovano mesto gde se čuvaju podaci potrebni za funkcionisanje aplikacije.

3.1. RavenDB

RavenDB je dokument orijentisana (*document oriented*) nerelaciona (*NoSQL*) baza podataka koja se ističe svojom sposobnošću efikasnog rukovanja nestrukturiranim podacima. *RavenDB* čuva podatke u formatu *JSON* (*JavaScript Object Notation*) dokumenata, što olakšava rad sa složenim podacima i omogućuje jednostavnu integraciju sa modernim aplikacijama. Ono što karakteriše i izdvaja ovu nerelacionu bazu jesu *ACID* osobine: [3]

- Atomičnost (*Atomicity*) – sve naredbe jedne transakcije se izvršavaju u celosti ili nikako, čime se osigurava integritet podataka u bazi u slučaju dešavanja greške.
- Konzistentnost (*Consistency*) – baza podataka mora biti u konzistentnom stanju i pre i posle izvršavanja transakcije. Jedna transakcija dovodi bazu podataka iz jednog u drugo konzistentno stanje.
- Izolacija (*Isolation*) – svaka transakcija je izolovana i zaštićena od istovremeno izvođenih drugih konkurentnih transakcija nad istim podacima.
- Trajnost (*Durability*) – Garancija da će svaka uspešno izvršena transakcija biti zapamćena u bazi podataka i u slučaju sistemske greške.

RavenDB koristi *Lucene* biblioteku za pronalaženje informacija (*Lucene Search Engine*). Ova biblioteka predstavlja moćan i otvoren izvor pretraživačkih tehnologija koje omogućavaju brzu i efikasnu pretragu tekstualnih podataka. Indeksiranje predstavlja proces organizovanja i skladištenja podataka u strukturu koja

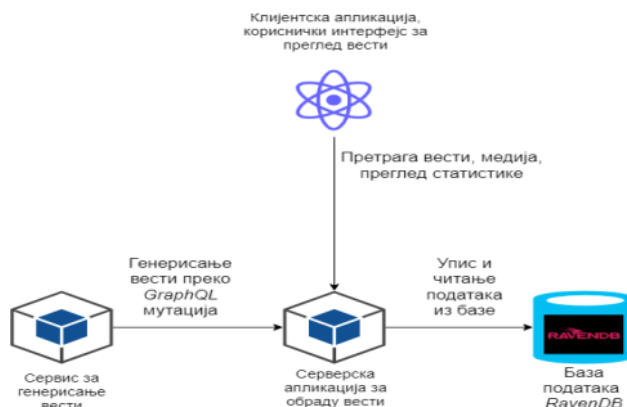
omogućava brzu i efikasnu pretragu. Osnovna ideja indeksiranja je transformacija tekstualnih podataka u optimizovan format koji omogućava brzu pretragu i pruža informacije o lokacijama gde se ti podaci nalaze.

RavenDB radi sa indeksima koji predstavljaju serverske funkcije koje definišu nad kojim poljima i vrednostima iz dokumenta je moguće raditi pretragu. Indeksi predstavljaju jedini način da se izvrše upiti u *RavenDB*-u. Oni se mogu podeliti naspram izvora na statičke i dinamičke, dok se naspram funkcionalnosti dele u tri grupe: [4]

1. *Map Index* – sadrži jednu funkciju mapiranja koja ukazuje koja polja iz dokumenta treba da budu indeksirana.
2. *Multi Map Index* – sadrži više funkcija mapiranja koje dozvoljavaju da se indeksiraju podaci iz više različitih kolekcija (npr. polimorfističnih).
3. *Map Reduce Index* – indeksi koji omogućavaju kompleksne agregacije podataka uz brzo izvršavanje upita nad njima. Pri dodavanju ili izmeni podataka, ovaj indeks će odraditi ponovno kalkulisavanje i uvek imati najsvežije podatke. Indeks se sastoji iz dve faze:
 - a. Faza mapiranja – izvršava funkcije mapiranja nad dokumentima iz navedenih kolekcija.
 - b. Faza stapanja – izvršava funkciju grupisanja podataka po određenom kriterijumu, a zatim radi agregaciju podataka.

4. SPECIFIKACIJA SISTEMA

U ovom poglavlju je predstavljena arhitektura sistema prikazana na slici 4.1.



Slika 4.1. Arhitektura sistema

U sistemu se može identifikovati nekoliko komponenti:

1. Servis za generisanje vesti – Ova komponenta predstavlja serversku aplikaciju koja generiše vesti i pomoću *GraphQL* mutacija poziva serversku aplikaciju za obradu vesti. Ovaj servis je napravljen kako bi simulirao pristizanje velikog broja vesti u sistem.
2. Serverska aplikacija za obradu vesti – Ova komponenta predstavlja glavni servis ovog rada. To je serverska aplikacija koja nam nudi *GraphQL* server sa propisanom šemom i

tipovima podataka. Preko upita nudi mogućnost čitanja podataka iz baze podataka, dok preko mutacija omogućava upisivanje novih podataka. Ima konekciju ka bazi podataka, a predstavlja programski interfejs aplikacije koji konzumira klijentska aplikacija.

3. Baza podataka (*RavenDB*) – Komponenta koja nam služi za skladištenje podataka.
4. Klijentska aplikacija – Ova komponenta predstavlja korisnički interfejs preko kojeg krajnji korisnici imaju mogućnost pregleda i pretrage vesti i medija, kao i prikaz statistike.

5. IMPLEMENTACIJA

U ovom poglavlju je dat detaljniji opis implementacije svake komponente sistema.

5.1. Servis za generisanje vesti

Ovaj servis je napravljen kako bi bilo simulirano kreiranje i pristizanje vesti na serversku aplikaciju. Napravljen je kao *.NET* aplikacija koja ima svoj pozadinski zadatak (*Background job*) i okida se periodično. Period okidanja je podesiv kao i broj podataka koji se generiše. Kako bi sistem ostao u konzistentnom stanju, generišu se svi entiteti koji su potrebni za uspešno indeksiranje podataka i redosled generisanja je bitan. Unutar serverskih aplikacija je iskorišćen *CQRS (Command Query Responsibility Segregation)* šablon [5]. Iz pozadinskog zadatka pozivamo komandu za generisanje vesti.

5.2. Serverska aplikacija za obradu vesti

Ovaj servis predstavlja glavnu serversku aplikaciju koja pruža sve potrebne funkcionalnosti, predstavlja *GraphQL* server i radi indeksiranje podataka koji se skladište u *RavenDB* bazu podataka. Konfiguracija *GraphQL* servera se radi pomoću biblioteke *HotChocolate*. Prilikom konfiguracije same *.NET* aplikacije je potrebno samo navesti da se ona koristi kao *GraphQL* server. Dodatno, pored definisanja *GraphQL* servera, potrebno je definisati tipove upita i mutacija. Ovi tipovi definišu sve moguće upite i mutacije koje je moguće odraditi nad servisom. Definisani upiti i mutacije se oslanjaju na definisane tipove podataka, ulazne argumente i povratne vrednosti. Sve ovo čini *GraphQL* šemu [6].

Validacija podataka i upravljanje greškama je bitan proces unutar ovog servisa. Ovaj proces predstavlja osiguranje da su uneseni podaci u saglasnosti sa zadatim standardima, formatima i pravilima, što sprečava unos pogrešnih ili nevalidnih podataka u sistem i dovođenje sistema u nekonzistentno stanje. Zbog svojih karakteristika, *GraphQL* nudi validaciju na više nivoa, a to obuhvata:

- Semantičku validaciju upita;
- Validaciju tipova podataka;
- Validaciju ulaznih argumenata;

Sledeći nivo validacije podataka predstavlja ispravnost podataka naspram domenske i biznis logike. Ukoliko ova ispravnost podataka nije ispoštovana, takođe je potrebno sprečiti upit da se izvrši i pozivaoca upita je potrebno obavestiti o grešci.

5.3. *RavenDB* baza podataka

Pokretanjem skripte, moguće je dobiti pristup bazi podataka u lokalnom okruženju. Za konfiguraciju konekcije sa bazom podataka, prvo je neophodno dodati zavisnost (*dependency*) na biblioteku *RavenDB.Client*. Dodavanjem ove biblioteke dobija se pristup klasama i metodama koje omogućavaju laku konfiguraciju konekcije. Za uspešnu konekciju, neophodna su dva podatka - *URL* do servera baze i ime konkretne baze podataka nad kojom se izvršavaju upiti.

5.4. Klijentska aplikacija

Ovaj servis predstavlja korisnički interfejs koji konzumira podatke sa serverske aplikacije za obradu vesti i prikazuje te podatke korisnicima. Klijentska aplikacija je implementirana u *React* radnom okviru. Za stilizaciju i komponente interfejsa korišćena je biblioteka *Fluent UI React Northstar*. Dodatno, korišćene su biblioteke koje omogućavaju upravljanje stanjem podataka, tokom podataka i *GraphQL*-om, a one su:

- *React Redux Toolkit* – biblioteka koja predstavlja skup alata koji omogućavaju implementaciju upravljanja stanjem unutar *JavaScript* aplikacija, olakšavajući pristup podacima i ažuriranje podataka. Prednost ove biblioteke dolazi do izražaja u aplikacijama koje koriste veliki broj komponenti i istovremeno moraju da ažuriraju korisnički interfejs na više delova [7].
- *Redux Saga* – biblioteka za upravljanje asinhronim radnjama u *Redux*-u kao što su *HTTP* zahtevi, promene stanja i obrada bočnih efekata.
- *GraphQL Apollo Client* – biblioteka koja se koristi u razvoju veb aplikacija zasnovanih na *GraphQL*-u. Ova biblioteka omogućava laku integraciju i upravljanje *GraphQL*-om zahvaljujući svojim karakteristikama:
 - Lako izvršavanje upita i mutacija.
 - Keširanje što omogućava brze upite i sprečava ponovne zahteve ka serveru za iste podatke.
 - Upravljanje stanjem aplikacije.
 - Integracija sa modernim radnim okvirima.
- *Codegen* – biblioteka koja omogućava korišćenje karakteristike *GraphQL*-a introspekcije, tj. Omogućava da se na klijentskoj strani automatski izgenerišu šema, upiti, mutacije i tipovi podataka koje server nudi. Biblioteka uzima u obzir *GraphQL* fajlove kreirane na klijentskoj strani. Zahvaljujući njoj, eliminiše se potreba za ručnim pisanjem svih navedenih stavki, čime se programeru ubrzava rad sa *GraphQL*-om.

6. ZAKLJUČAK

U ovom radu se mogao videti način implementiranja jedne veb aplikacije za prikaz vesti, koje su postale svakodnevnicom procesom digitalizacije. Ono na čemu je bio akcenat jesu performanse takve aplikacije. Rešenje takvog problema je adresirano na dva načina, a to su

korišćenje *GraphQL API*-a kao korisničkog interfejsa aplikacije i korišćenje baze podataka koja se oslanja na *Lucene Search Engine* kako bi podržala upite i pretrage nad punim tekstom.

GraphQL kao korisnički interfejs aplikacije omogućio je strogo definisanje šeme, upita, mutacija i tipova podataka, čime klijentska aplikacija može da kreira upite i na taj način da odabere samo one podatke koji su joj zaista potrebni. Dobavljanje preterane količine (*over-fetching*) i male količine (*under-fetching*) podataka su problemi koji su rešeni korišćenjem *GraphQL*-a.

RavenDB, kao odabrana baza podataka omogućila je da se uz visoke performanse izvršavaju upiti i skladište podaci. Kako koristi *Lucene Search Engine* u pozadini, to je omogućilo indeksiranje podataka na onaj način koji je optimalan za pretragu vesti koje pristižu u sistem u velikim količinama. Pored pretrage vesti, jako bitnu ulogu je odigrala u geolokacijskoj pretrazi i statistikama.

U ovom radu nije iskorišćen *WebSocket* protokol i *GraphQL* pretplate, te se to može izdvojiti kao potencijalno unapređenje sistema. *GraphQL* pretplate omogućavaju komunikaciju servera sa klijentskom aplikacijom. Prilikom pristizanja novih vesti u sistem bi bilo moguće obavestiti klijentsku aplikaciju o promenama i na neki način korisniku staviti do znanja da postoje najnovije vesti. Korisnički interfejs bi mogao odmah da prikaže novu vest na stranici za prikaz, međutim sa strane korisničkog iskustva, ovo može da bude problem ukoliko vesti pristižu u kratkim intervalima. U tom slučaju, obaveštenje sa određenim detaljima bi bilo dovoljno da se prikaže na korisničkom interfejsu. Dodatno, kako se ne bi gomilali nepotrebni podaci u sistemu i time opterećivali bazu podataka i uticali na performanse sistema, mogli bi se implementirati pozadinski zadaci koji bi se okidali svaki dan i čistili bazu od podataka koji nisu više relevantni.

7. LITERATURA

- [1] Facebook introducing GraphQL, <https://engineering.fb.com/2015/09/14/core-infra/graphql-a-data-query-language/>
- [2] GraphQL, official documentation, <https://graphql.org/>
- [3] RavenDB, ACID database, official documentation <https://ravendb.net/why-ravendb/acid-transactions>
- [4] RavenDB Indexes, official documentation, <https://ravendb.net/docs/article-page/5.4/csharp/indexes/what-are-indexes>
- [5] CQRS Pattern, official Microsoft documentation, <https://learn.microsoft.com/en-us/azure/architecture/patterns/cqrs>
- [6] Schema definition, <https://graphql.org/learn/schema/>
- [7] Marc Garreau, *Redux in action*, 2018, ISBN 9781617294976

Kratka biografija:



Igor Malešević rođen je 19.01.1998. godine u Beogradu. Godine 2016. upisao je Fakultet Tehničkih Nauka u Novom Sadu, odsek Računarstvo i automatika. Osnovne studije završio je u septembru 2020. Od oktobra 2020. Upisuje Master akademske studije.

SOFTVERSKI ALAT ZA OPORAVAK DATOTEKA U SISTEMU DATOTEKA NTFS**SOFTWARE TOOL FOR NTFS FILE SYSTEM FILE RECOVERY**Nikola Milosavljević, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – Gubitak podataka može da ima ozbiljne posledice u savremenom informatičkom okruženju, kako za organizacije tako i za pojedince. Iz tog razloga, oporavak podataka predstavlja ključni aspekt upravljanja podacima. Ova oblast zahteva stalno ažuriranje tehnika oporavka podataka i alata koji implementiraju te tehnike, s obzirom na brz razvoj hardvera i softvera. U radu su analizirane različite tehnike oporavka podataka, a naglasak je stavljen na NTFS fajl sistem. Rad takođe prikazuje implementaciju alata za oporavak podataka, koji je baziran na analizi metapodataka fajl sistema. Demonstrirano je kako se koristi implementirani alat i analizirane su njegove prednosti i mane u odnosu na druga slična rešenja.

Ključne reči: Oporavak podataka, digitalna forenzika, fajl sistema, NTFS

Abstract – Data loss can have serious consequences in modern environments, both for organizations and individuals. For this reason, data recovery is a crucial aspect of data management. This field requires continuous updates to data recovery techniques and tools that implement these techniques, considering the rapid development of hardware and software. This paper analyzes various data recovery techniques, with focus on the NTFS file system. The emphasis is placed on implementation of a data recovery tool based on the analysis of file system metadata. The paper demonstrates how to use the implemented tool and analyzes its advantages and disadvantages compared to other similar solutions.

Keywords: data recovery, digital forensics, file systems, NTFS

1. UVOD

Gubitak podatak se može smatrati ozbiljnom pretnjom za individue, preduzeća i organizacije. Oblast oporavka podataka zahteva ažuriranje metodologija i tehnika u skladu sa brzim tempom razvoja hardvera i softvera. Različite metode oporavka podataka uključuju softverski i hardverski pristup i mogu se koristiti u različitim slučajevima i situacijama.

Motivacija za ovaj rad predstavlja napredovanje informacionih tehnologija i njihov uticaj na bezbednost podataka. Postojeći alati za oporavak podataka nisu u potpunosti ispratili korak napretka tehnologija, što je dodatni stimulans za rad na ovu temu.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Stevan Gostojić, red. prof.

Izvršeno je i poređenje implementiranog rešenja sa trenutnim standardima na tržištu i izvučene su jasne crte napretka i prostor za poboljšanje implementiranog rešenja.

2. FAJL SISTEMI

Fajl sistem je metod i način strukturiranja podataka koji operativni sistemi koriste kako bi kontrolisali kako se podaci čuvaju i dobavljaju. Odgovorni su za organizaciju, skladištenje i upravljanje podacima, čineći ih esencijalnim za svakodnevnu upotrebu računara, pametnih telefona i drugih uređaja. Uprkos njihovoj vitalnoj ulozi, fajl sistemi takođe postaju izvor problema i izazova kada se suočavamo sa situacijama gubitka podataka ili potrebom za digitalnom forenzikom.

2.1. Skladišni medijumi

Skladišni medijum je fizički uređaj koji prima i čuva elektronske podatke koji stižu od operativnog sistema i korisnika i omogućava dostupnost i trajnost podataka. Sami skladišni medijumi se mogu nalaziti u unutrašnjosti kompjutera, kada se zovu interni ili mogu biti priključeni računaru sa njegove spoljašnosti i tada se zovu eksterni skladišni medijumi.

2.1.1 Tvrdi diskovi

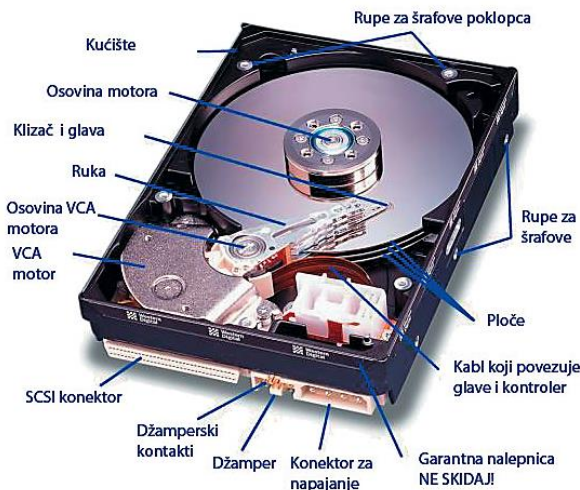
Moderni tvrdi diskovi se ne razlikuju bitno u pogledu delova koji ih sačinjavaju, slika 1. Gledano spolja, na prosečnom tvrdom disku se najpre uočava štampana ploča na kojoj su smeštene komponente koje upravljaju radom uređaja i obezbeđuju stabilno napajanje svih mehaničkih i elektronskih komponenti. Na ovoj ploči se nalaze stabilizatori napona, kontroler, read-only memory (ROM) i random-access memory (RAM). ROM diska je posebno značajan jer sadrži softver koji prilagođava rad konkretnog diska.

Po otvaranju diska vide se ploče na kojima se smeštaju podaci. Moderni tvrdi diskovi koriste ploče koje su najčešće izrađene od feromagnetnog materijala.

Pored ploča vide se i glave koje služe za čitanje i upisivanje podataka. Kako bi se izbeglo trenje i štetno habanje ploča, na tvrdim diskovima glave ne dodiruju ploču već se nalaze na jako malom rastojanju od ploča.

2.1.2. Solid-state diskovi

Solid-state diskovi koriste tip memorije pod nazivom „fleš memorija“ koja je slična RAM-u, ali za razliku od RAM memorije podaci ostaju tu nakon isključivanja računara. Koriste mrežu stranica za brzo slanje i primanje podataka gde je svaka mreža razdvojena na delove koje nazivamo stranice. SSD mogu pisati samo na prazne stranice u bloku.



Slika 1. Prikaz tvrdog diska

Kako je brisanje esencijalna funkcionalnost nekog diska, postavlja se pitanje kako SSD briše podatke. Kada je dovoljno stranica u bloku označeno kao neiskorišćeno, SSD će uzeti sadržaj tog bloka, staviti ga u RAM memoriju i obrisati ceo blok. Onda će uzeti prethodno kopirani sadržaj i upisati ga na novi blok bez neiskorišćenih stranica.

2.2 Definicija fajl sistema

Fajl sistemi donose mehanizam za korisnike da čuvaju podatke u hijerarhiju fajlova i direktorijuma. Fajl sistemi se sastoje od strukturalnih i korisničkih podataka koji su organizovani tako da računar zna gde da ih nađe. U velikom broju slučajeva fajl sistem je nezavistan od bilo kog specifičnog kompjutera [1].

Slično kao i razlika između klase i instance, postoje razlike između tipa fajl sistema i fajl sistema. Tipovi fajl sistema su mnogi i među njih spada New Technology File System (NTFS) koji predstavlja fokus ovog rada. Fajl sistem je konkretan sistem datoteka na nekom konkretnom skladišnom medijumu. Predstavlja instancu jednog tipa fajl sistema sa konkretnim podacima.

2.3 Aspekti fajl sistema

Postoji mnogo aspekata jednog fajl sistema kao što su dozvoljena imena fajlova, upravljanje prostorom, metapodaci i drugi. Fajl sistem detaljno raspoređuje prostor, obično pomoću više fizičkih jedinica na skladišnom medijumu. Njegova osnovna funkcija jeste organizacija datoteka i direktorijuma.

Ime datoteke koristi se kako bi se identifikovalo mesto za skladištenje u fajl sistemu [2]. Većina fajl sistema ima ograničenja u vezi sa dužinom imena datoteke i najčešće su osetljivi na mala i velika slova.

Pored fajlova, fajl sistemi čuvaju i metapodatke vezane svaku datoteku. Dužina fajla, vreme kreiranja, vreme poslednje modifikacije, permisije i tome slično predstavljaju samo neke od metapodataka koje fajl sistem upotrebljava za efikasan rad sa podacima.

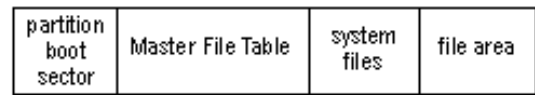
3. NTFS

New Technology File System (NTFS) [3] je fajl sistem dizajniran za pouzdanost, bezbednost i efikasnost prilikom

rukovanja sa velikom količinom podataka. Razvijen je od strane Microsoft-a i počevši od Windows NT 3.1, on je standardni fajl sistem Windows NT porodice zamenivši File Allocation Table (FAT) [4]. Na žalost, ne postoji oficijalna specifikacija toga kako NTFS čuva podatke na disku. Deskripcije visokog nivoa postoje, dok za niži nivo postoje samo nagađanja koja su dobijena analizom.

3.1 Struktura

Formatiranje particije NTFS fajl sistemom kao rezultat daje kreiranje nekoliko sistemskih fajlova.



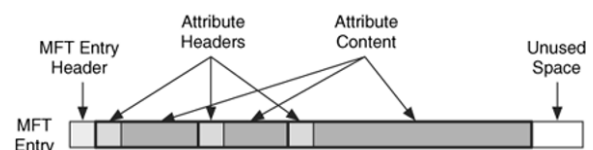
Slika 2. Izgled NTFS particije nakon formatiranja

Na slici 2. se vidi da se NTFS sastoji od nekoliko komponenti. Prva je PBS (partition boot sector) i ovde se nalaze informacije o tome kako fajl sistem treba da koristi particiju. Druga se zove MFT (Master File Table) [5] koja čuva zapis o svim fajlovima i direktorijuma na fajl sistemu. Sledeći je niz drugih meta fajlova koji pomažu u efikasnom radu fajl sistema nakon kojeg dolazi deo za fajlove.

NTFS sadrži nekoliko fajlova koji organizuju fajl sistem. Neki od njih su \$MFT i \$MFTMirr koji predstavljaju master fajl tabelu i njenu rezervnu kopiju koji služe da vode računa o tome gde se nalaze fajlovi na skladišnom medijumu. \$Volume fajl sadrži informacije o particiji, konkretno identifikator objekta particije, oznaku particije i flegove. \$Bitmap čuva niz bitova gde svaki bit pokazuje da li je odgovarajući klaster u upotrebi (alociran) ili slobodan (dostupan za alokaciju). Pored ovih ima i drugih koji u zbiru daju efikasan rad fajl sistema i sveukupno dobro iskustvo upotrebe istog.

3.2 MFT zapis i atributi

Najbitniji meta fajl za ovaj rad je \$MFT koji se sastoji od MFT unosa. Svaki unos predstavlja jednu fajl strukturu na sistemu. NTFS je usvojio pristup da je sve fajl, te je i zapis o \$MFT fajlu sačuvan u samom \$MFT fajlu.



Slika 3. Izgled MFT zapisa

Na slici 3. se vidi prikaz jednog zapisa. Zaglavlje zapisa je fiksne dužine 42 bajta u okviru kojih je specificirana svojstva samog zapisa. Prvo predstavlja magične bajtove NTFS fajla a to su bajtovi „FILE“ ili „BAAD“. Osim ovih, čuvaju se podaci vezani za broj linkova na fajl sistemu, ofset do prvog atributa, identifikator atributa i tome slično. Svaki zapis je 1024 bajta i sav prostor izuzev zaglavlja je rezervisan za attribute.

Svaki MFT zapis može imati više atributa vezanih za sebe. Sami atributi imaju tip, opciono ime atributa i vrednost predstavljenu kao niz bajtova. Atributi mogu biti rezidentni ili nerezidentni u zavisnosti od broja bajtova koje njihova vrednost uzima. Ako vrednost atributa može stati u jedan MFT zapis, tada je atribut rezidentan.

Ako ne može, što je slučaj sa DATA atributom, tada se kao vrednost atributa čuva mapa alokacija koja pokazuje gde se na disku zapravo nalaze podaci vezani za vrednost ovog atributa i koje su oni dužine.

Pored već spomenutog DATA atributa postoje razni, od kojih je bitan STANDARD_INFORMATION atribut koji čuva informacije o tome kad je fajl napravljen, poslednji put otvaran, permisije i tome slično. ATTRIBUTE_LIST je atribut koji se umeće kada bi broj atributa prevazišao veličinu od 1024 bajta kako bi se zapis jedne datoteke raširio na više MFT zapisa.

3.3 Interoperabilnost

NTFS fajl sistem ima mogućnost i drajvere za upotrebu na različitim operativnim sistemima. Iako je napisan za Windows, i najčešće je u potpunosti kompatibilan unazad i unapred, postoje tehnički faktori koje treba uzeti u obzir prilikom maotovanja novih NTFS fajl sistema na starijim verzijama Windows operativnog sistema.

Verzije Linux kernela od 2.1.74 uključuju drajver napisan od strane Martina fon Luisa koji ima sposobnost čitanja NTFS particija. Tek od verzije 5.15 [6] Paragonov NTFS drajver je integrisan i podržava čitanje i pisanje običnih i kompresovanih datoteka.

4. OPORAVAK DATOTEKA

U računarstvu, oporavak datoteka je proces vraćanja obrisanih, nedostupnih, izgubljenih, oštećenih ili formatiranih podataka sa nekog skladišta ili medija kada se podacima u njima ne može pristupiti na običan način. Oporavak može biti potreban zbog fizičkih ili logičkih oštećenja na uređajima za skladištenje ili logičkih oštećenja na fajl sistemu koje sprečavaju da se fajl sistem mauntuje od strane operativnog sistema.

Logičke greške se dešavaju kada su tvrdi diskovi funkcionalni ali korisnik ili operativni sistem ne može povratiti ili pristupiti podacima sačuvanim na njima. Mogu nastati zbog izgubljenih particija, oštećenja čipa, greške u radu firmware-a ili greške pri formatiranju.

Širok spektar neispravnosti može izazvati fizičku štetu na skladišnim medijumima. Ona se mogu desiti zbog ljudskih grešaka i prirodnih katastrofa. CD-ROM može imati oštećen metalni substrat ili sloj boje, tvrdi diskovi mogu patiti od mnogih mehaničkih neispravnosti rada glava za čitanje, motora ili se trake jednostavno mogu slomiti.

4.1 Metode oporavka podataka

Postoje dve metode oporavka podataka, oporavak podataka analizom metapodataka i oporavak podataka pretragom za poznate tipove podataka. Obe imaju svoje prednosti i mane i koriste se u različitim slučajevima.

Oporavak podataka analizom metapodataka je najčešće prvi način kojim program za oporavak podataka pokušava da oporavi podatke. Čitanjem metapodataka iz fajl sistema mogu se oporaviti fajlovi sa svojim originalnim imenom, putanjama, datumom kreiranja i poslednjim modifikacijama i permisijama. U ovom slučaju, program pokušava da pronađe informacije o fajlovima koji su dealocirani (prostor koji oni zauzimaju je označen kao slobodan za alociranje), a kada ih nađe, on prekopira sadržaj fajla na neko novo mesto, prateći instrukcije iz metapoda-

taka za to gde se ti podaci nalaze. Ovaj metod ne radi ako je po disku puno pisano otkako su podaci izgubljeni ili ako je instanciran novi fajl sistem na disku jer se svi metapodaci tada brišu. Često daje zadovoljavajuće rezultate, ali u forenzičkim istragama se nekad pribegava i sofisticiranijoj drugoj metodi.

Iako uspešnija od prve metode po količini pronađenih podataka ova metoda ne može povratiti same metapodatke kao što su ime, putanja, permisije i tome slično. Zasniva se na ideji da se ceo disk skenira i traže se potpisi fajlova. Potpisi fajlova su magični bajtovi koji opisuju određen tip fajlova. Kako ovi potpisi nisu standardizovani ovo predstavlja veliki problem za ove alate. Postoje fajlovi koji imaju početni potpis, neki imaju samo krajnji potpis, dok treći imaju i početni i krajnji potpis. Veliki problem je oporavak fajlova koji su isprekidani na disku. Međutim i ovi problemi su donekle prihvatljivi jer se najčešće ovom metodom oporavlja dobar deo neophodnih informacija iako je ona dosta sporija od prve metode.

4.2 Step en uspešnosti oporavka podataka

Faktori koji igraju ključnu ulogu za ishod oporavka podataka su kvalitet softvera za oporavak podataka, ispravnost akcija prilikom vršenja oporavka podataka i stepen oštećenosti skladišnog medijuma. Prva i druga stavka su u velikoj meri kontrolisane i svi uključeni imaju za cilj da oporavak prođe dobro. Poslednji faktor je izvan svačije kontrole i činjenica je da je on od presudnog značaja.

Dobra stvar je što se velika oštećenja na disku dešavaju jako retko.

Samo pisanje velike količine podataka na disk može da ga ošteti do te mere da se ne može oporaviti. U velikom broju slučajeva se izgube samo metapodaci i moguće je primenom druge metode oporaviti veliku količinu nefragmentiranih podataka.

4.3 Faze oporavka podataka

Proces se sastoji od četiri faze. Prva faza je popravka tvrdog diska gde se on osposobljava za rad. Mogu se zameniti neispravne glave, motor i tome slično.

Druga faza je pravljenje kopije diska na drugi disk ili u fajl sa slikom diska, kako bismo osigurali da su podaci očuvani na originalu i da se oni neće menjati a dalji rad se vrši sa kopijom.

Treća faza je logički oporavak fajlova, particija, MBR i drugih struktura fajl sistema. Ako je disk logički neispravan, postoji nekoliko razloga za to i koristeći klon moguće je popraviti tabelu particija ili MBR kako bi se pročitala struktura podataka fajl sistema.

Poslednja faza je vezana za popravku oštećenih fajlova koji su povraćeni. Oštećenje može da se desi kada se podaci upišu na logički neispravan sektor na disku. U nekim slučajevima je moguće u potpunosti oporaviti oštećene fajlove.

5. ALAT ZA OPORAVAK PODATAKA

Cilj alata jeste da podmladi stara rešenja za oporavak podataka i da usput postigne idealno iste ili bolje rezultate upotrebom metode oporavka podataka na osnovu metapodataka.

Okvir samog projekta je postavljen tako da alat podržava samo NTFS fajl sistem sa kojeg će pokušati oporavak podataka. Takođe, alat može biti pokrenut isključivo sa Linux operativnih sistema koji ujedno predstavlja i očekivano razvojno okruženje.

Funkcionalnosti alata su detekcija fajl sistema, pri čemu ako fajl sistem nije NTFS alat trenutno prekida sa radom, parsiranje PBS (partition boot sector) kako bi utvrdio gde se nalazi MFT, čitanje i parsiranje MFT tabele i oporavak obrisanih podataka iz rezidentnih i nerezidentnih atributa. Pored toga, alat je namenjen za upotrebu putem komandne linije, gde nudi mali grafički interfejs za selekciju pronađenih fajlova koje treba povratiti.

5.1 Funkcionalni zahtevi

Korisnik treba da može da izabere opciju između oporavka podataka sa mauntovanog diska ili oporavka na osnovu slike diska. Alat mora da ima podršku za skeniranje diska/slike. Ovo predstavlja mogućnost detekcije fajlova na disku bez njihovog oporavka. Alat mora da ima mogućnost da zaista oporavi podatke koji su pronađeni na osnovu skeniranja diska/slike upotrebom metode oporavka na osnovu metapodataka. Takođe, treba da može da oporavi njihovu tačnu strukturu u što je većem stepenu moguće. Kao sigurnosnu funkcionalnost alat mora da omogući log zapis svih akcija i koraka koje preduzima da ispuni svoj cilj.

Da bi ispunio ove funkcionalnosti, alat nakon parsiranja validira unesene parametre i ako su validni inicijalno pokušava da skenira disk i pronađe MFT i zapise u njemu. Nakon toga prolazi kroz iste i prezentuje korisniku one koji su označeni kao nealocirani, nakon čega korisnik treba da izabere koje fajlove želi da oporavi. Ako alat radi u režimu koji nije „dry-run“ režim, alat ih oporavlja.

6. ZAKLJUČAK

Iako je alat za oporavak podataka implementiran uspešno u okvirima koji su predviđeni specifikacijom, postoje određena ograničenja. Prvo ograničenje je starost skladišnog medijuma. Što je skladišni medijum stariji i što se više koristi, manje su šanse za savršen oporavak. Stepenn degradacije se može umanjiti samo ako se zaplenjeni skladišni medijum odmah klonira i samim tim onemogućiti da se ošteti kao digitalni dokaz.

Implementirani alat je testiran na tvrdom disku (koji je magnetni) i na USB fleš disku (koji predstavlja fleš memoriju). Nisu primećene značajne razlike u brzini između ova dva skladišta podataka, ali to može biti zbog približno jednakih veličina MFT.

Stepen fragmentacije nije nešto što utiče na rad alata direktno. Alat je dizajniran tako da prolazi kroz sve mape alokacije vezane za fajl i spaja ih u jedan kontinualni fajl na određišnom fajl sistem. Kod fragmentiranih fajlova, ponašanje alata slično je kao i ponašanje njemu sličnih.

7. LITERATURA

- [1] B. Carrier, File System Forensic Analysis: What is a File System?, Massachusetts: Addison-Wesley Professional, 2005.
- [2] B. Carrier, File System Forensic Analysis: File name category”, Massachusetts: Addison-Wesley Professional, 2005.
- [3] B. Carrier, File System Forensic Analysis: NTFS Concepts”, Massachusetts: Addison-Wesley Professional, 2005.
- [4] “File Allocation Table” – FAT16 F, FATX F. [Online]. Available: http://cs.williams.edu/~jannen/teaching/s19/cs333/readings/FAT/File_Allocation_Table-Wikipedia.pdf. [Accessed: Sep. 16, 2023]
- [5] “NTFS Master File Table (MFT)” [Online]. Available: <https://www.ntfs.com/ntfs-mft.htm>. [Accessed: Sep. 27, 2023]
- [6] “Linux kernel version history” Jan. 11, 2020. Available: https://en.wikipedia.org/wiki/Linux_kernel_version_history#Releases_5.x.y. [Accessed: Sep. 23, 2023]

Kratka biografija:



Nikola Milosavljević rođen je 4.7.1999. u Novom Sadu, Vojvodini. Pohadao je gimnaziju „Isidora Sekulić“. Fakultet tehničkih nauka upisao je 2018. na smeru Računarstvo i automatika. Master studije na studijskom programu Računarstvo i automatika upisuje 2022. godine.

ESTIMACIJA RADIO KANALA NA OSNOVU PILOTSKIH SEKVENCI PILOT SEQUENCES ASSISTED RADIO CHANNEL ESTIMATION

Aleksandar Birmančević, *Fakultet Tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIČKO I RAČUNARSKO INŽENJERSTVO

Kratak sadržaj – *Estimacija radio kanala u ovom radu vezana je za postupak procene sistemske funkcije korisničkih kanala. Procene su potpomognute Zadoff-Ču pilotskim sekvencama koje se dodeljuju pojedinim korisnicima. Linearni Bajesov estimator procenjuje trenutno stanje kanala minimizovanjem svog rizika, gde je funkcija cene - srednja kvadratna greška. Estimator unosi grešku procene koja, kada je minimizovan Bajesov rizik, dostiže svoju minimalnu vrednost među svim linearnim/afinim estimatorima. Kvalitet procene kanala opisan je normalizovanim srednjim kvadratnim greškama.*

Ključne reči: *Bajesov estimator, radio kanal, antensko polje, pilotska sekvenca, pilotska kontaminacija*

Abstract - *The Radio Channel Estimation in this paper is related to the procedure of evaluating the system function of the user channels. These evaluations are assisted by the Zadoff-Chu pilot sequences that are assigned to users. A linear Bayes estimator estimates the instantaneous channel state by minimizing its risk, where the cost function is the Mean-Squared Error (MSE). The estimator induces an estimation error that reaches its minimum value among all linear/affine estimators once the Bayes risk is minimized. The quality of channel estimation is described by the Normalized Minimum Mean-Squared Errors (NMSEs).*

Keywords: *Bayes estimator, radio channel, antenna array, pilot sequence, pilot contamination*

1. UVOD

Povećanje broja korisničkih terminala, ali i minimalnih zahteva za nesmetan rad servisa koji se pokreću na ovim terminalima, zahtevaju ulaganja u dalji razvoj mrežne infrastrukture i tehnika prenosa informacija. Kada bazna stanica razlikuje dolazne uglove talasa na svom antenskom polju pod kojim ih korisnički terminali emituju, umetnuti signali u ovim talasima mogu da se ponavljaju u istim vremenskim trenucima, da zauzimaju identičan frekventni opseg, pa čak i da prenesu istu informaciju. Bazna stanica će uspešno izdvojiti korisničke informacije i pridružiti ih svakom korisniku

koji ih je poslao. Na ovaj način, u uzlaznom smeru prenosa se štede vremensko-frekventni resursi u okviru ćelije koju opslužuje pomenuta bazna stanica. Uspešno izdvajanje korisničkog signala zavisi od ključnog faktora - kvaliteta procene njegovog kanalnog odziva, naročito u otežanim okolnostima kada su prostorni uglovi dolaznih talasa više korisnika vrlo bliski, na šta se ovaj rad orijentiše.

2. MOBILNI RADIO KANAL

Mobilni radio kanal predstavlja linearni vremenski promenljiv filter. Neka se korisnički signal sastoji od jednog simbola, $x(t)$ i prenosi kroz kanal koji, na osnovu [1], preslikava jedan ulaz na konačno mnogo prebrojivih izlaza, $y_m(t; \mathbf{p}_m)$, $m \in \{1, \dots, M_j\}$. Relacija ulaz-izlaz za ovakav filter se, prema slici 1,

$$\underline{x}(t) \longrightarrow \boxed{\underline{g}_m(\tau, t)} \longrightarrow \underline{y}_m(t; \mathbf{p}_m)$$

Slika 1: Blok šema razmatranog (mobilnog) radio kanala.

zapisuje u obliku,

$$\begin{aligned} \underline{y}_m(t; \mathbf{p}_m) &= \int_{\mathbb{R}} \underline{g}_m(\tau, t) \underline{x}(t - \tau) d\tau \\ &= \int_{\mathbb{R}} \int_{\mathcal{S}^2} \underline{g}_m(\tau, t, \omega) \underline{F}_m(\omega) \underline{x}(t - \tau) d\omega d\tau. \end{aligned} \quad (1)$$

$\mathcal{S}^{2 \times 1}$ označava skup svih tačaka koje zadovoljavaju jednačinu sfere $x^2 + y^2 + z^2 = 1$, pri čemu $\{x, y, z\} \in \mathbb{R}^3$. Jedinični vektor pravca dolaznog ravnog talasa, ω dat je u sfernom koordinatnom sistemu tako da važi

$$\omega = \{\phi, \theta\} \in \mathcal{S}^{2 \times 1} \subset \mathbb{R}^{3 \times 1}. \quad (2)$$

Ugao $\phi \in [-\pi, \pi]$ vezan je za azimut, dok je elevacioni ugao, $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. Vektor ω se predstavlja u Dekartovom koordinatnom sistemu na osnovu sfernih koordinata prema

$$\omega = -[\cos(\theta)\cos(\phi), \cos(\theta)\sin(\phi), \sin(\theta)]_{3 \times 1}^T. \quad (3)$$

Impulsni odzivi $\underline{g}_m(\tau, t)$ i $\underline{g}_m(\tau, t, \omega)$ odgovaraju radio i propagacionom kanalu. Pri tome je $\underline{g}_m(\tau, t, \omega)$, dat Furijeovom transformacijom

$$\underline{g}_m(\tau, t, \omega) = \int_{\mathbb{R}} \underline{g}_m(\tau, \nu, \omega) e^{j2\pi\nu t} d\nu, \quad (4)$$

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio doc. dr Milan Narandžić.

pri čemu za konačno mnogo propagacionih putanja, N , važi da je

$$\underline{g}_m(\tau, \nu, \omega) = \sum_{n=1}^N \alpha_n \delta(\tau - \tau_n) \delta(\nu - \nu_n) \delta(\omega - \omega_n). \quad (5)$$

Koeficijent slabljenja putanje $n \in \{1, \dots, N\}$ označen je sa $\alpha_n \in \mathbb{C}$. Dirakovi impulsi obeleženi su sa $\delta(\tau): \mathbb{R} \mapsto \mathbb{R}$, $\delta(\nu): \mathbb{R} \mapsto \mathbb{R}$ i $\delta(\omega): \mathcal{S}^2 \mapsto \mathbb{R}$.

Kompleksna skalarna vrednost karakteristične funkcije zračenja kompleksnih vrednosti za element m , $\underline{E}_m(\omega)$ data je sa

$$\underline{E}_m(\omega) = a_m(\omega) e^{-j2\pi\langle\omega, \mathbf{q}_m\rangle}, \quad (6)$$

pri čemu je $a_m(\omega)$ dijagram zračenja intenziteta električnog polja antenskog elementa m , dok je $2\pi\langle\omega, \mathbf{q}_m\rangle$ fazni pomak antenskog elementa m . Normalizovani vektor položaja ovog elementa antenskog polja obeležen je sa $\mathbf{q}_m \in \mathbb{R}^3$,

$$\mathbf{q}_m = \frac{\mathbf{p}_m}{\lambda}, \quad (7)$$

znajući da je λ talasna dužina. Unutrašnji proizvod između ω i $\mathbf{q}_m$, iznosi

$$\langle\omega, \mathbf{q}_m\rangle = \mathbf{q}_m^H \omega = \mathbf{q}_m^T \omega. \quad (8)$$

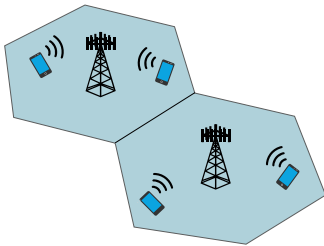
U radu su usvojene sledeće pretpostavke:

- antenski elementi su izotropne antene koje podjednako zrače u svim pravcima, a postavljene su u niz na jednakom međusobnom rastojanju od polovine talasne dužine;
- kanal je vremenski nepromenljiv u trajanju njegovog vremena koherencije, T_c koje je duže od trajanja (sekvence) simbola za prenos;
- kanal je frekvencijski neselektivan, odnosno opseg koherencije kanala, B_c širi je od spektra signala koji se prenosi.

Vremensko-frekvencijski neselektivan impulsni odziv radio kanala izotropnog elementa m tokom vremena koherencije i u opsegu koherencije obeležava se sa $g_m \in \mathbb{C}$, te više nije funkcija ni vremena, a ni frekvencije.

3. MODEL SISTEMA

Primljeni signal na jednoj baznoj stanici j od ukupno L , sa stanovišta $\tau_p + \tau_u$ odbiraka u vremenu može se u domenu amplituda prema slici 2 zapisati kao matrica, $\mathbf{Y}^j \in \mathbb{C}^{M_j \times (\tau_p + \tau_u)}$.



Slika 2: Skica postavke sistema u uzlaznom smeru prenosa

Zapis je dat na sledeći način:

$$\mathbf{Y}^j = \underbrace{\sum_{k=1}^K \mathbf{g}_{jk}^j \mathbf{x}_{jk}^T}_{\text{Željeni signali}} + \underbrace{\sum_{l=1, l \neq j}^L \sum_{i=1}^K \mathbf{g}_{li}^j \mathbf{x}_{li}^T}_{\text{Međućelijska interferencija}} + \underbrace{\mathbf{N}^j}_{\text{Šum}}. \quad (9)$$

Korisnički signal koji dospeva iz iste ćelije dat je kao $\mathbf{x}_{jk} \in \mathbb{C}^{(\tau_p + \tau_u) \times 1}$, za razliku od poslatog signala korisničkog terminala izvan ćelije j , opisanog sa $\mathbf{x}_{li} \in \mathbb{C}^{(\tau_p + \tau_u) \times 1}$. Kanalni odzivi i šum označeni su sa $\mathbf{g}_{jk}^j \in \mathbb{C}^{M_j \times 1}$, $\mathbf{g}_{li}^j \in \mathbb{C}^{M_j \times 1}$ i $\mathbf{N}^j \in \mathbb{C}^{M_j \times (\tau_p + \tau_u)}$, tim redom.

Za prvih τ_p odbiraka složenog signala $\mathbf{Y}^j$ namenjenih prenosu pilotskih sekvenci, može se definisati oznaka, $\mathbf{Y}_j^p \in \mathbb{C}^{M_j \times \tau_p}$. Korisnički signali u osnovnom opsegu tokom τ_p odbiraka, $\mathbf{x}_{jk} \in \mathbb{C}^{\tau_p \times 1}$ i $\mathbf{x}_{li} \in \mathbb{C}^{\tau_p \times 1}$ postaju

$$\mathbf{x}_{jk} = \sqrt{p_{jk}} \phi_{jk} \quad (10)$$

$$\mathbf{x}_{li} = \sqrt{p_{li}} \phi_{li}, \quad (11)$$

pri čemu se pretpostavlja da su korisničke snage $p_{jk} \in \mathbb{R}$ i $p_{li} \in \mathbb{R}$ nepromenljive u trajanju τ_p odbiraka. Karakteristike korisničkih pilotskih sekvenci, $\phi_{jk} \in \mathbb{C}^{\tau_p \times 1}$ i $\phi_{li} \in \mathbb{C}^{\tau_p \times 1}$ razmatraju se u sekciji 4.1. Poželjna je situacija u kojoj svi korisnici poseduju međusobno ortogonalne pilotske sekvence unutar ćelije.

Proizvod vremena i opsega koherencije u vremensko-frekvencijskoj ravni određuje dimenzije pravougaoznika koji nosi naziv „koherentni blok”. Vremensko-frekvencijski resursi koherentnog bloka podeljeni su u vremenu na dva dela: uzlazni i silazni smer prenosa informacija. Ovaj protokol zadržava osobinu recipročnosti fizičkog kanala, bez obzira na smer prenosa.

4. ESTIMACIJA STANJA KANALA

Korisnička pilotska sekvenca predstavlja skup znakova za raspoznavanje u svrhu estimacije njegovog kanala, pored drugih namena: identifikacije korisnika i sinhronizacije prenosa. U ovom radu pilotska sekvenca se upotrebljava isključivo za prvu od prethodno navedenih namena.

4.1. Dizajn pilotskih sekvenci

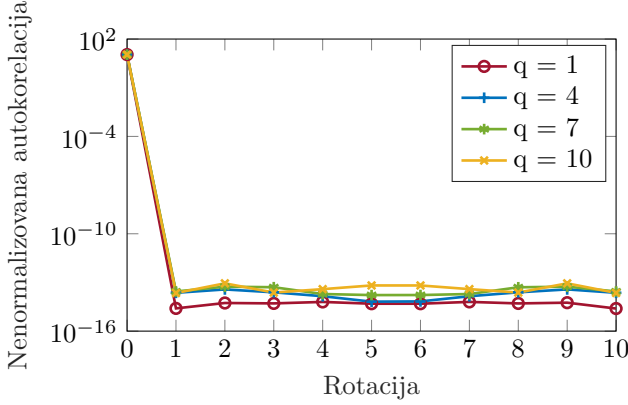
Optimalna knjiga pilotskih sekvenci za τ_p korisnika može se zapisati kao $\Phi \in \mathbb{C}^{\tau_p \times \tau_p}$, pri čemu $\Phi^H \Phi = \tau_p \mathbf{I}_{\tau_p}$ ukazuje na poželjnu osobinu međusobne ortogonalnosti svih njenih vektor-kolona, odnosno njenih τ_p korisnika. Unutrašnji proizvod nad Φ između bilo koje pilotske sekvence (uključujući i sekvencu ϕ_{jk}) sa sekvencom ϕ_{jk} je,

$$\phi_{l'i'}^T \phi_{jk}^* = \begin{cases} \tau_p & \text{ako } (l', i') \in \mathcal{P}_{jk} \\ 0 & \text{inače.} \end{cases} \quad (12)$$

Skup $\mathcal{P}_{jk}$ sadrži sve korisnike kojima je dodeljena identična pilotska sekvenca kao korisniku k iz ćelije j ,

$$\begin{aligned} \mathcal{P}_{jk} = \{ (l', i') : \phi_{l'i'} = \phi_{jk} \\ l' = 1, \dots, L, i' = 1, \dots, K \\ (l'', i'') : \phi_{l''i''} = \phi_{jk} \\ l'' = 1, \dots, L, i'' = 1, \dots, K \}. \end{aligned} \quad (13)$$

Modeluju se Zadof-Ču sekvence prema teorijskim osnovama iz rada [2]. Na slici 3, svi parovi dužine sekvenci $N_{\text{Zadof-Ču}} = 11$ i indeksa korena $q \in \{1, \dots, N_{\text{Zadof-Ču}} - 1\}$ su relativni prosti brojevi. Primećuje se minimalno odstupanje od željenih vrednosti autokorelacija/međukorelacija za idealne sekvence iz izraza (12), koje može biti posledica konačne preciznosti korišćene računarske platforme.



Slika 3: Autokorelacije Zadof-Ču sekvenci dužine 11

4.2. Pilotska kontaminacija

Pilotska kontaminacija je vid korisničke interferencije kao posledice nedovoljnog broja jedinstvenih pilotskih sekvenci za dodelu korisnicima. Dodatno, svi korisnici koji dele (ponavljaju) takve pilotske sekvence poseduju međusobno neortogonalne prostorne autokorelacione matrice impulsnih odziva radio kanala. U takvim slučajevima, spektralna efikasnost opada sa povećanjem stepena ponavljanja date pilotske sekvence. Dakle, kada postoje korisnici koji dele pilotsku sekvencu, kontaminacija nastupa u trenucima neortogonalnosti prostornih korelacionih matrica korisnika ponavljajuće sekvence.

4.3. Linearni Bajesov estimator

Bajesovi estimatori kanalnog odziva minimizuju rizik - očekivanu vrednost cene za datu funkciju cene,

$$\mathbb{E}\{l(\mathbf{g}_{jk}^j, \hat{\mathbf{g}}_{jk}^j(\mathbf{y}_{jk}^j))\}, \quad (14)$$

gde je funkcija cene $l(\cdot, \cdot)$, estimirani vektor je $\hat{\mathbf{g}}_{jk}^j(\mathbf{y}_{jk}^j)$, dok je apriori nepoznat vektor $\mathbf{g}_{jk}^j$. Za Bajesov rizik definisano je mnoštvo funkcija cene, tako da dizajner estimatora brine o izgledu funkcije cene. Neka je odabrana ℓ^2 -norma vektora greške procene,

$$\tilde{\mathbf{g}}_{jk}^j = \mathbf{g}_{jk}^j - \hat{\mathbf{g}}_{jk}^j(\mathbf{y}_{jk}^j), \quad (15)$$

tako da je funkcija cene

$$l(\mathbf{g}_{jk}^j, \hat{\mathbf{g}}_{jk}^j(\mathbf{y}_{jk}^j)) = \|\mathbf{g}_{jk}^j - \hat{\mathbf{g}}_{jk}^j(\mathbf{y}_{jk}^j)\|^2. \quad (16)$$

Linearni estimator minimalne srednje kvadratne greške procenjuje odziv kanala na osnovu izraza

$$\tilde{\mathbf{g}}_{jk}^j = \mathbf{A}_{jk}^j \mathbf{y}_{jk}^j + \mathbf{b}_{jk}^j, \quad (17)$$

gde se kvadratna matrica $\mathbf{A}_{jk}^j \in \mathbb{C}^{M_j \times M_j}$ i vektor-kolona $\mathbf{b}_{jk}^j \in \mathbb{C}^{M_j \times 1}$ združeno biraju u cilju minimizacije Bajesovog rizika za pomenuti estimator. Bajesov

rizik, $\mathbb{E}\{\|\mathbf{g}_{jk}^j - (\mathbf{A}_{jk}^j \mathbf{y}_{jk}^j + \mathbf{b}_{jk}^j)\|^2\}$ minimizuje se odabirom parametara $\mathbf{A}_{jk,\min}^j$ i $\mathbf{b}_{jk,\min}^j$ na sledeći način:

$$\begin{aligned} \mathbf{A}_{jk,\min}^j &= \mathbf{C}_{\mathbf{g}_{jk}^j \mathbf{y}_{jk}^j}^j \mathbf{C}_{\mathbf{y}_{jk}^j \mathbf{y}_{jk}^j}^{j-1} \\ \mathbf{b}_{jk,\min}^j &= \mathbb{E}\{\mathbf{g}_{jk}^j\} - \mathbf{C}_{\mathbf{g}_{jk}^j \mathbf{y}_{jk}^j}^j \mathbf{C}_{\mathbf{y}_{jk}^j \mathbf{y}_{jk}^j}^{j-1} \mathbb{E}\{\mathbf{y}_{jk}^j\}. \end{aligned} \quad (18)$$

tako da je estimator oblika

$$\hat{\mathbf{g}}_{jk}^j = \mathbf{A}_{jk}^j \mathbf{y}_{jk}^j + \mathbf{b}_{jk}^j \Big|_{\mathbf{A}_{jk}^j = \mathbf{A}_{jk,\min}^j, \mathbf{b}_{jk}^j = \mathbf{b}_{jk,\min}^j}. \quad (19)$$

Srednja kvadratna greška pri proceni pomenutog kanalnog odziva data je sa

$$\text{MSE}_{jk}^j = \mathbb{E}\{\|\mathbf{g}_{jk}^j - (\mathbf{A}_{jk}^j \mathbf{y}_{jk}^j + \mathbf{b}_{jk}^j)\|^2\} = \text{tr}(\mathbf{C}_{\tilde{\mathbf{g}}_{jk}^j}^j). \quad (20)$$

Minimizovana srednja kvadratna greška linearnog Bajesovog estimatora za korisnika k iz ćelije j na baznoj stanici j zapisuje se prema

$$\begin{aligned} \text{LMMSE}_{jk}^j &= \mathbb{E}\{\|\mathbf{g}_{jk}^j - (\mathbf{A}_{jk,\min}^j \mathbf{y}_{jk}^j + \mathbf{b}_{jk,\min}^j)\|^2\} \\ &= \text{tr}(\mathbf{C}_{\mathbf{g}_{jk}^j \mathbf{g}_{jk}^j}^j - \mathbf{C}_{\mathbf{g}_{jk}^j \mathbf{y}_{jk}^j}^j \mathbf{C}_{\mathbf{y}_{jk}^j \mathbf{y}_{jk}^j}^{j-1} \mathbf{C}_{\mathbf{y}_{jk}^j \mathbf{g}_{jk}^j}^j). \end{aligned} \quad (21)$$

Prema modelu sistema iz (9), nakon projektovanja prijemne sekvence $\mathbf{Y}_j^p$ na konkretnu pilotsku sekvencu, ϕ_{jk} dobija se

$$\begin{aligned} \mathbf{y}_{jjk}^p &= \mathbf{Y}_j^p \phi_{jk}^* \\ &= \sum_{(l', i') \in \mathcal{P}_{jk}} \sqrt{p_{l' i'}} \tau_p \mathbf{g}_{l' i'}^j + \mathbf{N}_j^p \phi_{jk}^*. \end{aligned} \quad (22)$$

Parametri za estimaciju kanala postaju:

$$\begin{aligned} \mathbf{A}_{jjk,\min}^p &= \mathbf{C}_{\mathbf{g}_{jk}^j \mathbf{y}_{jjk}^p}^j \mathbf{C}_{\mathbf{y}_{jjk}^p \mathbf{y}_{jjk}^p}^{p-1} \Big|_{\mathbf{y}_{jjk}^p = \mathbf{y}_{jjk}^p} \\ \mathbf{b}_{jjk,\min}^p &= \mathbb{E}\{\mathbf{g}_{jk}^j\} - \mathbf{C}_{\mathbf{g}_{jk}^j \mathbf{y}_{jjk}^p}^j \mathbf{C}_{\mathbf{y}_{jjk}^p \mathbf{y}_{jjk}^p}^{p-1} \mathbb{E}\{\mathbf{y}_{jjk}^p\} \Big|_{\mathbf{y}_{jjk}^p = \mathbf{y}_{jjk}^p} \end{aligned} \quad (23)$$

smatrajući da su poznate i kovarijansne matrice:

$$\begin{aligned} \mathbf{C}_{\mathbf{g}_{jk}^j \mathbf{y}_{jjk}^p}^j &= \sqrt{p_{jk} \tau_p} \mathbf{C}_{\mathbf{g}_{jk}^j \mathbf{g}_{jk}^j}^j \\ \mathbf{C}_{\mathbf{y}_{jjk}^p \mathbf{y}_{jjk}^p}^p &= \sum_{\substack{\{(l', i'), \\ (l'', i'')\} \in \mathcal{P}_{jk}}} \sqrt{p_{l' i'} p_{l'' i''}} (\tau_p)^2 \mathbf{C}_{\mathbf{g}_{l' i'}^j \mathbf{g}_{l'' i''}^j} \\ &\quad + \mathbf{C}_{\mathbf{N}_j^p \phi_{jk}^* \mathbf{N}_j^p \phi_{jk}^*}. \end{aligned} \quad (24)$$

4.4. Rezultati istraživanja

U svakoj od ukupno $L = 16$ ćelija postavljena je po jedna bazna stanica. Bazne stanice su opremljene sa identičnim antenskim nizom sa uniformnim rastojanjem $d = \frac{\lambda}{2}$ između $M_j = 64$ antenskih elemenata, $\forall j$. U svakoj ćeliji nalaze se po $K = 4$ fiksna korisnička terminala opremljena sa po jednom antenom. Preostali sistemski parametri preuzeti su iz [3, Sekcija 4.1.3].

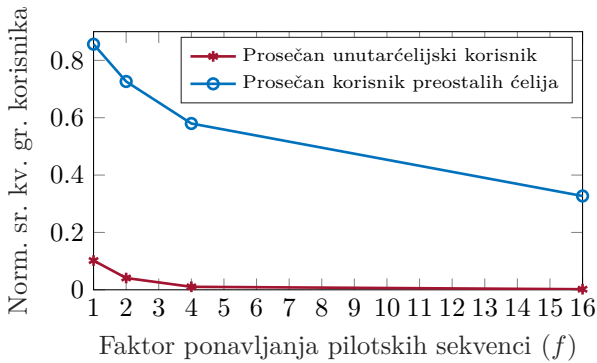
Estimiraju se korisnički kanali između svih kombinacija baznih stanica i korisničkih terminala u skladu sa tim da li se korisnici nalaze u istoj ćeliji j kao i bazna stanica ili u okolnim ćelijama $l, l \neq j$. Prosečna normalizovana srednja kvadratna greška pri proceni impulsnog odziva unutarćelijskih korisnika računa se prema

$$\overline{\text{NMSE}}_{\text{unutarćelijsko}} = \frac{1}{K} \sum_{k=1}^K \sum_{j=1}^L \frac{\text{tr}(C_{\mathbf{g}_{jk}^j \mathbf{g}_{jk}^j})}{\text{tr}(C_{\mathbf{g}_{jk}^j \mathbf{g}_{jk}^j})}. \quad (25)$$

Prosečna normalizovana srednja kvadratna greška pri proceni impulsnog odziva korisnika preostalih ćelija računa se prema

$$\overline{\text{NMSE}}_{\text{preostalih ćelija}} = \frac{1}{K} \sum_{k=1}^K \sum_{j=1}^L \sum_{l=1, l \neq j}^L \frac{\text{tr}(C_{\mathbf{g}_{lk}^j \mathbf{g}_{lk}^j})}{\text{tr}(C_{\mathbf{g}_{lk}^j \mathbf{g}_{lk}^j})}. \quad (26)$$

U oba slučaja se uprosečavanje izvršava po svim korisnicima na nivou svih odgovarajućih ćelija. U tom postupku se vrednost dobijena sumiranjem normalizuje sa brojem korisnika, K . Na slici 4 iscrtavaju se grafici normalizovane srednje kvadratne greške, NMSE_{lk}^j u zavisnosti od faktora ponavljanja pilotskih sekvenci, $f \in \{1, 2, 4, 16\}$.



Slika 4: Zavisnost greške estimacije od faktora ponavljanja pilotskih sekvenci

Ekstremne vrednosti koje f uzima objašnjavaju se na sledeći način:

- Kada je $f = 1$, svaka ćelija ponavlja identični skup pilotskih sekvenci (tako da korisnik k u jednoj od ćelija koristi identičnu sekvencu kao korisnik k u svim ostalim ćelijama, dok je korisnik j jedne ćelije i dalje ortogonalan sa korisnikom k iz svoje i svih ostalih ćelija).
- Kada je $f = 16$, svaka ćelija koristi jedinstven skup pilotskih sekvenci, što znači da su sada čak i korisnici k iz različitih ćelija međusobno ortogonalni. Dakle, bilo koja sekvenca je ortogonalna sa svim ostalim sekvencama gde god se njihovi korisnici nalazili u mreži.

Kada ne postoji pilotska kontaminacija u mreži ($f = 16$), greška estimacije unutarćelijskih korisničkih kanala teži nuli, dok je za korisničke kanale iz ostalih ćelija znatno viša. Ovakav ishod se može objasniti postojanjem uticaja drugih faktora na grešku estimacije, osim faktora ponavljanja pilotskih sekvenci. Ključni preostali faktori su predajna snaga željenog korisničkog signala i njegovo slabljenje na trasi, interferencija koja nije posledica pilotske kontaminacije i šum.

5. ZAKLJUČAK

Uveden je model sistema u uzlaznom smeru prenosa sa tehnikom višestrukog pristupa korisnika zasnovanoj na prostornoj raspodeli. Za Zadof-Ču sekvence ispitane su i ilustrovane autokorelacione i među-korelacione osobine. Za estimaciju nepoznatog kanala korišćen je linearni Bajesov estimator koji minimizuje srednju kvadratnu grešku procene.

U izloženim rezultatima istraživanja za evaluaciju performansi razmatranog estimatora korišćena je normalizovana srednja kvadratna greška. Utvrđeno je da pored kontaminacije pilotskih sekvenci (tj. upotrebe neortogonalnih sekvenci), ukupnoj grešci estimatora doprinose i interferencija i šum. Iscrtane amplitudske karakteristike greške estimacije su opadajuće, pri čemu je pretpostavljeno da su neophodna statistička predznanja potpuno poznata, iako ih je u praktičnim okolnostima neophodno proceniti. U stvarnim sistemima se očekuju više vrednosti greške estimacije korisničkih kanala, u odnosu na prikazane vrednosti u ovom radu.

6. LITERATURA

- [1] G. Del Galdo, "Geometry-Based Channel Modeling for Multi-User MIMO Systems and Applications," doktorska disertacija, jul 2007. Dostupno na: https://www.db-thueringen.de/receive/dbt_mods_00011002
- [2] J. G. Andrews, "A Primer on Zadoff Chu Sequences," 2023. Dostupno na: <https://doi.org/10.48550/arXiv.2211.05702>
- [3] E. Björnson, J. Hoydis i L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Foundations and Trends® in Signal Processing*, vol. 11, no. 3-4, pp. 154–655, 2017. Dostupno na: <http://dx.doi.org/10.1561/20000000093>

Kratka biografija:

Aleksandar Birmančević rođen je u Šapcu 1996. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Telekomunikacioni sistemi odbranio je 2024. god.

Kontakt: abirmancevic@uns.ac.rs

MIKROSERVISNA ARHITEKTURA NA PRIMERU WEB APLIKACIJE ZA PERSONALNE TRENERE**MICROSERVICE ARCHITECTURE APPLIED TO THE PERSONAL TRAINER WEB APPLICATION**

Petar Stošić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – U ovom radu izneta je teorijska osnova o različitim tipovima softverske arhitekture, iznete su ključne karakteristike kao i prednosti i mane svakog od njih. Fokus je bio na mikroservisnoj arhitekturi sistema koja predstavlja najpopularniju, ali i najčešće pogrešno shvaćenu i pogrešno korišćenu arhitekturu.

Ključne reči: Mikroservisna arhitektura, monolitna arhitektura, saga patern, tipovi interprocesne komunikacije.

Abstract – The paper describes microservice architecture, other types of software system architecture, their benefits but also it shows their problems as well. It explains what decisions developers need to make on the way when choosing the right architectural approach of their project. Microservice architecture is most popular nowadays, but it really has some high financial and organizational impacts, which needs to be foreseen.

Keywords: Microservice architecture, monolite architecture, saga pattern, interprocess communication patterns.

1. UVOD

U današnjem dinamičnom svetu informacionih tehnologija, sposobnost brze i efikasne dostave softverskih rešenja postaje sve važnija. U potrazi za boljim performansama, skalabilnošću i olakšanim održavanjem, razvijaju se različite arhitekture sistema [1]. Jedna od njih koja se sve više ističe je mikroservisna arhitektura.

Ova inovativna arhitektura omogućava razdvajanje velikih aplikacija na manje, samostalne i međusobno povezane delove, poznate kao mikroservisi. U poređenju sa tradicionalnom monolitnom arhitekturom, mikroservisna arhitektura donosi brojne prednosti, ali takođe nosi i određene izazove i mane.

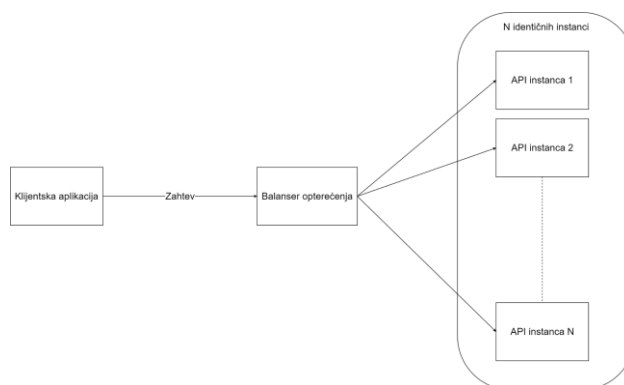
2. MONOLITNA ARHITEKTURA

Jedna od bitnijih karakteristika softvera baziranog na monolitnoj arhitekturi jeste ta da je njegova čitava funkcionalnost obuhvaćena (enkapsulirana) jednom aplikacijom, pa tako njeni moduli ne mogu biti izvršavani odvojeno.

Ovaj tip arhitekture softvera smatra se usko povezanom (eng. tightly-coupled), tako da se čitava logika oko obrade zahteva izvršava u okviru jednog procesa. Korišćenjem osnovnih jezičkih struktura i osobina, moguće je ovakve aplikacije unaprediti u smislu strukture, zavisnosti i performansi.

Ovakve sisteme uz određena ograničenja moguće je horizontalno skalirati. Horizontalno skaliranje ovakvih aplikacija najčešće se implementira korišćenjem servisa za balansiranje sistema (eng. Load balancer-a) kao što prikazuje Slika 1. Ovakav tip aplikacije može poslužiti za kreiranje inicijalne verzije i upoznavanjem kompleksnosti sistema kao i granica između različitih komponenta. Međutim, kada projekat i broj ljudi u timu poraste, ovakav tip arhitekture aplikacije počinje da pokazuje svoje nedostatke.

Slika 1. prikazuje način potencijalnog horizontalnog skaliranja aplikacije bazirane na monolitnoj arhitekturi.



Slika 1. Primer skaliranja monolitne aplikacije

2.1. Prednosti monolitne arhitekture

Monolitna arhitektura ima određene prednosti. Naime, ovakav pristup organizacije softverskog sistema pogodniji je za brzi razvoj aplikacija čiji je glavni cilj brzi izlazak na tržište i na kojima radi manji tim ljudi. Veličina tima je jako bitna jer se čitav kod kod monolitnih aplikacija nalazi na jednom repozitorijumu, i čitava aplikacija izvršava u okviru istog procesa. Da ovakav pristup ne bi krenuo u pravcu toga da se izgubi ikakva arhitektura, jako je bitno usredsrediti se na to da se izoluju komponente sistema. Ukoliko se komponente jasno izdvoje i definišu u okviru monolitne arhitekture, kasnija evolucija ka mikroservisnoj arhitekturi će biti dosta

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kupusinac, red. prof.

olakšana. Još jedna bitna prednost ovog načina organizacije jeste dosta lakši proces (eng. build-ovanja) i (eng. release-ovanja) aplikacije. Naime potrebno je kreirati samo jedan paket i odraditi (eng. deploy).

2.2 Problemi monolitne arhitekture

Osim prednosti, ovaj tip arhitekture sistema nosi sa sobom i neke probleme.

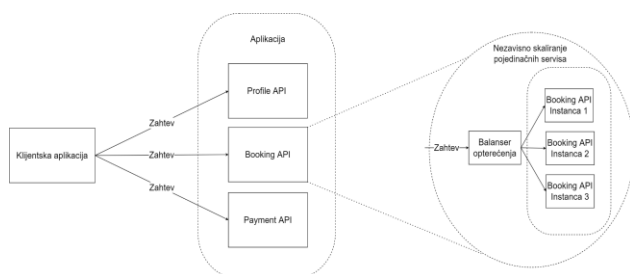
- Poteškoće sa skaliranjem
- Sporiji razvoj i isporuka kada su u pitanju veliki razvojni timovi
- Tehnička raznolikost je veoma ograničena
- Problemi održavanje i kompleksnost projekta vremenom postaju sve veći.
- Tehnički dug (eng. Tech debt) vremenom postaje sve veći.

3. MIKROSERVISNA ARHITEKTURA

Mikroservisna arhitektura predstavlja pristup kreiranja softverskog rešenja koje se sastoji iz više malih servisa koji svojim odgovornostima upotpunjavaju ukupnu funkcionalnu celinu aplikacije. Svaki od tih servisa, trebao bi biti nezavistan i izolovan proces, koji komunikaciju sa ostalim servisima ostvaruje nekim od mehanizama za među procesnu komunikaciju. Jedan od najčešće korišćenih mehanizama je HTTP protokol. Mikroservisi se kreiraju tako da predstavljaju neku od odvojenih celina koje grade određeni poslovni sistem. Idealno je da budu što je moguće više samostalni i da imaju što manje zavisnosti ka ostalim servisima. Najčešće se za ovakve sisteme konfigurišu automatizovani procesi za (eng. build) i (eng. release) fazu. Na taj način omogućava se pouzdan sistem razvoja, i sprečava se uticaj ljudskog faktora, gde bi zbog većeg broja servisa bilo gotovo nemoguće posao odraditi bez greške.

Dekompozicijom monolita na mikro servise, postiže se i veća otpornost sistema na greške. Ukoliko jedan od servisa ima problem u određenom trenutku, ostatak sistema (koji ne zavisi od problematičnog dela) može neometano da nastavi sa radom. U monolitu, ukoliko bi postojao sličan problem, čitav sistem bi bio ugrožen.

Slika 2. prikazuje primer sistema zasnovanog na mikroservisnoj arhitekturi, kao i mogućnosti nezavisnog skaliranja servisa koji ima veće opterećenje od ostalih.



Slika 2. Sistem zasnovan na mikroservisnoj arhitekturi

3.1 Prednosti mikroservisne arhitekture

Mogu se izdvojiti nekoliko ključnih prednosti koje karakterišu mikroservisnu arhitekturu sistema:

- Omogućava implementiranje (eng. CI/CD)-a velikih i kompleksnih aplikacija
- Servisi su najčešće mali i lako održivi

- Servisi se mogu razvijati nezavisno jedan od drugog
- Servisi se mogu skalirati nezavisno
- Omogućava autonomiju timova
- Omogućava eksperimentisanje i prihvatanje novih tehnologija
- Karakteriše je bolja tolerancija grešaka

3.2 Problemi mikroservisne arhitekture

I ova arhitektura poseduje više ozbiljnih problema koje treba pravilno uočiti, identifikovati i rešiti. Veći deo tih problema je takav da je moguće rešiti ga, ali generalno prilikom projektovanja ovakvih sistema treba biti veoma oprezan i upoznat sa domenom problema koji se rešava. Neki od ključnih problema ove arhitekture su:

- Pronalaženje pravog skupa servisa
- Distribuirani sistemi su dosta zahtevni i kompleksni i čine sve aspekte razvoja softvera dosta težim
- (eng. Deploy-ovanje) novih stvari koje uključuju više od jednog servisa je dosta zahtevno i traži dobru koordinaciju i planiranje
- Donošenje odluke kada preći na mikroservisnu arhitekturu.

4. PROJEKTOVANJE SISTEMA METODOM DEKOMPOZICIJE

S obzirom da aplikacije postoje kako bi izvršavale zahteve, prvi korak bi upravo trebao biti identifikacija zahteva na apstraktnom nivou, kao i njihova formalizacija u okviru (eng. User story-a). Ove zahteve možemo definisati u vidu sistemskih operacija. Sistemska operacija je apstrakcija konkretnog zahteva i kao takva omogućava nam da na ovom nivou nemamo previše nepotrebnih detalja koji će doći kasnije u procesu konkretizacije. Sistemska operacija može se podeliti na dva tipa:

- prvi tip su operacije koje menjaju stanje sistema, nazivaju se komande (eng. Commands)
- drugi tip operacije onaj koji ne menja stanje sistema već potražuje stanje sistema, nazivaju se upiti (eng. Queries).

Ponašanje svake komande je definisano na nivou apstraktnih domanskih modela, koji su takođe preuzeti iz zahteva. Sistemske operacije su na taj način svojevrsni arhitektonski scenariji, koji ilustruju kako servisi međusobno komuniciraju. Nakon identifikacije zahteva koje sistem mora da podrži, sledeći korak je identifikacija servisa. U ovom trenutku postoji više načina na koji se tom problemu može pristupiti a dva najčešća su:

- Na osnovu domenskog modela (eng. Domain Driven) na manje pod domene
- Na osnovu stvari koje čine biznis i koje on može da podrži (eng. Business capabilities)

Sušтина oba pristupa je da se na kraju servisi organizuju na osnovu koncepata koji postoje u samom biznis domenu, a ne na osnovu tehničkih koncepata. Treći korak u definisanju arhitekture aplikacije je određivanje i definisanje API -a svakog od identifikovanih servisa. To se postiže tako što se prethodno identifikovane sistemske operacije u prvom koraku dodeljuju identifikovanim servisima u okviru drugog koraka. Neke od operacija servisi mogu da izvrše samostalno, međutim neke

kompleksnije operacije zahtevaju koordinaciju nekada i više od dva servisa.

5. INTERPROCESNA KOMUNIKACIJA

Zbog dekomponovanja monolitnog sistema, u fizički odvojene servise, komunikacija na nivou jezičkih konstrukcija nije više dostupna. Često servisi nisu ni napisani korišćenjem istih programskih jezika niti softverskih paradigmi tako da je za komunikaciju među ovim servisima potreban drugi mehanizam. Ovakvi mehanizmi se nazivaju Interprocesna komunikacija (eng. Interprocess communication) IPC. Postoji više tipova IPC-a, a glavna podela se može izvršiti na osnovu prirode same komunikacije [2] na:

- Sinhronu (komunikaciju zasnovanu na zahtevima i odgovorima)
- Asinhronu (komunikaciju zasnovanu na porukama)

IPC je, kao i druge odluke vezane za arhitekturu sistema, veoma kompleksna i zahtevaju dobru analizu konkretnog problema. Izbor načina komunikacije određuje ponašanje sistema, njegovu dostupnost kao performanse sistema. Postoje različita tehnološka rešenja i konkretizacije za sinhroni i asinhroni tip komunikacije, koja se najčešće razlikuju u protokolima i formatima poruka koje se razmenjuju. Postoje protokoli koji su ljudski čitljivi, a postoje i oni kod kojih se podaci prenose u binarnom formatu. Koji god od mehanizama da se odabere, jako je bitno prvo definisati API servisa koristeći jezik za definisanje interfejsa nezavisno od konkretnih implementacija (eng. interface definition language) IDL. Način na koji se API specificira, zavisi od toga koji IPC mehanizam odaberemo. Ukoliko je u pitanju komunikacija zasnovana na porukama, API definicija se sastoji od specifikacije kanala, tipa poruke, i formata poruke. Ukoliko se koristi HTTP protokol, API definicija se sastoji od specifikacije URL-a, naziva HTTP metode, formata zahteva i formata odgovora.

6. TRANSAKCIJNA LOGIKA

Transakcije su esencijalni činilac svake poslovne aplikacije. Bez transakcija ne bi bilo moguće garantovati konzistentnost podataka. Transakcija predstavlja nedeljivi skup operacija koje je na nivou softverskog sistema potrebno izvršiti kako bi se jedan korisnički zahtev uspešno obradio. Kod sistema zasnovanih na monolitnoj arhitekturi koji koriste jednu instancu baze podataka za rad sa transakcijama koriste se postojeći mehanizmi koje najčešće poseduje sama baza podataka.

Problem nastaje kada je za obradu zahteva potrebno orkestrirano ponašanje više od jednog servisa u mikroservisnoj arhitekturi.

Može se desiti da jedan servis sačuva podatke u svoju bazu, dok sledeći servis iz nekog razloga nije u mogućnosti da uspešno sačuva podatke. Na taj način konzistencija podataka sistema je narušena jer imamo parcijalno sačuvano stanje. Postoje dva načina da se problem transakcija u mikroservisnim sistemima reše:

- Distribuirane transakcije
- Saga šablon (eng. Saga pattern)

Distribuirane transakcije se najčešće implementiraju korišćenjem standarda (eng. X/Open Distributed

Transaction Processing (DTP) Model) (X/Open XA). X/Open XA koristi algoritam dvostruke potvrde (eng. two-phase commit), da bi osigurao da su svi učesnici u transakciji potvrdili ili poništili izmene. Da bi se koristio mehanizam distribuiranih transakcija u sistemu, sve komponente koje se koriste moraju biti implementirane na način da podržavaju X/Open XA standard.

Saga pattern [3] se koristi kako bi se održala konzistentnost podataka u mikroservisnoj arhitekturi, bez korišćenja prethodno pomenutih distribuiranih transakcija. Prilikom korišćenja ovog patterna, bitno je da se definiše nova saga za svaku komandu u sistemu koja mora da izmeni podatke u više od jednog servisa. Saga predstavlja niz lokalnih transakcija. Svaka od lokalnih transakcija na nivou jednog servisa garantuje atomičnost izvršavanja ACID, odnosno nedeljivost operacije. Svaki korak u okviru izvršavanja jedne sage, nakon završetka izvršavanja poziva prosleđivanjem odgovarajuće asinhronne poruke sledeći korak. Za razmenu poruka koriste se asinhroni mehanizmi, pa je tako mala verovatnoća da će neka poruka biti izgubljena i neobrađena. Međutim kod saga patterna postoji problem vraćanja na prethodno stanje ukoliko jedan od koraka sage ne bude validan i ne može da se izvrši sa pozitivnim ishodom. Dok je kod sistema koji imaju ACID transakcije vraćanje na prethodno stanje (eng. roll-back) dostupan na nivou samog (eng. framework)-a, kod sage se za to mora kreirati kompenzujuća operacija. Dva osnovna tipa saga šablona su distribuirana koreografija [4] i orkestrator [5].

7. APLIKACIJA ZA PERSONALNE TRENERE

Cilj ovog rada je razvoj rešenja koje će biti korišćeno od strane registrovanih klijenata (ličnih trenera) kao i od strane korisnika usluga konkretnih trenera. Prilikom izrade ovog aplikativnog rešenja, u obzir su uzete sve prethodno navedene metodologije modelovanja softverske arhitekture. Korišćena arhitekture serverskog dela sistema je mikroservisna arhitektura.

7.1. Korišćene tehnologije

Aplikativno rešenje sačinjeno je od infrastrukturnog, servisnog i dela za interakciju sa korisnicima. Fokus ovog rada je na infrastrukturnom i servisnom delu. Za opipšivanje i definisanje infrastrukture korišćen je (eng. Terraform) programski alat. Za razvoj samog aplikativnog rešenja korišćen je programski jezik C# i .NET7 programska platforma. Za čuvanje podataka korišćena su dva tipa prostora za skladištenje, zavisno od potrebe. Za čuvanje strukturiranih podataka, korišćena je SQL baza, dok je za čuvanje fajlova, i drugih nestruktuiranih informacija korišćen (eng. Azure Blob Storage). Za razmenu poruka u komunikaciji servisa korišćen je (eng. Azure Service Bus) koji omogućava veoma pouzdan način razmene poruka, i podržava oba modela razmene poruka, (eng. one to one) i (eng. one to many). Za skrivanje arhitekture sistema od klijentskih aplikacija korišćen je (eng. API management) [6]. Razvojna okruženja kojima su korišćena su:

- Visual Studio Professional 2022
- Visual Studio Code
- SQL Management Studio
- Azure Data Explorer

7.2. Opis rešenja problema

Polazna tačka prilikom rešavanja ovog problema bila je modelovanje zahteva odnosno njihovo formalno prikupljanje. Ovom prilikom identifikovane su dve osnovne, kao i jedna administrativna uloga korisnika u sistemu. Za svaku od navedenih uloga, identifikovan je konačan skup operacija koje mogu izvoditi u sistemu.

Identifikovane uloge u sistemu:

- Administrator
- Klijent
- Trener

Za svaku od navedenih uloga identifikovane su operacija u sistemu koje konkretna uloga može da izvrši. Nakon što smo pažljivom analizom došli do skupa operacija, obavljeno je grupisanje operacija na osnovu njihove koherentnosti u domenskom smislu. Identifikovani su sledeći servisi:

- (eng. User Management Service)
- (eng. Membership Service)
- (eng. Payment Service)
- (eng. Feed Service)
- (eng. Training Data Service)
- (eng. Training Material Service)
- (eng. Slot Service)
- (eng. Booking Service)
- (eng. Messages Service)
- (eng. Notification Service)

Zavisno od problema koji svaki od ovih servisa rešava definisane su strukture, kao i način čuvanja podataka. Neki od servisa podatke čuvaju u Azure blob storage-u, u formatu manje striktnosti. Dok neki od servisa koriste SQL baze podataka.

8. ZAKLJUČAK

Odabirom mikroservisne arhitekture, veoma je povećana kompleksnost sistema. Sam razvoj je bio u velikoj meri otežan, samom količinom aplikacija, njihovom konfiguracijom, procesom deploy-ovanja kao i povezivanjem komunikacije među servisima. Za veličinu tima koja je mala, definitivno bi trebalo dobro razmisliti da li je inicijalno mikroservisna arhitektura prava opcija. Vreme razvoja bi bilo značajno kraće kada bi ova aplikacija bila razvijena u monolitnoj arhitekturi i kada bi se koristila centralizovana baza podataka. Takođe treba napomenuti da je cena izvršavanja ovakvog sistema na Cloud-u značajno veća od monolitnog sistema sa istim skupom funkcionalnosti.

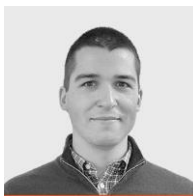
Količina App Service-a, SQL baza, Blob Storage-a u značajnoj meri doprinose tome da cena ovakvog sistema bude visoka. Ukoliko bi tim bio veći, to bi dodatno povećalo potrebu za postojanjem procesa razvoja, a samim tim i troškova kompanije koja bi takav sistem razvijala.

Zaključeno je da pored toga što mikroservisna arhitektura ima ogroman skup prednosti koje su nabrojane u prethodnom delu rada, poseduje i određene probleme koji nisu toliko tehničke prirode, već se tiču finansijskih mogućnosti naručioca sistema, veličine tima, ali i stepena zrelosti proizvoda.

9. LITERATURA

- [1] <https://www.synopsys.com/glossary/what-is-software-architecture.html> (pristupljeno u decembru 2023.)
- [2] <https://learn.microsoft.com/en-us/azure/architecture/guide/technology-choices/messaging> (pristupljeno u decembru 2023.)
- [3] <https://learn.microsoft.com/en-us/azure/architecture/reference-architectures/saga/saga> (pristupljeno u decembru 2023.)
- [4] <https://learn.microsoft.com/en-us/azure/architecture/patterns/choreography> (pristupljeno u decembru 2023.)
- [5] <https://learn.microsoft.com/en-us/azure/architecture/reference-architectures/saga/saga#orchestration> (pristupljeno u decembru 2023.)
- [6] <https://learn.microsoft.com/en-us/azure/api-management/api-management-key-concepts> (pristupljeno u decembru 2023.)

Kratka biografija:



Petar Stošić rođen je u Nišu 1993. godine. Osnovne akademske studije na Fakultetu tehničkih nauka Univerziteta u Novom Sadu upisao je 2012. godine. Diplomirao je 2016. godine i iste godine upisuje master akademske studije.

Kontakt: petar.stosic993@gmail.com

АЛАТ ЗА ДЕКРИПЦИЈУ ZIP АРХИВА ПРИМЕНОМ NVIDIA CUDA ТЕХНОЛОГИЈЕ ZIP ARCHIVE PASSWORD RECOVERY TOOL USING NVIDIA CUDA TECHNOLOGY

Лука Курељушић, *Факултет техничких наука, Нови Сад*

Област – ЕЛЕКТРОТЕХНИЧКО И РАЧУНАРСКО ИНЖЕЊЕРСТВО

Кратак садржај – У овом раду описан је развој алата за опоравак лозинке ZIP архива WinZip формата, енкриптованих помоћу AES алгоритма. Централна идеја алата обухвата паралелизацију напада грубом силом на ZIP архиву употребом једне или више графичких картица. Структура предложеног алата је модуларна и састоји се од фронтенд дела који обавља поступке парсирања архиве и даљег процесирања улазних информација и бекенд дела, имплементираног помоћу NVIDIA CUDA технологије који се извршава на графичкој картици и обухвата паралелизацију PBKDF2 алгоритма деривације кључа и накнадне процесе верификације добијеног кључа над улазним подацима. Алат остварује задовољавајуће перформансе за нумеричке и краће алфанумеричке шифре, док присуство специјалних карактера значајно отежава успех алата.

Кључне речи: ZIP архива, декрипција, опоравак лозинке, паралелизација, CUDA

Abstract – This paper presents the development of a password recovery tool for WinZip format ZIP archives encrypted using the AES algorithm. The central idea of the tool involves parallelizing a brute-force attack on a ZIP archive using one or more graphics cards. The structure of the proposed tool is modular and consists of a frontend part that performs archive parsing and further processing of input information, and a backend part, implemented using NVIDIA CUDA technology, which is executed on a graphics card and includes the parallelization of the PBKDF2 key derivation algorithm and the subsequent processes of verification of the obtained key over input data. Tool achieves satisfactory performance for numeric and shorter alphanumeric codes, while the presence of special characters significantly hinders the success of the tool.

Keywords: ZIP archive, decryption, password recovery, parallelization, CUDA

1. УВОД

У ери у којој безбедност података и приватност постају најзначајнији аспекти информационих система, употреба лозинком заштићених архива, тиме и ZIP датотека постаје све чешћа.

НАПОМЕНА:

Овај рад је проистекао из мастер рада чији ментор је био др Стеван Гостојић, ред. проф.

Мотивација за развој алата за опоравак лозинке ZIP датотеке је двојака. Са једне стране алат налази своју употребну вредност у практичним изазовима са којима се суочавају појединци и организације када изгубе приступ архивама заштићеним лозинком. Са друге стране, идеја алата јесте и допринос пољу дигиталне безбедности, где се алат може употребити за испитивања безбедности делова информационих система који се односе на рад са ZIP архивама.

За потребе рада развијен је алат базиран на NVIDIA CUDA технологији, која представља платформу за имплементацију концепта паралелизације заснованог на употреби графичких картица. Сам поступак разбијања шифре подразумева итеративни процес генерисања и потоње провере кандидата шифре методом грубе силе над претходно (кориснички) дефинисаним простором претраге који представља рестрикцију над скупом ASCII карактера.

У овом раду најпре ће бити извршен преглед стања у области уз осврт на актуелна најсавременија решења отвореног кода која се баве датом проблематиком. Потом ће бити описана предложена архитектура и спецификација захтева алата након којих ће бити представљени најзначајнији имплементациони детаљи и примењене технике оптимизације. Потом ће бити представљени и дискутовани резултати мерења перформанси алата у односу на друге, савремене алате. У последњем поглављу рада биће представљени закључци рада уз коментар о потенцијалним правцима даљег развоја.

2. ПРЕГЛЕД СТАЊА У ОБЛАСТИ

У одабиру репрезентативних савремених алата везаних за област разбијања лозинке ZIP архива може се увести подела у погледу методологије приступа решавању проблема напада грубом силом, где се различита решења могу сврстати у процесорске алате (што представља прву групу) и алате базиране на графичким картицама. Додатно, ови алати могу подржавати употребу у cloud окружењу за постизање масовне паралелизације. Сходно томе, као два репрезентативна алата узети су алати отвореног кода *John The Ripper* као превасходно процесорски алат и *Hashcat* као алат базиран на графичким картицама.

2.1. John The Ripper

John The Ripper [1] нуди различите прилагодљиве методе напада од којих је напад грубом силом од највећег значаја за овај рад, иако су у погледу рада са ZIP архивама свакако интересантни и хибридни и

mask напади. *John The Ripper* се у стандардним подешавањима увек ослања на *CPU*, док се додатним специфицирањем може дефинисати употреба *GPU*, за које је потребна инсталација *OpenCL*.

Једна од значајних предности *John The Ripper* алата јесте модул за екстракцију битних информација и добијање хеша *ZIP* датотеке - *zip2John*. Овако добијене информације (које у ширем смислу могу да се назову хеш иако то суштински нису) служе као улаз основном модулу *John The Ripper*. *Hashcat* као очекивани улаз такође очекује хеш специфициран по нешто измењеном [2] излазном формату *zip2John* модула, те је овај модул практично неопходан и за употребу *Hashcat*, пошто сличан алат не долази у склопу *Hashcat-a*.

2.2. Hashcat

Hashcat [3], као алат који се веома често користи у области дигиталне безбедности, долази као стандардан уз инсталацију неких оперативних система намењених за употребу у овој области, као што је *Kali Linux*.

Иако старије верзије подржавају и процесорски режим рада, савремене верзије *Hashcat* алата подржавају извршавање искључиво на графичким картицама. Као и у претходном случају, од значаја за овај рад је режим напада грубом силом.

Битно је напоменути да *Hashcat* представља алат за разбијање хеша, те као такав нема могућност директног рада са *ZIP* датотекама, већ се улазни параметри (хеш), потребни за добијање шифре морају прибавити ручно или употребом неког екстерног алата (најчешће поменути *zip2John*).

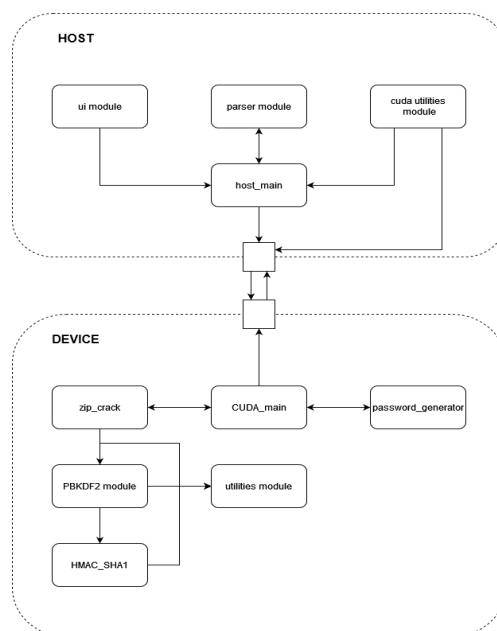
3. СПЕЦИФИКАЦИЈА

Апликација се према начину извршавања кода може поделити на фронтенд део, који се извршава на процесору и бекенд део, који обухвата функционалности које се извршавају на једној или више графичких картица. На слици 1 приказана је шема предложене архитектуре алата.

Најзначајнији модули фронтенда јесу *host_main* и *parser* модули. Модул *host_main* представља централно место апликације и задужен је за контролу тока програма, регистрацију парсер и бекенд модула, обраду информација добијених из парсера и креирања одговарајућег окружења на графичкој картици (трансфер података, покретање кернел функција итд.).

Осим тога, ослањајући се на функционалности *UI* и *cuda_utilities* модула, главни модул врши креирање простора претраге шифре на основу улазних података *UI* модула. Парсер модул обухвата функционалности детекције и провере интегритета *ZIP* архиве, поступке парсирања и екстракције неопходних информација.

У току овог поступка врши се секвенца провера у односу на верзију архиве, тип архиве, тип енкрипције и слично, путем којих се одбацује свака *ZIP* архива која, иако валидног формата, евентуално није подржана од стране алата.



Слика 1 - Шематски приказ структуре алата

Бекенд део ослања се на *CUDA_main* као централни модул који посредује између свих осталих елемената бекенда и фронт дела. Модул *password_generator* има улогу постављања иницијалног речника (простора претраге) и креирања и доделе шифре свакој нити. Поред тога, садржи и функцију која као аргумент прима сваку неуспелу шифру на основу које враћа логички следећу чиме је омогућен итеративни рад програма.

Други значајни скуп функционалности обухвата *zip_crack* модул, који енкапсулира све функционалности везане за бекенд рад са *ZIP* архивама, те се као такав може посматрати као плагин за *ZIP*. Овај модул садржи подмодуле за деривацију кључа помоћу *PBKDF2* [4] алгоритма заснованог на *HMAC-SHA1* [5][6] псеудонасумичној функцији. *PBKDF2* ради у конфигурацији од 1000 итерација са дужином добијеног кључа од 528 бита.

4. ИМПЛЕМЕНТАЦИЈА

Како би се дефинисала оптимална конфигурација на графичкој картици (број нити, оптимална употреба регистара и сл.) најпре се на основу улазних аргумената програма врши рачунање укупног простора претраге и евентуално даље партиционисање истог (у случају извршавања алата на више графичких картица или рачунара).

Како би се подржао рад на више уређаја, алату је могуће проследити опционе аргументе који дефинишу број машина које врше обраду *ZIP* архиве и индексирајуће машине. Поред поменутих опционих аргумената, алат региструје све доступне *NVIDIA* графичке картице у оквиру дате машине. Као обавезни улазни аргументи специфицирају се максимална дужина претражене шифре као и речник над којим се претрага врши. Могуће је одабрати један од четири речника: нумеричка шифра, алфанумеричка (само мала слова), алфанумеричка шифра и комплетан речник (алфанумеричка са специјалним карактерима).

Претходно наведене информације користе се за дефинисање укупног простора претраге шифре, даље партиционисање истог и доделу дела посла свакој од регистрованих графичких картица на свим рачунарима. Формула 1 приказује начин рачунања укупног броја шифри које се испитују:

$$S = \sum_{i=1}^n x^i = \frac{x * x^{n-1}}{x - 1}$$

Формула 1– *Укупан простор претраге*

У претходној формули x представља дужину одабраног речника, а n максималну дужину шифре.

За сваку дефинисану графичку картицу отвара се *CUDA context*, на коме се специфицира покретање одређеног броја *stream*-ова. Формула 2 приказује начин на који се рачуна број *stream*-ова:

$$2^{\min(\lfloor \log_{10} k \rfloor - 6, 6)}$$

Формула 2 - *Рачунање броја stream-ова*

Коефицијент k представља целобројну вредност добијену као количник укупног простора претраге са бројем рачунара и графичких картица међу којима се врши подела посла и трију константи: број блокова – 6, број нити по блоку – 512 и основни број *stream*-ова – 26. Важно је напоменути да се ова формула примењује само за коефицијент већи од 10 милиона, док се у осталим случајевима примењује вредност 26. Након дефинисања броја *stream*-ова врши се рачунање „укупног посла“ (броја шифри које ће бити испитане) по нити графичке картице (уређаја), те се извршавање пребацује на уређај.

По покретању кернел функције уређаја, свакој нити додељује се јединствени идентификатор, који се користи као механизам за униформно мапирање простора претраге додељеног текућем уређају на конкретне нити. Поступак функционише по принципу креирања класа еквиваленције поделом додељеног простора претраге на укупан број нити, те доделе прве вредности сваке класе еквиваленције одговарајућој нити.

Овакво креирање класа еквиваленције резултује дуалном природом механизма претраге. Са једне стране, обезбеђује се униформно испитивање простора претраге *додељеног уређају* у оквиру текућег *stream*-а, док је, посматрано са глобалног аспекта, претрага ипак локализована, односно секвенцијалне природе на нивоу *stream*-а. Локална природа претраге на нивоу *stream*-а омогућава кориснику који има додатне информације о шифри (на пример познавање једног или више почетних карактера) задавање улазних аргумената који резултују таквим партиционисањем укупног простора претраге који обезбеђује да простор претраге *додељен уређају* садржи шифре које почињу датим карактерима, чиме се простор претраге фактички сужава.

Додатну предност дефинисања поменутих класа еквиваленција представља чињеница да је поступак

преласка на наредну шифру (следећа итерација) рачунарски веома једноставан. Наиме, за сваку нит предодређен је број шифри које ће бити испитане, па се поступак преласка на наредну шифру своди на једноставну операцију инкрементирања вредности последњег карактера текуће шифре. Ово је имплементирано једноставном употребом низа чија структура прати ASCII табелу, са померајем у лево. Инкрементирање шифре постиже се читањем вредности овог низа са позиције која одговара целобројној вредности последњег карактера текуће шифре, те заменом старе вредности новом.

Даљи процес рада алата своди се на употребу поменуте *PBKDF2* функције у конфигурацији од 1000 итерација са дужином добијеног кључа од 528 бита, када се по добијању кључа врши провера последњих 2 бајта у односу на верификациону вредност добијену парсирањем архиве. Дужина верификационе вредности узрокује добијање лажно позитивних резултата у $1/2^{16}$ случајева, што представља 1 лажно позитиван резултат на сваких 65,536 шифри [7]. Из овог разлога, шифре за које прођу иницијалну проверу верификационе вредности врши се провера аутентификационе вредности (такође добијене парсирањем архиве). Успешна провера аутентификационе вредности подразумева употребу добијеног хеша као кључа за добијање *hmac-sha1* хеша над енкриптованим подацима архиве. Првих 10 бајтова новодобијеног хеша представља аутентификациону вредност, која би у случају исправног хеша шифре, требало да се подудара са вредношћу прослеђеном из парсера. У случају подударања ове вредности, контрола извршавања се враћа процесору, где се добијена шифра користи за декрипцију архиве и проверава њена исправност.

5. РЕЗУЛТАТИ

Сва мерења извршена су на машини следеће конфигурације:

- Процесор *intel i7-8750H* на *4.00 GHz*
- Графичка *NVIDIA GeForce GTX 1050Ti*
- Интегрисана графичка (*hashcat*) *Intel UHD 630*
- *RAM 16GB*

John the ripper алат у *CPU* конфигурацији није остварио приближно задовољавајуће перформансе (у односу на *GPU* решења), те је изузет из табеларног приказа. Наиме, у итеративном моду (груба сила), овај алат не даје брзе резултате већ за шифре дужине 4 карактера. Тестиране су шифре: 1234 – 1s; 2209 – 15s; 9909 – 785s; 1x1k – процес прекинут након сат времена рада.

Може се констатовати да алат може успешно да се носи са релативно кратким искључиво нумеричким шифрама, док са повећањем простора претраге осталим карактерима цео процес постаје превише дуг. Табела 1 даје приказ мерења *hashcat* и алата имплементираних у овом раду:

<i>Hashcat</i>							
Тип/дужина	6	7	8	9	10	11	12
нумеричка шиф	2s	16s	119s	19m	~3h10m	~1d7h	~13d
комплетан скуп	~9d12h	~2g180d	-	-	-	-	-
<i>Алат</i>							
нумеричка шиф	7.8s	79.2s	13m15s	2h12m	~22h	~9d	~90d
комплетан скуп	~58d	~15g	-	-	-	-	-
нум и мала слова	1h19m	~1d22h	~70d	-	-	-	-
алфанумеричка	~2d12h	~112d	-	-	-	-	-

Табела 1- Приказ мерења перформанси

Важно је напоменути да *Hashcat* користи и интегрисану поред *NVIDIA GPU*.

У табели је извршено поређење шифре комплетног скупа и нумеричке шифре, јер *Hashcat* алат не нуди директно друге две комбинације за речник. Последња два реда дају мерења и за преостала два речника.

Сва мерења до 2 сата трајања су извршена, док су остала (означена префиксом ~) дата као приближна процена.

Основни и кључни закључак који се може донети на основу наведених вредности јесте да приступ нападом грубом силом никако није ефикасан у већини случајева, осим код веома кратких или строго нумеричких шифри. Експоненцијални раст трајања процеса практично гарантује неуспех за сваку шифру дужине преко 8 која садржи комбинацију алфанумеричких карактера и специјалних карактера.

6. ЗАКЉУЧАК

У овом раду представљен је предлог архитектуре и имплементације алата за опоравак лозинке *ZIP* архиве енкриповане помоћу *AES-256* алгоритма по *WinZip* стандарду. Имплементација алата заснива се на паралелизацији хеш алгоритма употребом графичке картице, односно *NVIDIA CUDA* технологије.

Оно што овај алат издваја од осталих (што подразумева и поменуте *Hashcat* и *John the ripper* алате) јесте независност од употребе екстерних алата за извршавање процеса. Наиме и *Hashcat* и *John the ripper* ослањају се на *zip2John* модул *John the ripper* алата, који је претходно неопходно прибавити и извршити да би се обезбедио адекватан улаз у алат. Са друге стране, алат предложен у овом раду представља комплетно решење са једноставним корисничким интерфејсом, који од корисника захтева минималне улазне информације.

Као негативне стране алата, свакако се истиче ослањање искључиво на метод грубе силе, који сам по себи није ефикасан и дефинитивно није користан за сложене и дуге шифре.

Ток даљег развоја би могао да се одвија у правцу имплементације ефикаснијих метода генерисања кандидата шифре (у односу на метод грубе силе). Затим додатно унапређење корисничког интерфејса којим би се кориснику обезбедила додатна контрола

над комплетним процесом, у циљу смањења простора претраге шифре, као и подршка употребе интегрисаних и осталих не *CUDA* графичких картица.

7. ЛИТЕРАТУРА

- [1] <https://github.com/openwall/john> (приступљено у новембру 2023.)
- [2] <https://github.com/hashcat/hashcat/issues/2186> (приступљено у новембру 2023.)
- [3] <https://hashcat.net/hashcat> (приступљено у новембру 2023.)
- [4] <http://www.faqs.org/rfcs/rfc2898.html> (приступљено у новембру 2023.)
- [5] <https://datatracker.ietf.org/doc/html/rfc2104> (приступљено у новембру 2023.)
- [6] <https://www.rfc-editor.org/rfc/rfc3174> (приступљено у новембру 2023.)
- [7] <https://www.winzip.com/en/support/aes-encryption/#key-generation> (приступљено у октобру 2023.)

Кратка биографија:



Лука Курељушић рођен је 1999. године у Суботици, Република Србија. Основне академске студије завршио је 2022. године на Факултету техничких наука, на ком брани и мастер рад 2024. године из области Електротехнике и рачунарства смер софтверско инжењерство и инфо-рмационе технологије.

контакт:
lkureljusic@ymail.com

ИМПЛЕМЕНТАЦИЈА ПОДРШКЕ ЗА ПРЕНОС ДАТОТЕКА ПОМОЋУ CANOPEN И HTTP ПРОТОКОЛА У ИНДУСТРИЈСКОМ ИНТЕРНЕТУ СТВАРИ**IMPLEMENTATION OF SUPPORT FOR DATA TRANSFER USING CANOPEN AND HTTP PROTOCOL IN INDUSTRIAL INTERNET OF THINGS**Саша Бојанић, Иван Каштелан, *Факултет техничких наука, Нови Сад***Област – РАЧУНАРСТВО И АУТОМАТИКА**

Кратак садржај – У овом раду описан је приједлог рјешења за пренос датотеке са дијагностичким подацима у оквиру једног IoT система који укључује машину са посредничким уређајем и удаљени сервер. Пренос унутар машине одвија се преко Controller Area Network (CAN) магистрала, односно помоћу CANOpen протокола, док се пренос до удаљеног сервера одвија помоћу Hypertext Transfer Protocol (HTTP) протокола. Приликом преноса датотеке до удаљеног сервера уочен је недостатак уколико је временско ограничење за пренос фиксно. Да би се недостатак превазишао, имплементирано је израчунавање временског ограничења на основу ширине пропусног опсега и величине датотеке. Рад укључује и анализу резултата израчунавања временског ограничења.

Кључне речи: *Индустријски интернет ствари, CANOpen, HTTP, посреднички уређај, Fog computing*

Abstract – This paper presents an implementation of the support for data transfer with diagnostic data within an IoT system that includes the fog device and cloud. Data transfer within the device is performed using the CAN bus, following the CANOpen protocol, while the data transfer to the cloud follows the HTTP protocol. During the data transfer, a deficiency is noticed if the time constraint is fixed. To overcome this deficiency, the calculation of the time constraint based on the bandwidth and file size is implemented. The paper includes the analysis of the time constraint calculation results.

Keywords: *Industrial internet of things, CANOpen, HTTP, fog device, fog computing*

1. УВОД

Развојем и усавршавањем рачунарских технологија, као и комуникације путем интернета, створени су услови за интеграцију разних електронских уређаја са циљем побољшања квалитета живота, унапређења сигурности, као и уштеде енергије. С тим у вези настао је појам под именом „Интернет ствари“ (енгл. *Internet of Things – IoT*). Интернет ствари представља велики број умрежених механичких, електричних и виртуелних уређаја који међусобно комуницирају и на тај начин формирају аутоматизоване системе

НАПОМЕНА:

Овај рад проистекао је из мастер рада чије ментор је био др Иван Каштелан, ванр. проф.

којима је потребна минимална или никаква интеракција са људима. Интернет ствари је пронашао широку примјену у индустрији, те се тако развила и засебна грана – индустријски интернет ствари. Уређаји у оквиру индустријског интернета ствари умрежени су у оквиру фабрика или машина. Интернет ствари у оваквим машинама нашао је примјену у виду сензора и актуатора, као и посредничких уређаја и сервиса облака који су задужени за прикупљање, праћење и анализу података из саме машине.

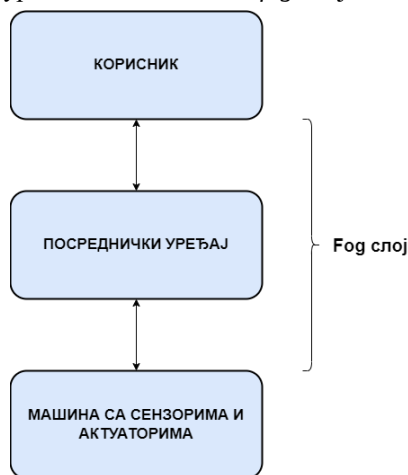
У овом раду је описана имплементација подршке за пренос података у индустријском интернету ствари. Циљ истраживања је остварење веће контролабилности периферних уређаја и стварање могућности за искоришћавање већих потенцијала које они посједују. Сама форма датотеке која учествује у преносу доприноси читљивости и прегледности и самим тим крајњем кориснику олакшава комуникацију, као и комплетан процес управљања машином, а израчунавање временског ограничења на основу ширине пропусног опсега обезбјеђује поузданост система, то јест даје својеврсну гаранцију да пренос датотеке неће бити прекинут због истека временског рока.

2. КОНЦЕПТ КОМУНИКАЦИЈЕ У IoT СИСТЕМУ

Основни концепт комуникације у оквиру интернета ствари заснован је на раду сензора и актуатора и њихове спреге са управљачком јединицом која може, али и не мора да буде под контролом корисника. Најчешће коришћен начин корисничке интеракције јесте путем система облака (енгл. *cloud*) који кориснику даје директан приступ машини у којој су смјештени сензори и актуатори.

Међутим, како је услед убрзаног развоја рачунарских технологија дошло до наглог повећања броја појединачних, али и читавих мрежа сензорских и актуаторских уређаја повезаних на интернет, а самим тим и до значајног пораста у количини генерисаних података, открили су се недостаци система облака. Због превелике количине података постало је немогуће да се сви подаци обраде, чиме су изазвана кашњења, преоптерећења мреже, то јест, систем је изгубио поузданост. Да би се превазишао овај проблем, у постојећи систем уведени су посреднички уређаји на којима се врши обрада, агрегација и селекција података за слање кориснику, то јест формира се међуслој који носи назив *fog* (енглеска

ријеч за маглу) [1]. На слици 1 приказана је архитектура система облака са *fog* слојем.



Слика 1. Резултати симулације

3. CANOPEN ПРОТОКОЛ

Controller Area Network Open (CANOpen) је комуникациони протокол заснован на CAN магистралаи, који помоћу сета стандарда дјелимично имплементира више нивое *Open Systems Interconnection* (OSI) модела. Уређаји у оквиру CANOpen протокола представљени су као чворови, а све информације везане за конфигурацију и функционалност чвора уписане су у табели која се назива рјечник објеката. Пренос података се одвија помоћу два главна комуникациона протокола: PDO (*Process Data Object*) и SDO (*Service Data Object*) [2].

3.1. PDO протокол

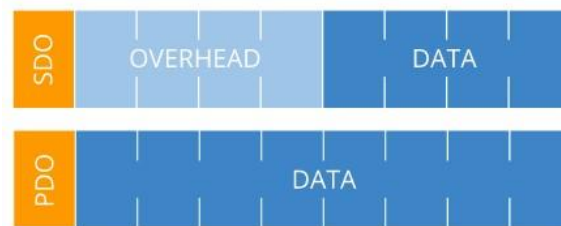
PDO протокол служи за дистрибуцију контролних и статусних информација високог приоритета, као и података који се преносе у реалном времену. Оквир се састоји од 8 бајтова чистих апликативних података. Погодан је за трансфер података који често мијењају вриједност и шаљу се периодично (подаци са сензора или подаци за специфичне актуаторе).

3.2. SDO протокол

SDO протокол омогућава приступ свим уносима рјечника објеката и заснован је на принципу клијент-сервер комуникације са потврдом. Чвор чијем се рјечнику објеката приступа представља сервер, а чвор који приступа рјечнику објеката другог чвора представља клијента. За разлику од PDO протокола, оквир SDO протокола носи 4 бајта апликативних података, док остала 4 припадају заглављу. Разлика у садржају оквира PDO и SDO протокола приказана је на слици 2.

Постоје три врсте преноса података у оквиру SDO протокола: 1) убрзани – користи се када подаци могу да стану у једну поруку, 2) нормални – назива се још и сегментирани и користи се када подаци не могу да стану у једну поруку и 3) блок трансфер – оптимизовани режим преноса за уносе у рјечнику објеката који садрже велику количину података. Један

блок може да садржи до 889 бајтова (127 порука по 7 бајтова) и тек након трансфера читавог блока обавља се потврда о пријему.



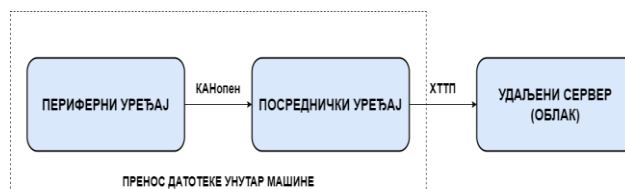
Слика 2. Оквири SDO и PDO протокола [3]

4. HTTP ПРОТОКОЛ

Hypertext Transfer Protocol (HTTP) протокол је један од најзаступљенијих и најпоузданијих протокола за комуникацију са сервисом облака. HTTP је протокол апликативног слоја који служи за пренос хипертекста, односно кода написаног помоћу *Hypertext Markup Language* (HTML) језика. Овај протокол ради на принципу захтјев-одговор. Клијентски захтјев се састоји из два дијела, а то су заглавље и тијело. У заглављу су садржане информације о верзији протокола, команда, име траженог документа итд. Тип захтјева дефинисан је командом односно наредбом. Неке од наредби су: *get*, *post*, *put*, *delete* итд. Одговор сервера се такође састоји из два дијела, а то су извјештај о статусу захтјева и сам садржај одговора. Извјештај о статусу је у форми троцифреног броја и кратког текста, док одговор има исту форму као и захтјев [4].

5. ПРЕНОС ДАТОТЕКЕ

Пренос датотеке је инициран од стране периферног уређаја који се налази у машини и повезан је CAN магистралом са посредничким уређајем. По пријему датотеке, посреднички уређај исту просљеђује удаљеном серверу и на тај начин датотека постаје доступна кориснику. Према томе, процес преноса датотеке може се подијелити у два дијела (слика 3): 1) дио за пренос датотеке унутар машине – пренос између периферног и посредничког уређаја и 2) дио за пренос датотеке од машине до удаљеног сервера.



Слика 3. Концепт рјешења

5.1. Пренос датотеке унутар машине

Пренос датотеке унутар машине обавља се помоћу CANOpen протокола, односно помоћу блок трансфера који представља један од типова преноса у оквиру SDO протокола. Рјечник објеката посредничког уређаја конфигуриран је тако да омогући писање датотеке од стране периферног уређаја, а самим тим дефинишу се

улоге оба уређаја – посреднички уређај има улогу SDO сервера, а периферни уређај има улогу SDO клијента. Пренос датотеке почиње приступом клијента рјечнику објеката сервера. Прво се провјерава тренутни статус преноса и уколико је сервер доступан, то јест ако нема недовршеног преноса, уписује се име датотеке и започиње пренос по блоковима. У једном захтјеву могућ је пренос 127 блокова по 7 бајтова односно 889 бајтова података, након чега сервер шаље потврду клијенту. Наведена процедура се понавља све док се не пренесе читава датотека.

5.2. Пренос датотеке од машине до удаљеног сервера

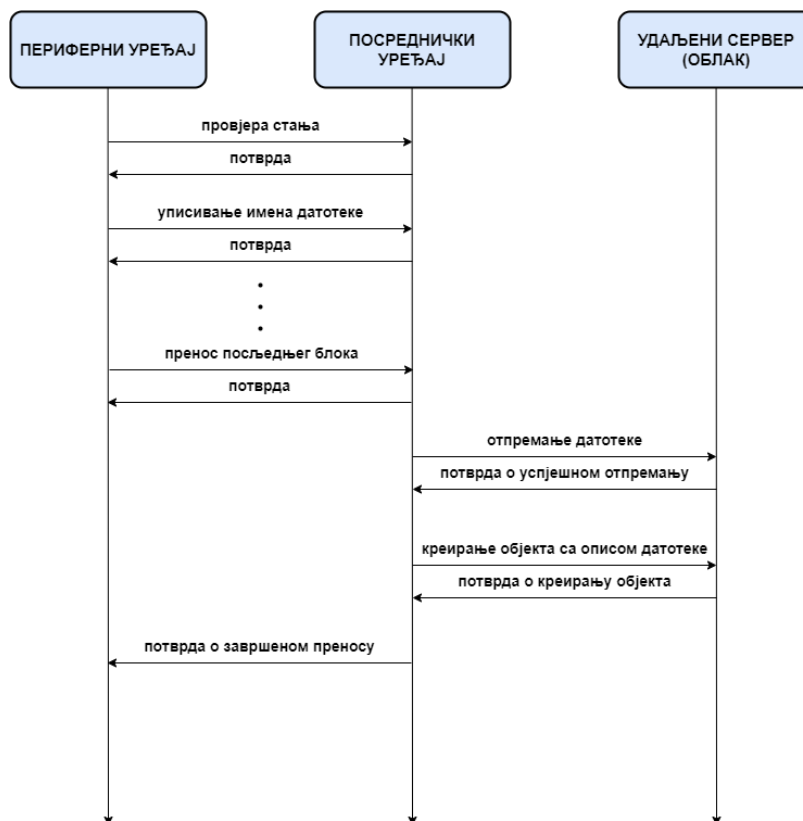
Процес преноса започиње уписом датотеке у директоријум. Након што је датотека у потпуности уписана, активира се механизам за провјеру и потенцијално просљеђивање датотеке на удаљени сервер. Механизам за провјеру обухвата провјеру величине датотеке, провјеру везе са удаљеним сервером и провјеру доступности CANOpen канала за пренос. Уколико су све провјере задовољене, процес преноса прелази у наредну фазу - израчунавање временског ограничења.

5.3. Израчунавање временског ограничења

Временско ограничење представља максималан период трајања отпремања датотеке, а његова вриједност добија се као количник величине датотеке и експериментално утврђеног коефицијента (1). Свака ширина пропусног опсега има одговарајући коефицијент, а његова вриједност дефинисана је као најнижа брзина преноса при оптималном раду комуникационог канала за слање података на интернет (енгл. *uplink*).

$$timeout = \frac{file_size}{bandwidth_coefficient} \quad (1)$$

Процес преноса наставља се отпремањем датотеке у репозиторијум који се налази на удаљеном серверу, а као одговор уређај добија адресу на којој се налази датотека. Након тога, креира се објекат који садржи информације о датотеци – име датотеке и адреса на којој се налази. Обје операције (пренос датотеке и креирање објекта) реализују се помоћу POST захтјева у оквиру HTTP протокола. Комплетан процес преноса приказан је дијаграмом комуникације на слици 4.



Слика 4. Дијаграм комуникације

6. РЕЗУЛТАТИ РАЧУНАЊА ВРЕМЕНСКОГ ОГРАНИЧЕЊА

Резултати тестова израчунавања временског ограничења на основу пропусног опсега помоћу наведених испитних случајева садржани су у табели 1. На основу резултата тестирања приказаних у табели 1, могуће је извести закључак о успјешном раду система. У свим испитним случајевима измјерено вријеме преноса мање је од израчунатог временског ограничења чиме је одговорено на захтјеве тестирања.

Међусобним поређењем резултата тестирања за *EDGE* и *UMTS* ширине пропусног опсега, могуће је уочити својеврсну аномалију. У сва три испитна случаја измјерено вријеме за *UMTS* ширину пропусног опсега је веће, иако ову ширину карактерише већа просјечна брзина од ширине *EDGE*. Овакви резултати настају због чињенице да је уређај који је слао датотеку у *EDGE* ширини пропусног опсега имао већу јачину сигнала. Дакле у процесу преноса датотеке осим фактора ширине пропусног опсега фигурише и

фактор јачине сигнала што је потврђено на основу поређења наведених испитних случајева.

Табела 1. Резултати рачунања временског ограничења

Величина Датотеке (Mb)	Ширина Пропусног Опсега	Израчунато Временско Ограничење (S)	Измјерено Вријеме Преноса (S)
1	EDGE	273	6.117
	UMTS	40	8.314
	LTE	8	6.147
10	EDGE	2730	28.906
	UMTS	409	36.945
	LTE	81	25.075
40	EDGE	10923	85.980
	UMTS	1638	135.021
	LTE	327	83.704

7. ЗАКЉУЧАК

Рад је показао да је могуће креирати поуздан систем за пренос дијагностичких података од периферног уређаја унутар машине до корисника који подацима приступа на сервису облака. За разлику од најчешће коришћене примјене CANOpen протокола за слање појединачних података, ово рјешење омогућава њихово слање у форми датотеке. На овај начин остварује се већа контролабилност периферних уређаја и ствара се могућност за искоришћавање већих потенцијала које они посједују. Сама форма датотеке доприноси читљивости и прегледности и самим тим крајњем кориснику олакшава комуникацију, као и комплетан процес управљања машином. Кључни резултати тестирања односе се на адекватно израчунато временско ограничење за пренос датотеке и на непромијењен садржај саме датотеке.

8. ЛИТЕРАТУРА

- [1] H. -J. Hong, "From Cloud Computing to Fog Computing: Unleash the Power of Edge and End Devices," 2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Hong Kong, China, 2017, pp. 331-334, doi: 10.1109/CloudCom.2017.53
- [2] Olaf Pfeiffer, Andrew Ayre, Christian Keydel, *Embedded Networking with CAN and CANopen*, Copperhill Technologies Corporation, 2008.
- [3] CSS Electronics CANLogger1000: canlogger1000.csselectronics.com/img/CANopen-PDO-SDO-Difference-Example-Message-Format-CAN-Bus.svg
- [4] И. Башичевић, М. Поповић, В. Ковачевић, „Основи рачунарских мрежа 1“, ФТН, 2019.

Кратка биографија:



Саша Бојанић рођен је 1996. године у Бањалуци, Република Српска, Босна и Херцеговина. Звање дипл. инж. електротехнике и рачунарства стекао на Факултету техничких наука у Новом Саду. На истом факултету, године 2021. уписује се на мастер академске студије на студијском програму Рачунарство и аутоматика. Запослен као Софтвер инжињер на институту РТ-РК у Бањалуци контакт: sasa-bojanic@hotmail.com



Иван Каштелан рођен је у Београду, 1985. године. Докторску дисертацију на Факултету техничких наука из области Рачунарска техника и рачунарске комуникације одбранио је 2014. године. контакт: ivan.kastelan@uns.ac.rs

СИСТЕМ ЗА ПРАЋЕЊЕ ПРИСУСТВА НА ДОГАЂАЈИМА**SYSTEM FOR MONITORING PARTICIPATION IN EVENTS**Светозар Вулин, *Факултет техничких наука, Нови Сад***Област - РАЧУНАРСТВО И АУТОМАТИКА**

Кратак садржај – У овом раду је описано решавање проблема праћења присуства на догађајима. Рад се заснива на опису различитих проблема и представљања могућих решења. Од описаних решења, одабрано је једно решење, које је имплементирано и описано у систему AirСофт платформе.

Кључне речи: догађај, валидација идентитета, валидација присуства, платформа

Abstract – This paper describes the solution to the problem of tracking attendance at events. The work is based on the description of various problems and the presentation of possible solutions. Of the described solutions, one was selected, implemented, and described in the specific AirSoft platform system..

Keywords: event, identity validation, presence validation, platform

1. УВОД

Овај рад се бави детаљним анализирањем и поређењем различитих приступа и решења за доказивање присуства особа на разним догађајима, било да су они формални или неформални, као што су спортска такмичења, конференције, предавања или приватна окупљања. Рад се бави проблемом потврде идентитета посетилаца, разматрајући како ефикасно извршити валидацију без стварања гужви, посебно на догађајима са великим бројем људи. За такве догађаје предлага се употреба аутоматизованих система који раде без људског фактора. С друге стране, за мање догађаје са вишим захтевима за сигурношћу, потребна је детаљнија провера идентитета како би се спречила злоупотреба и крађа идентитета.

Циљ рада јесте да представи и упоређи различите методе које особе могу користити да докажу своје физичко присуство на догађају, укључујући имплементацију и тестирање одабраног решења у оквиру AirСофт (АСФ) платформе. Такође, у раду су донети закључци о ефикасности имплементираних решења, потенцијална унапређења и будући развој различитих метода потврде присуства.

2. ПРОБЛЕМ ПОТВРДЕ ПРИСУСТВА НА ДОГАЂАЈИМА

Проблем потврде присуства на догађајима је чест изазов, како за веће, тако и за мање скупове. Потребно

је да се провери да ли су посетиоци који долазе на догађај, заправо они који су позвани или пријављени. За веће догађаје, обично се ангажују волонтери или запослени који би вршили проверу идентитета посетилаца на улазу. То може бити временски захтевно и стресно, а такође и стварати редове и кашњења. Постоји и ризик од злоупотребе позивница и крађе идентитета. Као решење предлага се развој система или софтвера који би убрзао и олакшао процес верификације. У раду се анализира проблем из две перспективе: корисника и администратора догађаја.

Администратор може бити организатор догађаја или неко ко има ауторитет за потврду присуства. Конкретан пример решења овог проблема анализира се у контексту АСФ платформе, која је имплементирана за организацију догађаја и валидацију присуства учесника.

2.1. Перспектива администратора

Администратор представља врсту ауторитета, односно ентитет који је одговоран за догађај, као и то да његово одржавање протекне како је планирано. Администратор треба да врши креирање догађаја, као и дефинисање времена и места одржавања. Уколико систем утврди да нису сви пријављени корисници присутни, администратор може да реагује на време. Такође, одговорност администратора је да се спречи злоупотреба позивница, лажирање идентитета, и сл. Особа која је администратор може, а не мора представљати особу која врши верификацију и може, а не мора да управља догађајем. У раду је потребно осврнути се на могуће приступе који би искључили администратора из процеса потврде делегирањем посла на систем.

2.2. Корисничка перспектива

Када се корисник пријави на догађај, очекује се да ће он бити одржан на одређеном месту и у дефинисаном времену. Важан део учешћа на догађају јесте верификација присуства корисника, која обухвата потврду идентитета и физичког присуства. Идентификација је важна због ризика од злоупотребе, као што је појављивање других особа уместо позваних, што може нарушити интегритет и приватност догађаја и створити конфликте при валидацији пријава.

Постоји проблем приватности, који се јавља приликом обраде личних података, јер могу постојати ризици од њихове злоупотребе, укључујући крађу идентитета или новчаних средстава. Због тога је тешко стећи поверење корисника и прикупљати њихове податке.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Горан Сладић, ред. проф.

2.3. Кораци у решавању проблема

2.3.1. Потврда идентитета

Да би се верификовао идентитет пријављеног корисника на догађају, неопходно је проверити његов идентитет, а узимајући у обзир учесталост крађа идентитета на интернету, важно је осигурати сигурно складиштење личних података. Постоје различити методи за потврду идентитета: скенирање личног документа, поређење фотографије особе са онима у систему користећи вештачку интелигенцију и употреба јединственог идентификационог броја додељеног кориснику на платформи. Свака од ових метода има своје предности и мане, као што је ризик од крађе личног документа приликом скенирања, коришћење фотографија лошијег квалитета приликом поређења вештачком интелигенцијом, крађа јединственог броја. Алтернатива употреби јединственог броја би могао бити QR код [1], који нуди већу сигурност, и не може бити украден као број јер се не може запамтити. Ове методе захтевају баланс између сигурности и практичности у циљу ефикасне и поуздане верификације идентитета на догађајима.

2.3.2. Валидација пријаве

Након идентификације корисника, потребно је проверити да ли је његова пријава за догађај валидна и да ли се налази на списку пријављених корисника на платформи. Пријава се сматра валидном ако је корисник или добио позив од организатора догађаја или се сам пријавио путем платформе. Овај процес валидације пријаве је важан да би се осигурало да на догађају буду само они који су заиста пријављени или позвани. Валидација се врши у оквиру платформе, где се подаци из корисничког профила упоређују са списком пријављених корисника. Успешна валидација потврђује да је корисник пријављен за догађај, а након тога следи провера његовог физичког присуства.

2.3.3. Потврда физичког присуства на догађају

2.3.3.1. Потврда без присуства администратора

Уколико је потребно ослободити администратора одговорности за потврду физичког присуства, систем треба да подржи методе које се ослањају на међусобну верификацију присутних особа на догађају или да омогући корисницима да самостално потврде своје присуство. Ово подразумева дефинисање броја и начина верификација потребних за валидацију од стране других особа или развој механизма који корисницима омогућава да индивидуално верификују своје присуство скенирањем докумената или потврдом локације преко платформе.

2.3.3.2. Потврда са присуством администратора

Овај приступ се ослања на особу која представља ентитет администратора, односно особу која има већу одговорност и задужења од обичног корисника, и која би вршила улогу валидатора. Администратор има могућност да изврши потврду присуства за сваку особу која се појавила на догађају, скенирањем кода, личног документа, или неким другим видом валидације. Проблем приликом имплементације овак-

вог приступа је у томе што се систем ослања на администратора, што укључује фактор људске грешке и обавезе да мора да постоји особа која ће бити ауторитет и обављати посао валидације, што може резултовати временски захтевном процесу.

3. ПОРЕЂЕЊЕ РАЗЛИЧИТИХ ПРИСТУПА ПОТВРДЕ ПРИСУСТВА

3.1. Геолокацијска претрага

Геолокацијска претрага користи ГПС (енг. *Global Positioning System, GPS*, [2,3]) технологију за одређивање географске локације уређаја, као што су мобилни телефони или рачунари. Овај вид претраге се ослања на ГПС технологију, која представља мрежу сателита и пријемних уређаја, који емитују сигнал, са циљем одређивања локације уређаја на земљи. Приликом креирања догађаја, дефинише се његова локација, а корисници који су се пријавили могу да потврде своје присуство тек када физички стигну на задату локацију. Систем проверава да ли се географске координате корисника поклапају са координатама места одржавања догађаја. Могући проблем настаје ако неко други користи уређај пријављеног корисника, што се може решити додатном верификацијом идентитета, као што је скенирање личних докумената или QR кодова. Овај систем омогућава самосталну потврду присуства од стране корисника, независно од администратора.

3.2. Скенирање QR кода од стране других посетилаца

Скенирање QR кода од стране других посетилаца на догађају представља метод валидације присуства који се ослања на друге посетиоце са циљем да они изврше валидацију присуства. Корисници добијају QR код при пријави на догађај, који је валидан само за тај догађај. На догађају, корисници морају скенирати кодове других корисника и на тај начин валидирати њихово присуство. Број потребних скенирања зависи од величине догађаја и може се дефинисати од стране администратора. За мање догађаје, потребно је да свака особа скенира свачији QR код, док за веће догађаје може постојати одређена граница скенирања. Овај метод повећава ниво заштите, јер га је теже заобићи са повећањем броја потребних скенирања. На примеру АСФ платформе, администратор може поставити лимит да је потребно скенирање од најмање 60% присутних, што осигурава чињеницу да и играчи из супротних тимова морају међусобно скенирати кодове. Ово омогућава да се валидација обави без укључивања администратора, али може бити временски захтевно и потенцијално лоше за корисничко искуство.

3.3. Скенирање QR кода од стране ауторитета

Скенирање QR кода од стране ауторитета је метод валидације где особа са ауторитетом скенира QR код генерисан од стране система након пријаве корисника на догађај. Овај код је специфичан за сваки догађај и не може се злоупотребити за друге догађаје. Овај приступ обједињује валидацију идентитета и потврду присуства, а валидацију пријаве врши систем.

Предности укључују једноставност употребе и решавање оба проблема од стране једне особе без потребе за додатним системима. Међутим, овај метод може бити временски захтеван и напоран за особу која се сматра ауторитетом, посебно на већим догађајима, што може довести до негативног корисничког искуства због дугог чекања на валидацију.

3.4. Коришћење NFC технологије за скенирање

NFC (енг. *Near Field Communication*) [4,5] технологија омогућава комуникацију између уређаја на кратким раздаљинама (до 4 цм), ослањајући се на радио комуникацију. Често се користи у паметним уређајима [6] за разне сврхе као што је бежично плаћање. *NFC* може заменити традиционалне методе валидације присуства као што су скенирање QR кодова. Уместо QR кодова, корисници могу користити *NFC* за међусобну верификацију присуства или за верификацију од стране ауторитета. Ова технологија олакшава процес скенирања, смањујући време потребно за валидацију и побољшава корисничко искуство. *NFC* може бити коришћен и за скенирање личних докумената као што су биометријски пасоши. Предности *NFC*-а укључују бржу валидацију и боље корисничко искуство, док је мана што можда немају сви корисници уређаје компатибилне са *NFC* технологијом.

4. СПЕЦИФИКАЦИЈА СИСТЕМА

4.1. Спецификација апликације и изазови

АСФ је спорт заснован на симулацији рата, где се користе реплике оружја и различити приступи игри, који се могу организовати на различитим теренима, у зависности од броја играча. Циљеви игре укључују елиминисање противничког тима, постављање бомби или освајање заставица.

АСФ платформа је мобилна апликација која омогућава креирање и организацију догађаја, турнира и тимова. Главни изазови у имплементацији решења за валидацију присуства корисника укључују питања сигурности, као што су заштита корисничких података, интегритет платформе и робусност самог решења. За обезбеђивање сигурности система важни су механизми као што су контрола приступа [7], ауторизација и аутентификација [8], као и складиштење хешованих (енг. *hash*, [9]) података у бази података. Интегритет система се односи на његово поуздано функционисање, док робусност решења подразумева да сви корисници који присуствују догађају буду успешно и поуздано валидирани.

4.2. Спецификација система

Систем се састоји од три главне компоненте: сервера, скенера и корисничке апликације. Сервер представља логичку компоненту, која је задужена за складиштење података и служи да би се прибавили подаци који су потребни за валидацију у одређеном тренутку. Скенер представља уређај који је повезан на платформу, и даје јој одобрење да користи његову камеру. Између сервера и скенера се дешава константна комуникација, из разлога што скенер све податке које прочита и валидира, шаље серверу, који даље врши валидацију

и потврду да ли је процес извршен успешно или није. Корисничку апликацију користе корисници да би комуницирали са платформом, како би се пријавили на догађај и потврдили своје присуство на истом. Они у корисничкој апликацији имају преглед свих догађаја који су креирани, и имају могућност да се пријаве на жељени догађај. Када се корисник пријави, и дође време за одржавање догађаја, корисник мора да валидира своје присуство користећи опције из корисничке апликације. Скенер прво скенира QR код који корисник приложи из апликације, и затим се обавља провера идентитета. Након тога корисник може да обави верификацију да се налази на догађају помоћу геолокацијске претраге.

5. ИМПЛЕМЕНТАЦИЈА СИСТЕМА

5.1. Архитектура система

Као што је већ наведено у одељку 4.2., систем се састоји из три главне компоненте. За изградњу изгледа и корисничког интерфејса, као и интерфејса који користи скенер, коришћена је библиотека *React.js*. За имплементацију скенер компоненте и система за читање и дешифровање QR кодова, коришћена је нпм (енг. *npm*) библиотека *react-qr-reader*. За сервер је коришћен *Node.js* и око њега радни оквир (енг. *framework*) *genzy.io* који се ослања на *Express.js* радни оквир. За чување података је коришћена *NoSQL* база података *MongoDB*.

5.2. Потврда валидације идентитета скенирањем QR кода

Приликом регистрације на платформу, сваки корисник након што је унео своје податке, добија јединствени QR код, који касније може да искористи у сврху валидације. Овај код не би требао да се дели са другима, да не би био злоупотребљен. Он једнозначно одређује власника профила и садржи његове личне податке.

Приликом скенирања, потребно је да корисник принесе свој уређај (на којем се налази QR код) скенеру, који затим врши скенирање и валидацију података. Након што су прочитани подаци са QR кода, потребно их је упоредити са подацима тренутно улогованог корисника. Валидација се обавља тако што се подаци који су сачувани приликом пријављивања на догађај, пореде са подацима који се налазе складиштени у QR коду. Уколико се информације поклапају, корисник је валидиран и његово присуство је успешно потврђено од стране система.

5.3. Имплементација геолокацијске претраге

Као што је објашњено у одељку 3.1., геолокацијска претрага представља претрагу по географској ширини и дужини. Након што је корисник дошао на догађај, потребно је да валидира своје присуство, тако што ће упоредити своју локацију, са предвиђеном локацијом за догађај. Ова радња се изводи тако што корисник у апликацији треба да означи да жели да потврди своје присуство коришћењем свог географског положаја (Листинг 1.). Предуслов за ову операцију јесте да је корисник одобрио апликацији да користи локацију његовог уређаја.

Након послатог захтева за валидацију, уређај преузима тренутну локацију и шаље је платформи.

```
navigator.geolocation.getCurrentPosition(
  (position) => {
    const latitude =
position.coords.latitude;
    const longitude =
position.coords.longitude;

    callback({ latitude, longitude },
eventPosition, thresholdInMeters);
  });
```

Листинг 1. Прибављање географских координата

Након што је платформа преузела тренутну локацију, врши се претрага у бази података и проналази се локација која је дефинисана за предвиђени догађај. Из разлога што је геолокација прецизна до неколико метара, потребно је задати одређени праг толеранције, који ће се користити у рачунању могућег опсега у ком треба да се налази уређај да би његова локација била иста као и за дефинисани догађај. Рачунање дистанце између локације корисника и локације задате приликом креирања догађаја рачуна се помоћу помоћне библиотеке *geolib* [10].

```
export const checkIsSamePosition =
(userPosition, eventPosition, threshold)
=> {
  return geolib.getDistance(userPosition,
eventPosition) < threshold;
}
```

Листинг 2. Функција за поређење локација са толеранцијом

Успешном потврдом да је корисник стварно на задатој локацији, његов захтев бива прихваћен.

6. ЗАКЉУЧАК

У овом раду описан је проблем који се тиче доказивања присуства особе на догађају. Како догађај представља било који вид окупљања, формални или неформални, потребно је било навести различите приступе, као и њихове концепте који се користе у решавању овог проблема. У зависности од тога који се догађаји креирају, могуће је изабрати различите приступе и решења. Након поређења решења, одабрана је и описана идеја решења на примеру АСФ платформе. Након тога је дат кратак опис платформе и проблема који се јавља у конкретном случају. Одабрани начини за решавање проблема стављени су у конкретан контекст платформе, описани су изазови који су се јављали приликом имплементације, и приказано је решење у конкретном систему.

Потенцијално унапређење ових решења може да се огледа у напретку саме технологије, где би АИ пружао веродостојније начине за валидацију идентитета, масовнију употребу и имплементацију *NFC* чипова у уређаје, што би омогућило усвајање *NFC* приступа потврде, као и развој нових технологија сличних *NFC*, које би довеле до потенцијалних нових решења и приступа у решавању оваквих проблема.

7. ЛИТЕРАТУРА

- [1] Tiwari, Sumit. "An introduction to QR code technology." 2016 international conference on information technology (ICIT). IEEE, 2016.
- [2] Gentile, Camillo, et al. Geolocation techniques: principles and applications. Springer Science & Business Media, 2012.
- [3] El-Rabbany, Ahmed. Introduction to GPS: the global positioning system. Artech house, 2002.
- [4] Du, Hongwei. "NFC technology: Today and tomorrow." International Journal of Future Computer and Communication 2.4 (2013): 351.
- [5] Al-Ofeishat, Hussein Ahmad, and Mohammad AA Al Rababah. "Near field communication (NFC)." International Journal of Computer Science and Network Security (IJCSNS) 12.2 (2012): 93.
- [6] Silverio-Fernández, Manuel, Suresh Renukappa, and Subashini Suresh. "What is a smart device?-a conceptualisation within the paradigm of the internet of things." Visualization in Engineering 6.1 (2018): 1-10.
- [7] Sandhu, Ravi S., and Pierangela Samarati. "Access control: principle and practice." IEEE communications magazine 32.9 (1994): 40-48.
- [8] Kim, Hokeun, and Edward A. Lee. "Authentication and Authorization for the Internet of Things." IT Professional 19.5 (2017): 27-33.
- [9] Sobti, Rajeev, and Ganesan Geetha. "Cryptographic hash functions: a review." International Journal of Computer Science Issues (IJCSI) 9.2 (2012): 461.
- [10] <https://www.npmjs.com/package/geolib>

Кратка биографија:



Светозар Вулин рођен је у Руми 1999. год. Основне академске студије је завршио 2022. године на Факултету техничких наука у Новом Саду. Мастер рад на Факултету техничких наука из области Рачунарство и аутоматика – Електронско пословање одбранио је 2024. године.

KOMPARATIVNA ANALIZA SISTEMA ZA UPRAVLJANJE SADRŽAJEM COMPARATIVE ANALYSIS OF CONTENT MANAGEMENT SYSTEMS

Milan Govedarica, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIČKO I RAČUNARSKO INŽENJERSTVO

Kratak sadržaj – *Sistemi za upravljanje sadržajem (Content Management Systems - CMS) igraju ključnu ulogu u olakšavanju procesa kreiranja, organizacije i održavanja digitalnih informacija. Ovaj istraživački rad detaljno istražuje različite aspekte CMS-ova koji se mogu primeniti u kreiranju online prodavnica nameštaja. Obuhvata implementaciju veb sajta za prodaju nameštaja u tri odabrana CMS-a, analizira ključne karakteristike svakog sistema, vršeci njihovu komparativnu analizu. Veb sajt za prodaju namještaja razvijen je korišćenjem WordPress-a, Shopify-a i PrestaShop CMS platformi.*

Ključne reči: *Sistemi za upravljanje sadržajem, CMS, veb aplikacije, komparativna analiza*

Abstract – *Content Management Systems (CMS) play a crucial role in simplifying the processes of creating, organizing, and maintaining digital information. This research explores various aspects of CMS that can be applied in the creation of online furniture stores. It encompasses the implementation of a website for selling furniture using three selected CMS platforms, analyzing the key features of each system and conducting a comparative analysis. The website for selling furniture was developed using the WordPress, Shopify, and PrestaShop CMS platforms.*

Keywords: *content management systems, CMS, web apps, comparative analysis*

1. UVOD

U današnjem ubrzanom digitalnom dobu i eri digitalne transformacije, gde se informacije neprestano generišu i dele, efikasno upravljanje digitalnim sadržajem postaje ključni faktor uspeha kako pojedinca, tako i organizacija. U tom kontekstu, Content Management Systems (CMS) igraju ključnu ulogu u olakšavanju kreiranja, organizovanja i održavanja digitalnih informacija. Oni predstavljaju sofisticirane alate koji omogućavaju korisnicima da ne samo kreiraju, uređuju i objavljuju sadržaj već da ga organizuju i održavaju sa lakoćom. CMS sistemi nude intuitivne korisničke interfejs, čineći proces kreiranja veb stranica, blogova ili online prodavnica dostupnim širem auditorijumu. Ovaj rad istražuje različite aspekte izabranih CMS sistema, za potrebe implementacije veb prodavnice.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Stevan Gostojić, red. prof.

Veb sajt za prodaju nameštaja implementiran je korišćenjem WordPress, Shopify i PrestaShop platformi, koji su odabrani kao CMS-ovi koji imaju najširu primenu za izradu veb sajtova koji se bave prodajom.

2. STANJE U OBLASTI

2.1. Osnovne karakteristike CMS-ova

CMS sistemi (Content Management Sistem) predstavljaju vrstu softvera koji omogućava upravljanje i organizovanje digitalnog sadržaja. Ovi sistemi se široko koriste za izradu i održavanje veb stranica, blogova i drugih onlajn platformi [1]. Većina CMS platformi ima zajedničke karakteristike i glavne komponente: administratorski panel, interfejs za korisnike/grupe, URL adrese koje su prilagođene za SEO, podrška za medijske datoteke, slike, video i audio, editor za kreiranje, menjanje i objavljivanje sadržaja, bazu podataka u kojoj se čuvaju podaci o sajtu, sistem šablona i mogućnost promene prema vašim potrebama, ažuriranja, ispravke grešaka, poboljšanja izvornog koda i nivoi zaštite, pomoćna dokumentacija, zajednica korisnika, forumi itd. [2].

Najviše korišćeni CMS sistemi, koji su obrađeni u okviru ovog master rada, su WordPress, Shopify i PrestaShop. Od velikog broja CMS-ova koji su danas popularni i koriste se širom sveta, bilo je teško izabrati 3 sistema koja će se obrađivati. Jako bitan kriterijum za izbor bio je sama statistika o korišćenju tog CMS-a. Drugi kriterijumi koji su omogućili da se dalje suzi izbor jesu da je taj CMS primarno razvijen za veb prodavnice, jer je i sama tema rada primena CMS-a za izradu veb sajta koji se bavi prodajom nameštaja, te da oni daju mogućnost samostalnog kucanja koda jer je bitna stavka da se pri implementaciji postigne originalni željeni dizajn, a bez te mogućnosti bi to bilo jako teško postići.

3. SPECIFIKACIJA

Aplikacija koja je implementirana u okviru ovog rešenja ima dve vrste korisnika: ulogovani korisnik, neulogovani korisnik. U okviru aplikacije definisani su sledeći slučajevi korišćenja za oba korisnika: prijava, registracija, pregled nameštaja kategorije, pregled kategorije nameštaja, filtriranje nameštaja, pregled izabranog nameštaja. Za ulogovanog korisnika definisani su slučajevi korišćenja: ocenjivanje i komentarisanje nameštaja, plaćanje, uvid u istoriju transakcija, uvid i izmena korisničkog profila. Da bi korisnik kupio nameštaj, prvo što mora da uradi je da se prijavi na korisnički nalog. Nakon uspešne prijave,

korisniku se prikazuje početna stranica na kojoj treba da odabere kategoriju nameštaja. Posle odabira željene kategorije, prikazuje mu se stranica sa svim proizvodima koji pripadaju toj kategoriji, gde može odabrati konkretan proizvod. Na stranici sa detaljima o izabranom proizvodu, korisnik može da doda nameštaj u korpu. Nakon toga, uvidom u korpu može da se započne proces plaćanja upotrebom PayPal naloga.

4. IMPLEMENTACIJA

4.1. WordPress

Prvi pokušaj implementacije u WordPress-u je bio putem WordPress.com veb sajta, ali vrlo brzo se moglo primetiti da je limitiran u pogledu funkcionalnosti koje su potrebne za implementaciju odabranog sajta. Takođe, potrebno se pretplatiti na neki od plaćenih paketa kako bi se plugin-ovi mogli instalirati.

Kao rešenje za te probleme moguće je instalirati WordPress i pomoću XAMPP alata podići ga u lokalnom okruženju, što je i urađeno u okviru ove implementacije. Ovaj alat omogućava rad u lokalnom razvojnom okruženju bez potrebe za pristupom internetu.

Potrebni dizajn za početnu stranicu je bilo lako postići upotrebom editora, ali bez plaćenih plugin-ova i tema nije pronađen način za prilagođavanje navigacione trake ulogovanim i neulogovanim korisnicima. Besplatni plugin-ovi su korišćeni za registraciju korisnika, prikaz korisničke slike, ocenjivanje objava i plaćanje. Prijava korisnika, menjanje slike korisnika, dodavanje objava u korisničku korpu je bilo složenije zbog odsustva besplatnih plugin-ova koji imaju mogućnost prilagođavanja CSS-a, pa je za ovaj deo bilo potrebno napraviti prilagođene plugin-ove koji se lako ubacuju upotrebom Shortcode elemenata.

4.2. Shopify

Da bi se moglo krenuti sa implementacijom u Shopify-ju, potrebno se pretplatiti na njihov veb sajt. U okviru ove implementacije izvršena je pretplata na probna 3 meseca koja iznosi 1\$ po mesecu. Nakon pretplate moguće je pristupiti Shopify admin panelu. U okviru tog panela pristupljeno je temama i odabrana je ona koja najviše podseća i odgovara funkcionalnostima i dizajnu veb sajta koji je potrebno implementirati.

Upotrebom editora koda može se pristupiti kodu teme i definisati potrebni HTML, CSS, Javascript ili Liquid kodovi, kao i dodati novi šabloni, sekcije i sl. Za implementaciju prijave i registracije napravljeni su prilagođeni šabloni. U okviru navigacionog menija iz admin panela napravljena su dva menija, jedan za prijavljene korisnike a jedan za korisnike koji nisu prijavljeni. U tim menijima definisane su stranice kojima korisnik može da pristupi u zavisnosti od toga da li je prijavljen ili ne. Iz admin panela pristupljeno je kolekcijama gde su definisane različite kategorije nameštaja.

Pošto kolekcija ne može da ima podkolekcije, a na ovom sajtu to je jedna od funkcionalnosti koja se mora omogućiti, pronađeno je rešenje koje podrazumeva upotrebu tagova. Prilikom definisanja proizvoda moguće je proiz-

vodu dodati tagove koji u ovom slučaju predstavljaju podkategorije. Takođe, za svaki proizvod definisane su i njegove slike, naziv, cena, opis i kategorija.

4.3. Prestashop

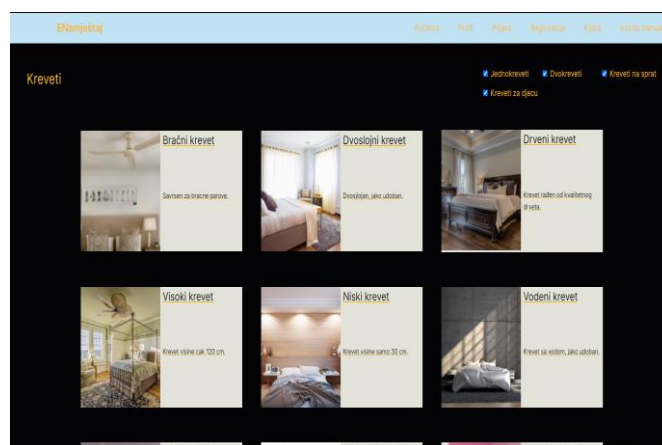
U okviru ovog rešenja, PrestaShop, kao i WordPress aplikacija, podignuta je pomoću XAMPP alata u lokalnom okruženju. Nakon podizanja moguće je pristupiti PrestaShop admin panelu. U okviru tog panela može da se upravlja narudžbinama, katalogima, korisnicima, modulima, dizajnom, plaćanjem itd. Za razliku od WordPress-a i Shopify-a, u PrestaShop-u ne postoji integrisani kod editor, pa je izmena fajlova vršena lokalno na računaru.

Što se tiče prijave, registracije i profila korisnika, uglavnom je menjan samo stil komponenti i teme s tim da je bitno napomenuti da PrestaShop ne podržava dodelu slike korisničkom profilu pa samim tim ta funkcionalnost nije implementirana unutar ovog rešenja. Koje sve informacije korisnik može popuniti pri registraciji izabrane su takođe u okviru parametara prodavnice. Prihvatanje uslova i politike privatnosti su obavezna polja pri registraciji korisnika.

Za komentarisane i ocenjivanje proizvoda od strane korisnika iskorišćen je Product Comments modul. U okviru admin panela može da se upravlja i plaćanjima i metodama za plaćanje. Postoji mogućnost povezivanja sa PayPal nalogom ali taj nalog mora biti poslovni nalog (ne postoji sandbox mod) zbog čega to nije urađeno u okviru ovog rešenja.

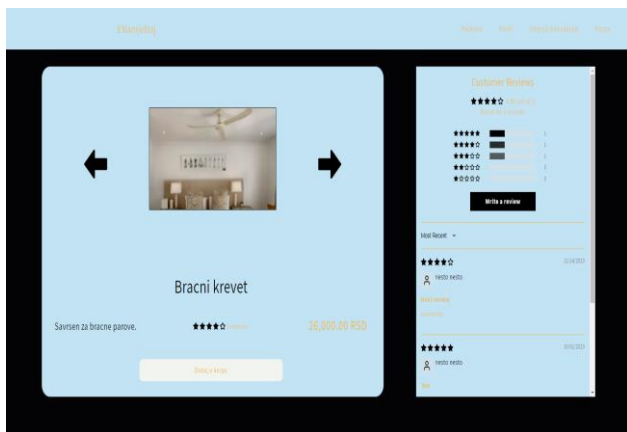
5. DEMONSTRACIJA

U okviru ovog dela biće prikazani scenariji korišćenja aplikacije.



Slika 1. Prikaz nameštaja jedne kategorije u WordPress-u

Kada korisnik izabere željenu kategoriju nameštaja, na ekranu se prikazuje stranica, kako je ilustrovano na slici 1. Na ovoj stranici, korisniku je omogućeno filtriranje nameštaja po podkategorijama, kao i izbor konkretnih komada nameštaja koji ga zanimaju.



Slika 2. Prikaz izabranog nameštaja u Shopify

Detalji o izabranom nameštaju prikazuju se na stranici koja je prikazana na slici 2. Na toj stranici korisnik može da vidi slike, opis, naziv, cenu nameštaja kao i ocene i komentare drugih korisnika veb aplikacije.



Slika 3. Prikaz korpe u Prestashop-u

Na slici 3 prikazana je korpa korisnika gde korisnik ima mogućnost da upravlja sadržajem korpe na različite načine. U okviru korpe, korisnik može lako povećati ili smanjiti broj proizvoda, kao i ukloniti konkretan proizvod iz korpe po svojoj želji.

6. DISKUSIJA

U okviru ovog poglavlja biće sumirani rezultati pomenutog istraživanja, kao i glavne prednosti i mane korišćenih CMS platformi.

6.1. WordPress

WordPress je poznat po svojoj jednostavnosti korišćenja, ima mogućnost pisanja vlastitih dodataka (plugin-ova) pa na taj način pruža dodatnu fleksibilnost. Ima intuitivan korisnički interfejs koji omogućava čak i osobama bez velikog tehničkog znanja da lako upravljaju sadržajem svoje veb stranice, a baš zbog toga ima veliku zajednicu korisnika i razvojnih programera [3].

Takođe, sadrži veliki broj tema (templates) i dodataka (plugins) koji omogućavaju dodavanje različitih funkcionalnosti veb stranici [4]. Neke od osnovnih prednosti koje olakšavaju rad u WordPress-u pri implementaciji opisanog sajta za prodaju nameštaja i čine ga funkcionalnijim su: besplatan je, laka integracija sa PayPal-om i testiranje u sandbox modu, moguće je napraviti interfejs za promenu korisničke slike, postoji grafički i kodni uređivač, mogućnost dodavanja prilagođenih atributa za objave, upravljanje kategorijama i podkategorijama, korisničkim ulogama, ograničenjima pristupa i navigacionim menijima iz admin panela.

Ono što bi moglo da bude problem za korisnike jeste odsustvo besplatnih tema sa više menija što otežava uređivanje navigacionog menija (navbar), a nameštaj se vodi kao objava u administratorskom panelu, te ne postoji poseban meni za upravljanje narudžbama. Ova ograničenja mogu predstavljati izazove za sajtove koji se fokusiraju na e-prodaju i imaju potrebu za efikasnim i lakim upravljanjem narudžbama.

6.2. Shopify

Shopify je popularna platforma za e-trgovinu koja omogućava pojedincima i kompanijama da lako otvore, upravljaju i razvijaju svoje online prodavnice. Iako je plaćena, ističe se jednostavnošću i brzinom postavljanja web prodavnice. Nudi grafički i kod editor koji omogućavaju prilagođavanje izgleda stranica, dok "orders" meni pruža pregled narudžbi, a "customers" meni olakšava upravljanje korisnicima. Koristi sličan sistem stranica kao WordPress. Iako Shopify ima određene okvire i ograničenja, nudi prilagodljivost kroz izbor različitih tema i dodataka, kao i mogućnost integracije sa različitim platformama za procesuiranje plaćanja [5] [6].

Glavne prednosti koje su se istakle tokom implementacije ovog CMS-a su: specijalizovan je za web prodavnice, postoji poseban meni za narudžbe, proizvode, upravljanje korisnicima, jednostavna integracija sa PayPal-om, mogućnost definisanja menija (navigacije) upotrebom admin panela.

Neke od stvari koje su nedostajale pri implementaciji sajta su: nije besplatan, pa je bilo potrebno uzeti mesečnu pretplatu pri implementaciji, kolekcije nemaju podkategorije - bez mogućnosti dodavanja podkategorija (podkolekcija) u kolekcije, prodavci mogu iskusiti ograničenja u strukturnom organizovanju proizvoda, nije moguće napraviti prilagođeni interfejs za korisničku izmenu e-mail adrese (od strane korisnika) i slike profila.

6.3. Prestashop

PrestaShop je još jedna popularna platforma za e-trgovinu koja omogućava kreiranje i upravljanje online prodavnicama [7]. Besplatan je i otvorenog koda. Otvoren kod podstiče razvojnu zajednicu i doprinosi stalnom poboljšanju platforme. PrestaShop podržava višejezičnost i viševalutnost što je posebno korisno za online prodavnice koje ciljaju međunarodno tržište. Omogućava integraciju sa PayPal-om, iako testiranje u sandbox modu zahteva poslovni račun [8]. Neke od bitnijih prednosti ovog CMS-a su: specijalizovan je za web prodavnice, mogućnost definisanja kategorija sa podkategorijama, nudi specijalizovane "orders" i "customers" menije zajedno s organizacijom proizvoda, što znatno olakšava pravljenje web prodavnica., integracija sa PayPal-om (iako bez sandbox-a).

Nema ugrađeni kod i grafički editor, za razliku od WordPress-a i Shopify-a. Ne postoji mogućnost uređivanja navigacionog menija iz admin panela, pa zahteva ručno uređivanje fajla. Obavezno je prihvatanje uslova i politike privatnosti pri registraciji korisnika na web prodavnicu, i nije moguće dodijeliti sliku korisničkim profilima.

Ono što je zajedničko za sve CMS-ove jeste to što koriste teme i nude slične interfejse za prilagođavanje web

prodavnice, kao i omogućavaju upravljanje kategorijama, proizvodima/objavama, korisnicima i ulogama.

7. ZAKLJUČAK

Na osnovu opsežne analize kao i iskustva stečenog tokom implementacije, može se sumirati sledeće: ukoliko je glavni cilj fleksibilnost u dizajnu i funkcionalnosti veb prodavnice odličan izbor je WordPress, ali zahteva malo više tehničkog iskustva u poređenju sa Shopify-em; za brzo i lako podizanje veb prodavnice bez detaljnog upuštanja u tehničke aspekte ali sa određenim mesečnim pretplatama izbor bi bio Shopify; PrestaShop kao i WordPress zahteva više tehničkog znanja ali potencijalno nudi veću fleksibilnost od Shopify-a i nije loš izbor kada korisniku ne smeta ručno mijenjanje fajlova upotrebom eksternih editora.

Neki pravci daljeg istraživanja i analiziranja CMS-ova kao sistema za kreiranje, organizaciju i održavanje veb prodavnica su:

Bezbednost CMS-ova – istraživanje bezbednosti CMS-ova je ključno kao put ka razumevanju potencijalnih rizika i sigurnosnih rupa. Osiguravanje bezbednosti veb sajta važan je aspekt upravljanja online prisustvom, posebno je bitno zaštititi osetljive informacije o transakcijama i kupcima.

SEO prilagođenost – istraživanje SEO prilagođenosti CMS-ova pomaže u razumevanju kako platforme podržavaju optimizaciju za pretraživače. CMS koji podržava efikasnu SEO optimizaciju pomaže nam da izgradimo vidljivost i da rangiramo aplikaciju više na pretraživačima, što može povećati posete i dovesti do povećanja prodaje i broja korisnika.

Višejezička i multivalutna podrška – istraživanje ovih aspekata važno je za postizanje većeg međunarodnog ili višejezičkog auditorijuma. Kako bi se prilagodio različitim jezičkim i kulturnim potrebama korisnika, CMS bi trebalo da omogući laku implementaciju i upravljanje sadržajem na različitim jezicima i sa različitim valutama.

Analiza prodaje i korišćenja sajta – istraživanje alata CMS-a za analizu prodaje i korišćenja sajta je bitno za dobijanje uvida u efikasnost veb sajta i korisničkog iskustva. Praćenje metrika kao što su transakcije, popularni proizvodi, korisnički angažman i statistike prodaje može doprineti boljem usmeravanju i optimizaciji sajta.

8. LITERATURA

- [1] “Šta je CMS?”, Oxford Web Studio, November 5, 2023. [Online]. Available: <https://www.oxfordwebstudio.com/da-li-znate/sta-je-cms.html>. [Accessed Dec. 10, 2023].

- [2] “Šta je CMS? Upotreba CMS sistema za održavanje sajtova”, Markom Dizajn, December 8, 2011. [Online]. Available: <https://www.markomdizajn.com/blog/sta-je-cms>. [Accessed Dec. 10, 2023].
- [3] “WordPress – Karakteristike – Prednosti i mane”, IT Usluge, April 10, 2022. [Online]. Available: <https://www.it-usluge.net/wordpress-karakteristike-prednosti-i-mane.html>. [Accessed Dec. 16, 2023].
- [4] “WordPress”, Wikipedia, January 25, 2024. [Online]. Available: <https://en.wikipedia.org/wiki/WordPress>. [Accessed Jan. 26, 2024].
- [5] R. Nguyen, “Shopify Features and Benefits You Need To Know About”, Shopify Knowledge, December 1, 2023. [Online]. Available: <https://pagefly.io/blogs/shopify/shopify-features>. [Accessed Jan 26, 2024].
- [6] T. Orem and R. Murphy “What Is Shopify and How Does It Work?”, Nerdwallet, April 13, 2023. [Online]. Available: <https://www.nerdwallet.com/article/small-business/what-is-shopify>. [Accessed Dec 16, 2023].
- [7] “What is PrestaShop?”, SupportHost, September 9, 2022. [Online]. Available: <https://www.prestashop-project.org/about/>. [Accessed Jan 28, 2024].
- [8] “Complete Information About PrestaShop E-commerce Platform”, ProBytes Software, May 15, 2018. [Online]. Available: <https://www.probytes.net/blog/complete-information-about-prestashop/> [Accessed Jan 28, 2024].

Kratka biografija:



Milan Govedarica rođen je 26.03.1999. godine u Gacku. Završio je srednju školu „Pero Slijepčević“ u Gacku, smer opšte gimnazije. Školske 2018/2019. godine upisao je Fakultet tehničkih nauka u Novom Sadu, smer Računarstvo i automatika i uspešno završio osnovne studije školske 2021/2022. Nakon toga nastavlja svoje obrazovanje na master studijama na smeru Računarstvo i automatika.

МЕРНО-УПРАВЉАЧКИ PID СИСТЕМ СА ЈЕДНОСМЕРНИМ МОТОРОМ**MEASUREMENT AND CONTROL PID SYSTEM WITH DC MOTOR**

Милица Митровић, Факултет техничких наука, Нови Сад

Област— ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – Тема овог рада јесте контрола брзине једносмерног мотора применом управљачког алгоритма заснованог на ПИД регулатору. У експерименталном делу је као објекат управљања коришћен Graupner Speed 400 мотор. Анализом перформанси објекта управљања при задавању брзине путем микроконтролера, примећена је нестабилност очитане вредности брзине због чега је и било потребно обезбедити PID регулацију представљену у раду. Како се једносмерни мотор чијом се брзином управља понаша као систем првог реда, у раду је представљен поступак пројектовања ПИД регулатора. Сви резултати су потврђени првобитно на нивоу симулације, а потом и експериментално.

Кључне речи: PID regulacija, Labview, Arduino, DC motor, mikrokontroler

Abstract – In this paper the problem of DC motor speed regulation is discussed through usage of PID system regulation. In the experimental part, the Graupner Speed 400 engine was used as the control object. By analyzing the performance of the control object when setting the speed via the microcontroller, the instability of the read speed value was observed, which is why it was necessary to provide the PID regulation presented in the paper. Since the DC motor, the speed of which is controlled, behaves as a first-order system, the design procedure of the PI controller is presented in the paper. All results were confirmed initially at the level of simulation, and then experimentally.

Keywords: PID regulation, Labview, Arduino, DC motor, microcontroller

1. УВОД

Пројектовање управљачког алгоритма представља важну инжењерску дисциплину која има примену у свим гранама индустрије и технологије. Концепт управљања у затвореној повратној спрези представља процес у коме се посматрани систем од интереса (објекат управљања) доводи у жељено стање посредством сензора, регулатора и актуатора који образују управљачку петљу чији је задатак да надзире понашање система поређењем задате и измерене вредности управљачке величине и на основу тога врши корективне акције уколико има потребе. Оваква структура се у теорији аутоматског управљања назива основно коло система аутоматског управ-

љања [1]. Услед комплексности и широке примене управљачких алгоритама, данас су развијени многобројни алгоритми за контролу па је овај рад посвећен само једном приступу у процесу пројектовања управљачког алгоритма. Посматрани објекат управљања је једносмерни мотор, а управљачки алгоритам, заснован на ПИД регулатору, треба да омогући контролу брзине поменутог система. Поред теоријске анализе система, описана је и спрега софтверских окружења који омогућавају примену целокупне математичке анализе на конкретну хардверску поставку. На тај начин креирано је окружење које омогућава комуникацију између рачунара, микроконтролера и једносмерног мотора.

2. PID РЕГУЛАЦИЈА

ПИД регулатори представљају најзаступљенију врсту регулатора у индустријским погонима где је преко 90% управљачких петљи реализовано на основу њих [3][4]. Оваквој свеопштој популарности поменуте технике управљања доприноси једноставна структура, солидан квалитет регулације и могућност подешавања мањег броја параметара. Сам назив ПИД регулатора потиче од три дејства која овај регулатор поседује, а могу се комбиновати на различите начине. Прво дејство је пропорционално дејство (П), друго дејство је интегрално дејство (И) и треће дејство је диференцијално дејство (Д). Закон управљања ПИД регулатора се дефинише следећим изразом

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}, \quad (1)$$

где је t временска променљива, u управљачки сигнал, e сигнал грешке, а параметри регулатора

K_p , K_i и K_d представљају константу појачања пропорционалног дејства, константу појачања интегралног дејства и константу појачања диференцијалног дејства, респективно [2][5]. Услед комплексности анализе у временском домену, примењује се Лапласова трансформација као математички апарат који омогућава прелазак из континуалног временског домена у комплексни домен у коме се систем описује функцијом преноса. У складу са тим, функција преноса ПИД регулатора се дефинише као

$$G(s) = \frac{K_p s + K_i + K_d s^2}{s}, \quad (2)$$

где је s Лапласова комплексна променљива [1][6].

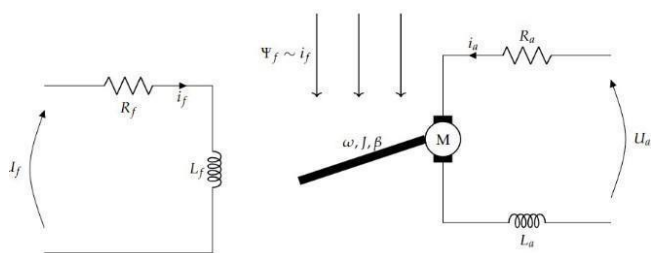
3. МОТОРИ ЈЕДНОСМЕРНЕ СТРУЈЕ

На слици 1 приказан је шематски приказ мотора једносмерне струје на основу кога се изводи математички модел система који се састоји од

Напомена:

Овај рад произтекао је из мастер рада чији ментор је био др Платон Совиљ, ред. проф.

математичких релација (једначина) које описују понашање система и омогућавају лакшу анализу његових перформанси.



Слика 1. Шематски приказ једносмерног мотора

Основни делови машине су ротор (покретни део) и статор (непокретни део). Величине које описују статорско коло у индексу имају f , док величине које описују роторско коло имају у индексу a . Принцип рада мотора једносмерне струје се заснива на електромагнетној индукцији, а комбиновањем основних једначина напонске и механичке равнотеже може се доћи до математичког модела система

$$\omega \cdot -\frac{1}{T_s} \omega + \frac{K_{sm}}{T_s} U_a, \quad y = \omega, \quad (3)$$

где је ω угаона брзина вратила машине, U_a је напон ротора, док су K_{sm} и T_s константе мотора. Прва једначина у (3) се назива једначина динамике стања, док се друга једначина у (3) назива једначина излаза. Сходно томе добијен је математички модел система првог реда. Еквивалентно као у претходном делу, применом Лапласове трансформације могуће је одредити функцију преноса једносмерног мотора у случају управљања његовом брзином

$$G(s) = \frac{K}{sT_d + 1}, \quad (4)$$

где је K статичко појачање система, а T_d је временска константа система [1][4].

3.1. ИДЕНТИФИКАЦИЈА ПАРАМЕТАРА ФУНКЦИЈЕ ПРЕНОСА ЈЕДНОСМЕРНОГ МОТОРА

Иницијални корак у поступку пројектовања управљачког алгоритма је идентификација објекта управљања, односно одређивање вредности непознатих параметара K и T_d који фигуришу у функцији преноса система (4). Данас постоје многобројни напредни алгоритми за идентификацију параметара система [7], али услед једноставности посматраног система првог реда, естимација је извршена директним задавањем улазног сигнала и мерењем одређених параметара на излазу. Поменути параметри су време успона система и вредност излазног сигнала у устаљеном стању, а на основу њих су, преко унапред познатих релација из теорије управљања, одређене вредности параметара из функције преноса система:

Табела 1. Естимирани вредности параметара функције преноса

K	T_d
2	0,37

4. ХАРДВЕРСКО ОКРУЖЕЊЕ

За имплементацију целокупног система коришћене су следеће хардверске компоненте: једносмерни мотор Graupner Speed 400, развојна плоча Arduino Uno, интегрисано коло L298N и енкодер.

4.1. МОТОР GRAUPNER SPEED 400

Мотори једносмерне струје су уређаји који претварају електричну енергију у механичку. Углавном раде на принципу електромагнетне индукције.

Главна предност једносмерних мотора у односу на оне који се напајају наизменичним величинама је лакше управљање брзином, моментом и смером обртања. DC мотори се углавном праве за мање снаге, јер би за веће снаге они морали да имају веће и јаче изворе једносмерне струје.

У овом раду коришћен је Graupner Speed 400 мотор који има веома ниску брзину обртања, али са друге стране поседује велики обртни моменат. Троши много мање струје и има знатне предности у погледу загревања па представља одличну замену за моторе са четкицама.

4.2. РАЗВОЈНА ПЛОЧА ARDUINO UNO

На Arduino Uno развојној плочи налази се микроконтролер Atmega328P. Поред тога, Arduino Uno има 14 дигиталних улазно/излазних (I/O) пинова, од којих 6 има могућност да дају PWM (енгл. *Pulse Width Modulation*) сигнал. Развојну плочу је могуће проширити додатним хардверским модулима који се непрестано развијају, како од стране компаније која их прави, тако и од самих корисника. Ако се узме у обзир да је плоча развијена као *open source* платформа, њен додатни развој од стране корисника је практично неограничен, па се и налази имплементирана у много апликација.

4.3. ИНТЕГРИСАНО КОЛО L298N

За покретање мотора једносмерне струје није довољно да се његови крајеви доведу на један од излазних пинова Arduino Uno система. Наиме, мотори једносмерне струје имају захтеве за струјом и напоном који су изван могућности микроконтролера или микрорачунара. Због тога је неопходно користити неку спољну електронику за погон и контролу мотора. Постоји неколико начина за покретање мотора једносмерне струје са излаза рачунаског уређаја. Један од њих је да применом транзистора и ово решење добро функционише под условом да не постоји потреба мењања смера обртања мотора. Други начин контроле мотора једносмерне струје је коришћење кола које се зове Х-мост. Х-мост представља структуру са таквим распоредом транзистора који омогућава контролу смера обртања и брзине мотора.

У овом раду ће бити представљен Х-мост модул заснован на L298N интегрисаном колу.

4.4. ЕНКОДЕР

Енкодер је уређај који претвара податке из једног формата у други.

У случају детекције положаја, енкодер је уређај који може детектовати и претворити механичко кретање у

аналогни или дигитални кодирани излазни сигнал. Тачније, он мери положај, док се брзина, убрзање и смер могу извести из положаја линеарним или ротационим кретањем. Класификација енкодера се врши према врсти кретања, те у том случају разликују две врсте: линеарни и ротациони. Код магнетних енкодера, линеарни енкодер је инсталиран изнад магнетне траке, и када се трака помери, енкодер чита пређени пут. Уколико се мери ротација, ротациони енкодер се окреће око своје осе. Оба се користе у електричним моторима јер прате положај ротирајуће основе и мере њен угао.

Према принципу кодирања, енкодери се деле на инкременталне и апсолутне. Мерење помоћу инкременталних енкодера се изводи бројањем импулса са енкодера, док апсолутни енкодери дају бинарни код тренутне позиције [8].

У овом раду кориштен је двофазни оптички ротациони енкодер LPD3806.

5. СОФТВЕРСКО ОКРУЖЕЊЕ

5.1. ARDUINO IDE

Arduino IDE (engl. *Integrated Development Environment*) представља софтверско окружење које је 'open-source' и намењено је за писање софтвера који се спушта на микроконтролер (Arduino Uno). Ова апликација је доступна за различите оперативне системе као што су Windows, Mac OS X и Linux, а подржава различите програмске језике као што су C и C++. Друга важна чињеница приликом адаптације софтверског са хардверским окружењем је и фреквенција импулса који се доводе на улаз микроконтролера. Наиме, ради се о сигналу велике брзине, а како је брзина којом ради микроконтролер физички ограничена његовим осцилатором, процесорско време потребно за извршавање читавог кода мора да буде што је мање могуће. Због тога се имплементира у коду интерапт рутина (engl. *Interrupt Service Routine*) која ће само узимати узлазне ивице импулсног сигнала доведеног на улаз микроконтролера. Рутина итерира бројач и то се дешава док вредност бројача не постане 100, након чега програм израчунава потребно време. На основу тог времена може се доћи до информације о угаоној брзини, а такође и до линеарне брзине којом би мотор радио. Ова брзина зависи од пречника 'точка'. У овом раду се мери пречник осовине мотора. Програм даље серијском комуникацијом шаље податке на COM (енгл. *Computer serial port*) порт рачунара брзином од 115 200 bps. Описани поступак је имплементиран у Arduino IDE софтверском окружењу и кориштен је за мерење брзине једносмерног мотора.

5.2. LabVIEW

Даља обрада података извршебна је у Labview програмском окружењу.

Labview програм је написан тако да омогући једноставан приказ података примљених од микроконтролера путем серијске комуникације. Због начина на који хардвер комуницира са Labview фронтендом програм је потребно написати каскадно из два дела, тако да први део програма иницијализује комуникацију, док други у бесконачној петљи

константно врши жељену обраду података, све док га корисник не заустави.

5.3. ИНИЦИЈАЛИЗАЦИЈА КОМУНИКАЦИЈЕ

Како је претходно напоменуто потребно је користити каскадну структуру унутар Labview-a, односно *flat sequence*. Реч је о структури која се понаша као слика унутар филма (што и објашњава њен изглед). Свака нова слика (део кода) не може да се прикаже (изврши) пре оне која јој је претходила.

Унутар прве структуре програм почиње са постављањем параметара за Visa серијску комуникацију. Прво што је потребно подесити је избор порта на коме се врши комуникација. Уколико су драјвери за уређај инсталирани на исправан начин и постоји подршка за Visa комуникацију истих, они не морају да се поново подешавају јер ће програм сам изабрати отворен Visa порт. У програму је ипак додата опција за ручно бирање порта. Следећа битна опција коју је потребно наместити је брзина комуникације на одабраном порту.

Како је претходно напоменуто да се подаци са микроконтролера (Arduino) шаљу брзином од 115.200 bps исту брзину потребно је поставити и у Labview програмском окружењу.

Последња опција која је остављена да се ручно унесе је број битова податка. У овом случају то је 8 битова што одговара броју битова које микроконтролер шаље.

Следеће на шта би се требало обратити пажња унутар кода је иницијализација рада са датотекама коју омогућава Labview. Пошто ће се прочитани подаци о брзини тренутно приказивати (у виду графика, и нумеричког индикатора) потребно их је и сачувати да би се њима касније приступило.

У ове сврхе користи се опција *Open/create/replace file* која омогућава креирање датотеке у локалној меморији рачунара. Да би се то успешно одрадило потребно је предефинисати неколико опција. Прва је путања на којој ће се чувати датотека, као и подразумевани назив саме датотеке. Поред тога, кориснику је остављена могућност додавања имена и екстензије датотеке. Због своје универзалности, CSV (*comma-separated values file*) је одабрана као екстензија.

Након креирања датотеке иницијално се исписују први подаци у њу. Исписивање се врши следећим редоследом: број мерења, брзина, време.

Ови стрингови су раздвојени зарезом, док се на последњем месту налази карактер за нови ред. Овако се постиже форматирање потребно за исправно чување CSV датотеке у којој ће бити уписана информација редног броја читавања, времена читавања као и брзине која је прочитана.

5.4. КОНТРОЛА И ОБРАДА ПОДАТАКА

Након што су извршена претходна иницијална подешавања неопходна за реализацију замишљеног програма, прелази се на следећу каскаду.

Први део унутар ње је отварање претходно креиране датотеке. Након тога, отвара се бесконачна петља која се извршава све док корисник не кликне на дугме *stop* постављено у фронт панелу. Први део програма унутар петље је приступ битовима података који се

шаљу серијсом комуникацијом. Да би се учитали, користи се *Visa read* функција. Она преузима податке као стринг након чега се они конвертују у нумерички тип података. Подаци се конвертују у нумерички тип да би се могли приказивати као нумерички индикатор и због лакшег приказа на графику.

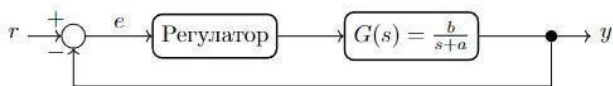
Када пристигне податак читава се и тренутно време као и број итерације понављајуће петље. Ови подаци се прослеђују путем функције *Concatenate string* по обрасцу који је претходно успостављен приликом уписа иницијалних података.

Након тога се задаје брзина мотору путем *Visa write* функције при чему се на скали одређује брзина од 0 cm/sec до 360 cm/sec и иста се конвертује у од 0 до 255 што представља фактор испуне PWM-а на излазном пину (израчунат као 2^8).

Као што је претходно напоменуто овај поступак се понавља све док га корисник не заустави притиском на дугме након чега се излази из бесконачне петље, датотека се затвара и програм се зауставља.

6. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

Као што је раније објашњено у поглављу 6.4 иницијални корак је идентификација објекта управљања, односно одређивање непознатих параметара у функцији преноса. Након завршетка идентификације, могуће је прећи на пројектовање управљачког алгорита, односно избор регулатора и одређивање његових параметара. За почетак, на слици 2 је приказана поједностављена структура основног кола система аутоматског управљања.



Слика 2. Поједностављени приказ управљачке петље

Приликом избора регулатора постоје два основна услова која треба да буду испуњена - да је систем стабилан у затвореној спрези и да има нулту грешку у устаљеном стању. Како посматрани објекат управљања представља систем првог реда без астатизма, закључује да је потребно имати бар један астатизам у директној грани регулационе петље како би се постигла нулта грешка у устаљеном стању. Добијени закључак имплицира да је неопходно изабрати регулатор са интегралним дејством (које уноси астатизам) па је сходно томе изабран најједноставнији регулатор који поседује интегрално дејство - ПИ регулатор.

Што се тиче другог услова везаног за стабилност система у негативној затвореној спрези он је уско повезан са вредношћу параметара регулатора, односно стабилност је експлицитно одређена вредношћу параметара регулатора. За рачунање вредности параметара регулатора се користе многобројне методе, а једна од њих, која ће бити приказана у наставку, је методе подешавања полова. Првобитно је потребно одредити карактеристични полином система у негативној затвореној спрези као именилац функције спрегнутог преноса (функције затвореног кола)

$$f(s) = s^2 + s(a + bKp) + bKi, \quad (5)$$

Непознати параметри регулатора Kp и Ki се могу одредити методом подешавања полова тако што се коефицијенти карактеричног полинома система изједначе са коефицијентима ткз. жељеног карактеристичног полинома који ће директно одредити понашање система у затвореној спрези. У овом раду изабрано је да сви полови система буду у тачки -2 што имплицира да је систем стабилан у затвореној спрези чиме је испуњен и други услов при пројектовању регулатора. Следи да је жељени карактеристични полином

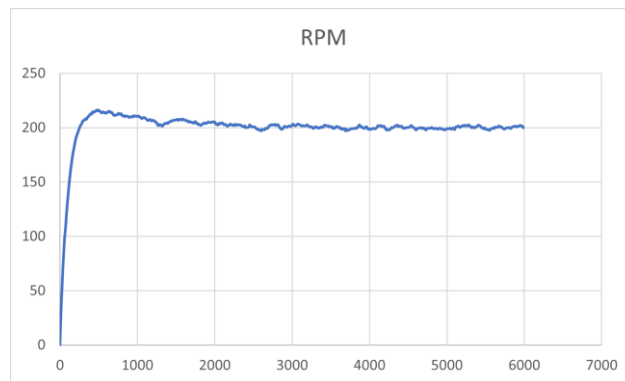
$$f_z(s) = (s + 2)^2 = s^2 + 4s + 4 \quad (6)$$

па се вредности параметара ПИ регулатора добијају изједначавањем коефицијената полинома из (5) и (6).

Kp	Ki
0,241	0,741

Табела 2. Вредности параметара регулатора

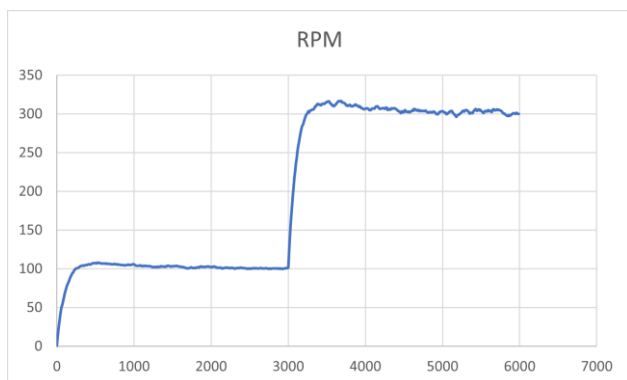
Добијене вредности су примењене на конкретан систем користећи претходно описано софтверско окружење. Извршено је више мерења, а резултати су приказани графички. За сваки експериментални поступак задата је референтна вредност брзине, док је снимана излазна вредност, односно постигнута брзина приказана на графицима у наставку. Постигнута брзина једносмерног мотора, приказана на ординатама графика у наставку, је изражена у јединицама обртаји у минути - rpm, док је време, приказано на апсисама, изражено у милисекундама- ms.



Слика 3. Графички приказ остварене брзине, референтна вредност брзине је 200 rpm

Приказани су примери одзива система на различите вредности побудног сигнала, са скоковитим променама вредности како би се анализирао брзина одзива система која је на задовољавајућем нивоу. Такође, постигнута је успешна регулација, односно праћење задате вредности брзине.

У самом одзивном сигналу су приметне благе осцилације које су последица несавршености лабораторијске хардверске поставке.



Слика 4. Графички приказ остварене брзине при скоковитој промени референтне вредности брзине са 100 rpm на 300 rpm

7. ЗАКЉУЧАК

У овом раду је приказан поступак пројектовања управљачког алгоритма чији је крајњи циљ контрола брзине обртања мотора једносмерне струје. Наиме, за посматрани објекат управљања је прво изведен математички модел у простору стања на основу чега је закључено да се приликом управљања брзином једносмерног мотора добија систем првог реда. На основу математичког модела је изведена функција преноса система у којој фигуришу два непозната параметра, статичко појачање и доминантна временска константа. Ови параметри имају и своју физичку интерпретацију, показују колико систем појачава сигнал са улаза и колико брзо систем улази у устаљено стање, респективно.

Да би се одредиле вредности ових параметара, односно да би се извршила идентификација функције преноса, спроведен је експеримент у коме је на улаз система доведен константан сигнал, а на излазу је снимљен одзив, односно његова вредност у устаљеном стању. Такође, на основу излазног сигнала је одређено и време успона на основу кога је одређена доминантна временска константа система која је даље једнозначно одредила и статичко појачање система. На тај начин је објекат управљања у потпуности идентификован.

Након потпуног одређивања функције преноса објекта управљања, уследило је пројектовање управљачке петље, односно избор регулатора и одређивање вредности параметара регулатора применом методе подешавања полова.

Структура регулатора је изабрана тако да вредност грешке у устаљеном стању на константан улазни сигнал, а са објектом управљања првог реда без астатизма, буде једнака нули. Најједноставнија структура регулатора која задовољава наведени критеријум је ПИИ регулатор. Оваква врста регулатора се састоји из пропорционалног и интегралног дејства, односно у структури регулатора фигуришу појачање пропорционалног и појачање интегралног дејства које је потребно одредити.

Непознати параметри регулатора су израчунати на основу методе подешавања полова у којој се врши директно изједначавање карактеристичног полинома система и жељеног карактеристичног полинома чији облик одређује понашање система у затвореној спреси (стабилност, брзина одзива итд.).

Добијени параметри регулатора, односно пројектовани управљачки алгоритам је примењен на објекат управљања при чему су добијени резултати снимљени и графички приказани. Упркос несавршеностима хардверске поставке, добијени резултати су задовољавајући што је потврђено и графичким резултатима.

8. ЛИТЕРАТУРА

- [1] Милан Р. Рапаић, Зоран Д. Јеличић. *Пројектовање линеарних регулатора и естиматора у простору стања*. Универзитет у Новом Саду, Факултет техничких наука, Нови Сад, 2014.
- [2] Karl J. Astrom, Bjorn Wittenmark. *Computer-Controlled Systems Theory and Design*. Dover Publications Inc. 3rd edition, New York, USA, 2011.
- [3] Милић Р. Стојић. *Системи аутоматског управљања*. 2006.
- [4] Зоран Д. Јеличић, Мирна Н. Капетина, Милан Р. Рапаић. *Дискретни управљачки системи*. Универзитет у Новом Саду, Факултет техничких наука, Нови Сад, 2023.
- [5] Karl J. Astrom, Tore Hagglund. *PID Controllers*. Instrumentation Society of America, 1995.
- [6] Милић Р. Стојић. *Дигитални системи управљања*. Грађевинска књига, Београд, 2006.
- [7] Александар Ердељан, Дарко Чапко. *Моделовање и симулација са примерима*. Универзитет у Новом Саду, Факултет техничких наука, Нови Сад, 2015.
- [8] W.Bolton, System models, Editor(s): W.Bolton, *Control Systems*, Newnes, 2002, Pages 37-70, ISBN9780750654616, <https://doi.org/10.1016/B978-075065461-6/50002-7>.

Кратка биографија:



Милица Митровић рођена је у Зворнику 1997. год. Мастер рад на Факултету техничких наука из области Електротехнике и рачунарства - Мерно-управљачки PID систем са једносмерним мотором одбранила је 2024. год.

Контакт: milica.mitroviczv@gmail.com

PAMETNE BEŽIČNE KOMUNIKACIONE TEHNOLOGIJE U SISTEMU ELEKTRIČNOG VOZILA**SMART WIRELESS COMMUNICATION TECHNOLOGIES IN ELECTRIC VEHICLE SYSTEM**

Ana Mičić, Vladimir Popović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Ovaj rad istražuje integraciju Ultra-Wideband (UWB), Near Field Communication (NFC), i Bluetooth tehnologije unutar automobilske okruženja koristeći IEEE 802.15 standard, posebno implementiran u CoSmA tehnologijama. Rad istražuje praktične primene i posledice proučavanja inkorporacije ovih bežičnih komunikacionih protokola u vozilu. Naglašavajući specifičnu implementaciju unutar CoSmA okvira, studija istražuje kako ova integracija poboljšava sigurnost, povezanost i pametne funkcije mobilnosti u automobilima, doprinoseći razvoju inteligentnih sistema za transport.

Ključne reči: komunikacioni protokoli, UWB, CoSmA, digitalni ključ;

Abstract – This paper examines the integration of Ultra-Wideband (UWB), Near Field Communication (NFC), and Bluetooth technologies within the automotive context, utilizing the IEEE 802.15 standard, particularly implemented in CoSmA technology. The research investigates the practical applications and implications of incorporating these wireless communication protocols in vehicles. Emphasizing the specific implementation within the CoSmA framework, the study explores how this integration enhances security, connectivity, and smart mobility features in automobiles, contributing to the advancement of intelligent transportation systems.

Keywords: communication protocols, UWB, CoSmA, digital Key;

1. UVOD

U savremenom društvu, postizanje besprekorne povezanosti i efikasne komunikacije između uređaja postalo je ključno za uspeh različitih tehnoloških aplikacija. Sa stalnim napretkom bežičnih komunikacionih tehnologija, istraživači i inženjeri suočavaju se s izazovom pronalazjenja optimalnih rešenja koja zadovoljavaju zahteve brze, pouzdane i sigurne komunikacije.

U tom kontekstu, protokoli kao što su Ultra-Wideband, Bluetooth i Near Field Communication se izdvajaju kao ključni akteri u digitalnom ekosistemu. UWB, s svojom sposobnošću slanja podataka širokim frekvencijskim opsegom, otvara vrata za visoko precizne i brze prenose

podataka, čime se pruža izuzetna osnova za raznovrsne primene, od lokalizacije do bežične povezanosti visoke propusnosti [1].

S druge strane, Bluetooth, dugogodišnji lider u bežičnoj tehnologiji, nastavlja evoluirati kako bi zadovoljio potrebe povezivanja uređaja u različitim scenarijima, od pametnih telefona do pametnih kuća [2].

Sa svoje strane, NFC pruža jednostavno i intuitivno povezivanje na kratkim udaljenostima, čime se često koristi u raznim aplikacijama poput mobilnog plaćanja i deljenja informacija [3].

CoSmA (engl. Continental Digital Access) tehnologija se oslanja na sinergiju ovih bežičnih protokola kako bi omogućila koordiniranu komunikaciju između vozila, infrastrukture i drugih povezanih uređaja u saobraćaju [4].

2. KOMUNIKACIONI PROTOKOLI

U nastavku se daje pregled bežičnih komunikacionih protokola koji se koriste za implementaciju digitalnog ključa vozila integrisanih u CoSmA tehnologije.

2.1 UWB tehnologija

UWB je bežična komunikaciona tehnologija koja koristi širokopojasne signale sa frekvencijskim opsegom iznad 1 GHz. Za razliku od tradicionalnih komunikacionih sistema koji koriste sinusoidalne nosioce, UWB se oslanja na uske pulsirajuće signale nesinusoidnog oblika na nivou nanosekunde za prenos podataka.

Jedna od glavnih prednosti UWB tehnologije je niska složenost sistema, kao i niska spektralna gustina snage signala pri odašiljanju. Takođe, UWB je manje osjetljiv na probleme poput gubitka signala, omogućava visoku preciznost pozicioniranja i ima veliku sposobnost prodiranja kroz prepreke.

S obzirom na čestu pojavu krađe automobila uz pomoć prenosa signala, mnogi razvojni inženjeri razmatraju koji bežični interfejs može rešiti ovaj problem. Standard IEEE 802.15.4-2020 je ažurirao fizički sloj (PHY) i MAC (engl. Media Access Control) sloj tehnologije za visoko precizne, sigurne primene merenja rastojanja.

HRP UWB sistem za merenje rastojanja se zasniva na ToF (engl. Time of Flight) i AoA (engl. Angle of Arrival) merenjima kako bi obezbedio pouzdane i čvrste vremenske oznake merenja za precizno merenje udaljenosti i smeru između uređaja [1].

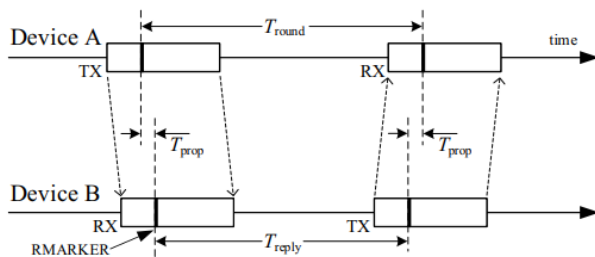
NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Vladimir Popović, doc.

Delovi procesa nazivaju se:

- **Inicijator:** Uređaj sposoban za određivanje rastojanja - koji započinje razmenu za određivanje rastojanja slanjem prvog okvira za određivanje rastojanja
- **Odgovor:** Uređaj sposoban za određivanje rastojanja - koji reaguje na inicijaciju za određivanje rastojanja putem

SS-TWR (*engl. single-sided two-way ranging*) obuhvata merenje vremena kašnjenja u oba smera za jednu poruku od jednog uređaja do drugog, s tim što drugi uređaj šalje odgovor nazad prvom. Funkcioniše kao što je prikazano na 0, gde uređaj A pokreće razmenu, a uređaj B odgovara da bi je završio. T_{prop} označava vreme propagacije tj. vreme koje signalu treba da putuje od jednog mesta do drugog, oznake za određivanje rastojanja **RMARKER**-a (*engl. ranging marker*) između uređaja.



Slika 1. Opis mehanizma za merenje rastojanja

Svaki uređaj precizno meri vremena slanja i prijema poruka, pa može izračunati vremena T_{round} i T_{reply} jednostavnim oduzimanjem. Na osnovu toga, rezultirajuće vreme leta ToF može se proceniti prema sledećoj jednačini.

$$\hat{T}_{prop} = \frac{1}{2} \cdot (T_{round} - T_{reply}) \quad (1)$$

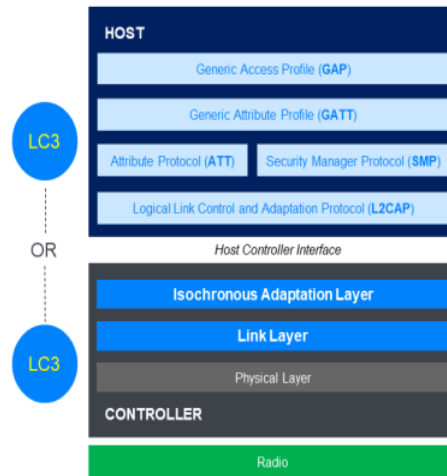
2.2 Bluetooth bežični protokol

Bluetooth Low Energy prvi put se pojavila u verziji 4.0 **Bluetooth Core Specification**. Ova varijanta podržava različite topologije komunikacije, uključujući režim emitovanja, gde jedan uređaj može prenositi podatke neograničenom broju prijemnika istovremeno.

Takođe, **Bluetooth Low Energy** je osnova za **Bluetooth mesh** mreže, koje omogućavaju stvaranje mreža sa desetinama hiljada uređaja, omogućavajući svakom uređaju da komunicira sa bilo kojim drugim uređajem u mreži. **Bluetooth Low Energy**, sa svojom ekonomičnom infrastrukturom niske potrošnje energije i svojom širokom dostupnošću u pametnim telefonima, postao je važan alat za proizvođače automobila.

Na primer, **BLE** se može koristiti kao alternativa tradicionalnim **LIN** (*engl. Local Interconnect Network*) i **CAN** (*engl. Controller area network*) mrežama, zamenjujući teške kablove bežičnom povezanošću. U hibridnim i potpuno električnim vozilima, **BLE** se može koristiti za slanje podataka o temperaturi i naponu iz baterijskih paketa glavnom računaru vozila, kao deo sistema upravljanja baterijom. **Bluetooth LE** stek sastoji se od nekoliko slojeva i funkcionalnih modula, prikazano na

slici 2. Ovi delovi steka su raspoređeni u okviru dva glavna arhitektonska bloka poznata kao domaćin (*engl. host*) i kontroler (*engl. controller*), a standardni logički interfejs definiše način na koji ove dve komponente mogu komunicirati. Domaćin je često nešto poput operativnog sistema, a kontroler je često sistem na čipu.



Slika 2. Opis Bluetooth LE Stack mehanizma

2.3 NFC komunikacioni protokol

Near Field Communication (NFC) je bežična komunikaciona tehnologija koja radi na radio talasima sa osnovnom frekvencijom od 13,56 MHz, sa tipičnim dometom do 2 cm i brzinom prenosa podataka od 46 kbit/s do 1,7 Mbit/s. NFC je tehnologija u usponu razvijena na osnovu RFID-a (*engl. Radio-frequency identification – RFID*) na način da se sastoji od interfejsa i protokola koji se zasnivaju na RFID-u. Veza između **NFC** i induktivnog sprežanja ima ključnu ulogu u strukturi bežične komunikacije koja se koristi u različitim kontekstima, posebno u aplikacijama gde je važna blizina uređaja.

Ova tehnika se oslanja na princip induktivnog sprežanja, gde se magnetna polja generišu u blizini dva provodnička kalema – jednog u inicijatoru i drugog u ciljnom uređaju (uređaju za slušanje). U osnovi, **NFC** koristi sličan princip kao transformatori, gde se magnetno polje jednog kalema koristi za indukovanje napona u drugom kalemu, omogućavajući prenos energije i podataka.

Ova komunikacija može biti aktivna, pasivna ili kombinacija oba, pri čemu **NFC** radi korišćenjem magnetskog sprežanja između uređaja. Uređaj koji može generisati svoje sopstveno radio-frekventno polje naziva se aktivni uređaj, dok se uređaj koji za prenos podataka koristi induktivno sprežanje naziva pasivnim uređajem.

3. COSMA TEHNOLOGIJA

Povećanje ponude pristupnih uređaja došlo je s razvojem **Key-as-a-Service (KaaS)**, sistemom koji se temelji na cloud tehnologiji, kao i sa **Continental Digital Access (CoSmA)**. **CoSmA** je digitalno rešenje za pristup vozilu koje se koristi putem pametnog telefona.

Bezbednost sistema za pristup vozilu zavisi od nekoliko faktora, uključujući karakteristike fizičkog sloja za bezbedno određivanje opsega, kriptografske bezbednosti

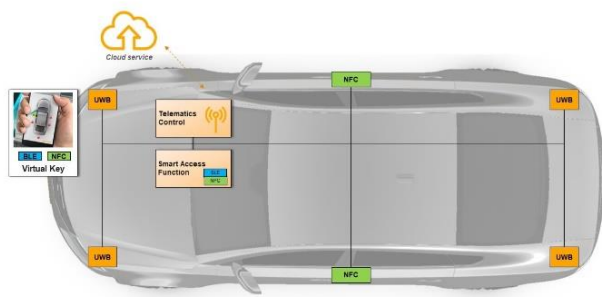
u softverskom steku i hardversku arhitekturu. Ključno je kako se ovi faktori kombinuju kako bi se efikasno odbranili od zlonamernih napada na nivou mreže ili uređaja. Postoji nekoliko načina kojima se ugrožava bezbednost sistema beskontaktnog ulaska [4]:

- *Relay* ili *man-in-the-middle* napadi, gde hakeri presreću radio talase automobila i proširuju njihov domet. Postavljaju drugi uređaj pored ključa kako bi preneli prošireni signal. To aktivira ključ, koji zatim šalje signal za otključavanje automobila. Analogni bežični signal obmanjuje vozilo da vlasnik stoji pored njega.
- Hakovanje dijagnostičkog porta na vozilu za pristup informacijama o šiframa ključa automobila - lopov može programirati novi ključ koji će pokrenuti automobil.
- Ometanje sistema - tehnika blokiranja signala koji dolazi sa ključa. Korisnik misli da je automobil zaključan, ali signal zapravo nije stigao do vozila. Lopov može otvoriti vozilo i nastaviti sa krađom

Na osnovu karakteristika različitih komunikacionih protokola u upotrebi za pristup vozilima, *UWB* je pokazao impresivne performanse za rešavanje trenutnih bezbednosnih problema.

Arhitektura obično obuhvata nekoliko komponenti i funkcionalnosti, opisano na slici 3:

- Elektronska kontrolna jedinica: Srce *CoSmA* sistema predstavlja softver koji efikasno upravlja različitim uređajima i funkcijama.
- Pametni telefoni: Ovlašćeni uređaji opremljeni *UWB* čipovima ili modulima deluju kao digitalni ključevi.
- Telematska kontrolna jedinica (TCU): U automobilske industrije je ugrađeni sistem na vozilu koji bežično povezuje vozilo sa uslugama u *cloud*-u ili drugim vozilima putem *V2X* standarda preko mobilne mreže.
- BLE/*UWB* primopredajni moduli: Predstavljaju neophodne elemente u okviru *Smart Access* sistema. Njihova uloga jeste omogućavanje komunikacije sa pametnim uređajem. Pozicioniranjem *BLE/UWB* satelita na strateškim mestima unutar vozila, sistem stiče sposobnost lociranja pametnog uređaja unutar i izvan vozila.
- NFC čitači: Uobičajeno je da se *NFC* antena integriše u ručku vrata ili A/B stub vozila.
- Usluge backend-a: *CoSmA* obično povezuje sa uslugama pozadine koje pružaju proizvođač originalne opreme. Ove usluge upravljaju upravljanjem ključevima, procesima autorizacije i autentifikacije.



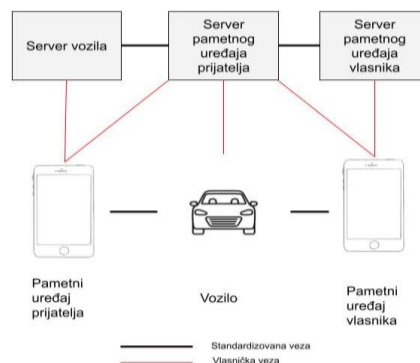
Slika 3. Arhitektura pametnog sistema pristupa [4]

4. OPIS PROTOTIPA SISTEMA ZA IMPLEMENTACIJU KONCEPTA COSMA

Sistem pristupa vozilu putem pametnog telefona (*CoSmA*), koristi pametni telefon kompatibilan sa *UWB*-om koji je programiran da funkcioniše kao ključ. Virtuelni ključ predstavlja ključnu komponentu sistema jer sadrži autorizaciju za pristup određenom vozilu. Pristup je moguć samo nakon uspešne autentifikacije putem upravljanja pravima na *cloud* sistemu. Stoga, unutar vozila su raspoređena najviše četiri *UWB* i četiri kombinovana *UWB* i *Bluetooth Low Energy* primopredajna modula. Sistem ih koristi kako bi locirao ovlašćeni pametni telefon putem *UWB*-a. Kada se autorizovani virtuelni ključ detektuje, sistem omogućava otključavanje vozila i pokretanje motora.

4.1 CCC Digitalni ključ

CCC Digital Key označava digitalni ključ koji je deo ekosistema *Car Connectivity Consortium* (CCC), [5]. To je tehnologija koja omogućava mobilnim uređajima da čuvaju, autentifikuju i dele digitalne ključeve za vozila na siguran način, čak i kada je baterija pametnog telefona slaba. *CCC Digital Key* je standardizovan kako bi bio kompatibilan sa različitim proizvođačima vozila i tipovima operativnih sistema, nudeći siguran i jednostavan način pristupa vozilima. Digitalni ključ omogućava potrošačima da lako i pouzdano koriste svoje mobilne uređaje, bez obzira na proizvođača ili tip operativnog sistema, kako bi pristupili vozilima. Pored snažnih mogućnosti i praktičnosti, nudi unapređenu sigurnost i zaštitu privatnosti.



Slika 4. Ekosistem Digitalnog ključa

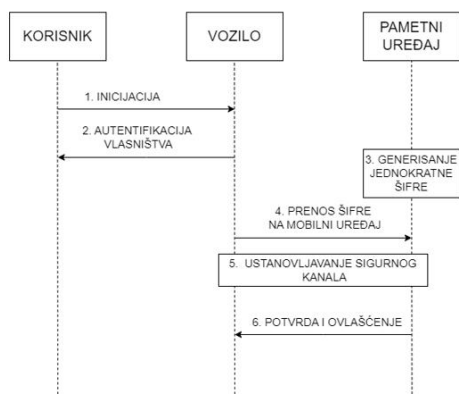
Ekosistem CCC digitalnog ključa sastoji se od:

- Vozila;
- Servera proizvođača vozila;
- Mobilnih uređaja;
- Servera mobilnih uređaja;

4.2 Uparivanje vlasnika

Vlasnik mora dokazati posedovanje vozila (metod zavisi od proizvođača automobila) i može pokrenuti proces uparivanja u aplikaciji proizvođača automobila koristeći *email* link primljen od proizvođača automobila ili iz menija vozila. U svim slučajevima, vlasnik mora predstaviti tajnu jednokratnu šifru za uparivanje *iPhone*-u, koja se koristi za generisanje sigurnog kanala uparivanja koristeći

SPAKE2+ protokol sa *NIST P-256* krivom [5]. Prilikom korišćenja aplikacije ili *email* linka, šifra se automatski prenosi na iPhone, gde se mora ručno uneti prilikom pokretanja uparivanja sa vozilom, slika 5.



Slika 5. Blok dijagram algoritma procesa uparivanja vlasnika

Standard Car Connectivity Consortium (CCC) za uparivanje vlasnika obično obuhvata sledeće faze:

1. *Inicijacija*: Vlasnik vozila pokreće proces uparivanja vlasnika, često putem aplikacije proizvođača ili specifične opcije u meniju vozila.
2. *Autentikacija vlasništva*: Korak gde se proverava ovlašćenje ili vlasništvo nad vozilom. Verifikuje da osoba koja pokreće proces uparivanja ima prava da upari mobilni uređaj sa vozilom.
3. *Generisanje jednokratne šifre za uparivanje*: Generiše se tajna jednokratna šifra za uparivanje. Ova šifra je ključni element koji se koristi za uspostavljanje sigurne veze između mobilnog uređaja i vozila.
4. *Prenos šifre na mobilni uređaj*: Generisana jednokratna šifra za uparivanje se bezbedno prenosi na mobilni uređaj.
5. *Ustanovljavanje sigurnog kanala*: Korišćenjem određenih sigurnosnih protokola definisanih CCC standardima, kao što je SPAKE2+ , mobilni uređaj i vozilo uspostavljaju siguran i enkriptovan komunikacioni kanal. Ovaj kanal osigurava poverljivost tokom interakcija.
6. *Potvrda i ovlašćenje*: Kada se uspešno uspostavi siguran kanal, vozilo potvrđuje i ovlašćuje mobilni uređaj kao 'uređaj vlasnika'. Ovo ovlašćenje daje uparenom mobilnom uređaju potpunu kontrolu i ovlašćenje nad funkcijama.

6. ZAKLJUČAK

U ovom radu detaljno su istraženi protokoli *UWB*, *BLE* i *NFC*, analizirajući njihove karakteristike, prednosti i primene. Poseban fokus je usmeren ka integraciji ovih tehnologija u okviru *CoSmA*, istražujući načine kako ova integracija transformiše automobilsku industriju. Protokol *UWB*, sa svojom sposobnošću preciznog određivanja raspona i visokog stepena propusnosti, izdvaja se kao ključni igrač u oblastima poput lokalizacije, praćenja i brze bežične povezanosti.

Integracija ovih tehnologija omogućila je precizno praćenje vozila, personalizovano povezivanje i sigurnu identifikaciju, stvarajući tako sveobuhvatan sistem koji ide iznad tradicionalnih granica automobilske sigurnosti. Ovo je posebno važno u kontekstu smanjenja krađe automobila, gde kombinacija lokacijskih informacija iz *UWB*-a, personalizovane veze putem *Bluetooth*-a i identifikacije putem *NFC*-a stvara složen sistem odbrane.

7. LITERATURA

- [1] 802.15.4z-2020 - IEEE Standard for Low-Rate Wireless Networks, Amendment 1: Enhanced Ultra Wideband (UWB) Physical Layers (PHYs) and Associated Ranging Techniques, June 2020.
- [2] The Bluetooth® Low Energy Primer, Document Version: 1.1.0, January 2023.
- [3] Naser Hossein Motlagh, Near Field Communication (NFC) - A technical Overview, Master's thesis for the degree of Master of Science in Technology, Vaasa, 28th of May 2012.
- [4] Access Systems for Security and Convenience, preuzeto sa <https://conti-engineering.com/>
- [5] Car Connectivity Consortium Digital Key Release 3, Version 0.2.6, 2021.

Kratka biografija:

Ana Mičić rođena je u Valjevu, 1996. godine. Diplomirala na Fakultetu tehničkih nauka u Novom Sadu 2020. Godine iz oblasti Elektrotehnike i računarstva – Energetska elektronika i električne mašine

Vladimir Popović rođen je u Somboru 1990. god. Doktorsku disertaciju na Fakultetu tehničkih nauka iz naučne oblasti Elektrotehnike i računarstva – odbranio je 2020. god, i u zvanju je docenta na Fakultetu tehničkih nauka. Oblast interesovanja su regulacija elektromotornih pogona i digitalna kontrola sistema automatskog upravljanja.

ODREĐIVANJE PARAMETARA ANTENE I PRILAGOĐENJE PREOSTALOM DELU KOLA**DETERMINATION OF ANTENNA PARAMETERS AND ADJUSTMENT TO THE REMAINING PART OF THE CIRCUIT**

Dražen Stojšin, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu prikazano je merenje karakteristika antene i određivanje njenih parametara da bi se obezbedilo prilagođenje sa balun kolom i preostalim delom kola. Kao primer, posmatran je uređaj sa ZigBee primopredajnikom, koji radi na frekvenciji od 2,45 GHz. Parametri antene su izmereni pomoću pomoću vektorskog analizatora mreže (VNA), a zatim podešeni korišćenjem balun kola. Elementi balun kola su određeni korišćenjem simulacionog programa SimSmith. Nakon usklađivanja balun kola razmotreni su rezultati merenja indikatora snage primljenog signala (RSSI).

Ključne reči: Antena, frekvencijski opseg antene, S-parametri, Smith dijagram, RSSI merenja.

Abstract – This paper presents the measurement of antenna characteristics and the determination of its parameters to ensure compatibility with the balun circuit and the remaining part of the circuit. As an example, a device with a ZigBee transceiver, which operates at a frequency of 2,45 GHz, was analyzed. The antenna parameters were measured using a vector network analyzer (VNA) and then adjusted using a balun circuit. The elements of the balun circuit were determined using the SimSmith simulation program. After adjusting the balun circuit, the results of the received signal strength indicator (RSSI) measurement were considered.

Keywords: Antena, frequency range, S-parameters (scattering parameters), Smith chart, RSSI measurements.

1. UVOD

Usled ekspanzije tehnologija interneta stvari i razvoja pametnih uređaja, pojavila se potreba za razvijanjem bežičnih tehnologija male potrošnje koja se mogu integrisati u minijaturna kućišta. ZigBee je bežična tehnologija razvijena kao otvoreni globalni standard koji treba da odgovori na zahteve niske cene i male potrošnje energije bežičnih M2M mreža [1]. Bazirana je na IEEE 802.15.4 standardu personalnih računarskih mreža (PAN – Personal Area Network), a radi u različitim opsezima, najčešće na 2,4 GHz, ali i na 784 MHz, 868 MHz ili 914 MHz [2]. Standard predstavlja radio protokol namenjen jeftinim uređajima koji se napajaju pomoću baterija.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji je mentor bila dr Mirjana Damjanović, red. prof.

Primena je najčešće u bežičnoj kontroli i praćenju, npr. u medicini, IoT, u pametnim mrežama (*smart energy/smart grid*), upravljanju osvetljenjem, sistemima automatizacije u zgradama, praćenju stanja u rezervoarima, medicinski uređaji itd.

Da bi se postigle što manje dimenzije uređaja, antena mora da bude što minijaturnija. Projektuje se na štampanoj ploči uređaja, da bi se izbeglo korišćenje eksterne komercijalne antene.

U ovom radu je prikazano projektovanje antene za primenu u ZigBee uređaju. Antena je upoređena sa komercijalno dostupnom antenom [3]. Izvršena je modifikacija antene i prilagodnog kola sa ciljem dobijanja maksimalne efikasnosti antene. Posmatrani su sledeći parametri antene: frekvencijski opseg, gubici, slabljenje signala i karakteristična impedansa antene [4].

2. OPIS ANTENE PROJEKTOVANE NA ŠTAMPAHOJ PLOČI

Antena je aktivni ili pasivni elektromagnetski uređaj ili deo koji u sklopu s nekim elektroničkim uređajima pretvara elektromagnetsku energiju, vezanu uz vodove ili talasovode, u prostorni elektromagnetski talas ili obratno. Dobro projektovana antena obezbeđuje optimalni domet signala. Svaka antena ima dva važna parametra: talasnu dužinu i napajanje.

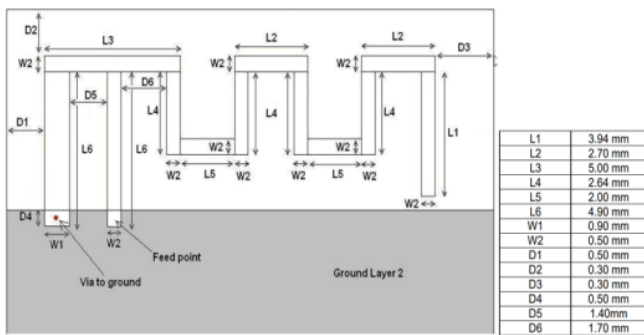
Za aplikacije na 2,4 GHz, većina antena na štampanim pločama [5] može se svrstati u nekoliko tipova: žičana antena, antena na štampanoj ploči i čip antena.

Idealna antena, ukoliko je usklađena sa ostatkom kola, zračiće celokupnu energiju bez refleksije.

Parametar koji pokazuje koliko je dobro prilagođena antena prenosnomvodu je povratni gubitak (*Return loss*), $RL = 10\log(P_{ref}/P_{inc})$, koji predstavlja odnos snage upadnog P_{inc} i reflektovanog signala P_{ref} . S_{11} parametar ima negativnu vrednost u odnosu na povratne gubitke, $S_{11} = -RL$. U većini slučajeva ukoliko su povratni gubici veći od 10 dB (odnosno, ako je $S_{11} \leq -10$ dB) smatra se da se radi o dobroj anteni [6].

Projektovana antena, koja je izrađena na štampanoj ploči (PCB), treba da radi na frekvenciji od $f = 2,45$ GHz, pa je njena talasna dužina $\lambda = v/f = 0,12$ m, gde je brzina prostiranja talas $v = 3 \cdot 10^8$ m/s, a potrebna dužina mikrostrip antene je $\lambda/4 = 30,61$ mm.

Male promene dimenzija antene mogu imati veliki uticaj na njene performanse. Kao referentna antena za postizanje optimalnih performansi koristila se antena prikazana na slici 1 [7].



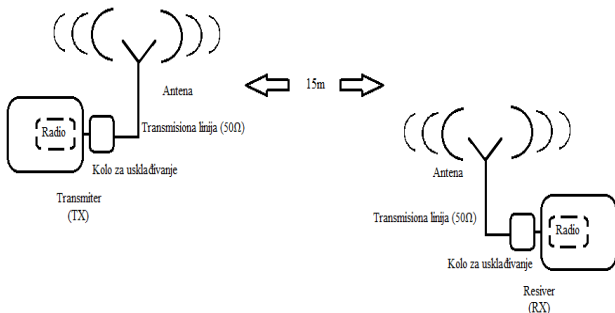
Slika 1. Dimenzije referentne antene [7]



Slika 2. Prikaz dizajna antene i izrađene štampane ploče

Izbor geometrije antene ima uticaj na zračenje antene, njen frekvencijski opseg, karakterističnu impedansu i domet. Postoji mogućnost da se usled promene geometrije parametri mogu veoma teško uskladiti ili da ih je u nekim slučajevima skoro nemoguće uskladiti sa ostatkom kola. Na slici 2 je prikazan dizajn antene kao i njen izgled nakon izrade štampane ploče.

Bežični prenos signala sa malim dometom između dva uređaja koji imaju antenu za bežični prenos ilustrativno je prikazan na slici 3.



Slika 3. Ilustrativni prikaz uobičajenog sistema bežičnog prenosa signala sa malim dometom

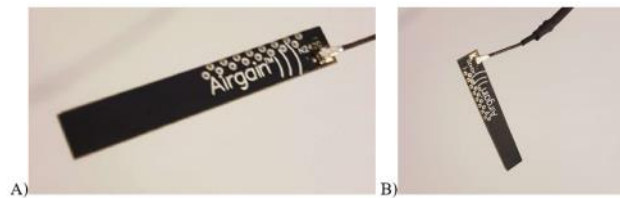
3. MERENJE PARAMETARA ANTENE

Uređaj korišćen prilikom merenja električnih karakteristika antene je vektorski analizator mreže, Rohde & Schwarz RS ZVA 50 [8]. Da bi se merne tačke priključile na VNA, napravljen je adapterski kabel, koji je izrađen od RG316 kabela sa priključenim V.FL kabelom (broj proizvoda: V.FL-2LPG-062N2D-A-(35)) [9]. Merenje je vršeno u frekvencijskom opsegu od 1,95 GHz do 2,95 GHz.

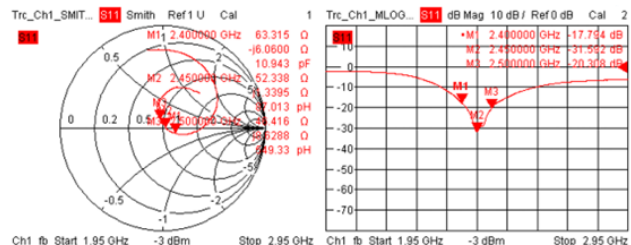
3.1. MERENJE PARAMETARA AIRGAIN N2420 ANTENE

Kao prvi korak, izvršeno je merenje komercijalno dostupne antene AIRGAIN N2420 (slika 4), koja radi na frekvenciji od 2,4 GHz. Upoređene su njene karakteristike (slika 5), sa kataloškim podacima [3].

Pri merenju parametara antene je potrebno obratiti pažnju na položaj sonde. Preporuka proizvođača je da se pristupni kabl postavi kao na slici 4a). Ukoliko se kabl postavi pod drugim uglom, može doći do greške prilikom merenja parametara antene.



(a) (b)
Slika 4. Postavka sonde za merenje parametara Airgain N2420 antene[3]: (a) ispravan položaj, (b) sonda priključena pod drugim uglom (loše)

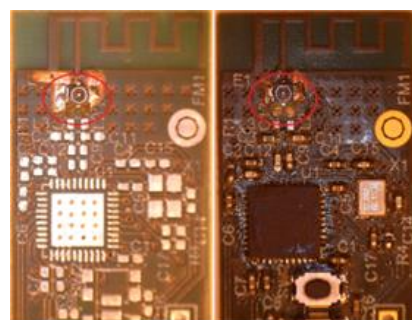


Slika 5. Prikaz merenja referentne antene N2420

Na slici 5 prikazan je Smitov dijagram (Smith) za referentnu antenu, na kojem su označene merne tačke sa karakterističnim impedansama i frekvencijama, M1 (63.315-j6.06 Ω) na 2,40 GHz, M2 (52.3+j5.3 Ω) na 2,45 GHz, M3 (48.416+j8.6 Ω) na 2,50 GHz. Tačka M2 ima karakterističnu impedansu veoma blisku 50 Ω i nalazi se na traženoj frekvenciji od 2,45 GHz. Sve tri tačke su dobro grupisane. Parametar S_{11} u tački M2 ima vrednost -30 dB. To nam pokazuje da se radi o dobro projektovanoj anteni za traženi frekvencijski opseg.

4. MERENJE PARAMETARA PROJEKTOVANE ANTENE (P119 R002)

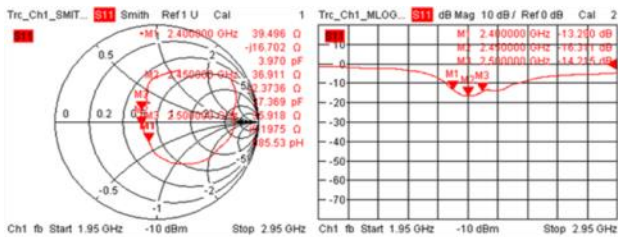
Da bi se izvršilo merenje sa što manjim uticajem okolnih komponenata na projektovanu antenu, najpre je urađeno merenje na štampanoj ploči bez dodatnih komponenti (slika 6a), a zatim i merenje nakon njihovog postavljanja (slika 6b).



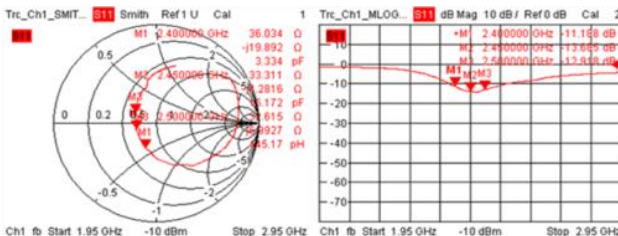
(a) (b)

Slika 6. Prikaz štampane ploče (a) bez postavljenih komponenti i (b) sa komponentama

Na slikama 7 i 8 prikazani su rezultati merenja za antenu na štampanoj ploči bez i sa zalemljenim komponentama, respektivno. U oba slučaja se može primetiti da su tačke M1, M2 i M3 grupisane, odnosno, u blizini jedna druge. Nakon sklapanja štampane ploče vrednost parametra S_{11} ostaje ispod -10 dB. Parametri u posmatranim tačkama su nešto veći, ali ne utiču na zračenje antene na željenoj impedansi od $Z_0=50 \Omega$.



Slika 7. Rezultati merenja za antenu na štampanoj ploči bez zalemljenih komponentata (slika 6a)



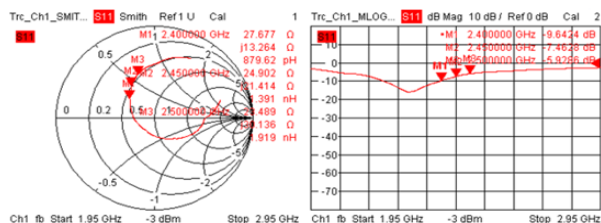
Slika 8. Rezultati merenja za antenu na štampanoj ploči sa zalemljenim komponentatama (slika 6b)

4.1 KOMPLETNO KOLO SA PROJEKTOVANOM ANTENOM UPAKOVANO U PLASTIČNO KUĆIŠTE

Nakon dodavanja plastičnog kućišta (slika 9) ponovljeno je merenje karakteristika projektovane antene. Da bi se merenje moglo obaviti, na plastičnom kućištu je izbušen mali otvor za prolaz adaptera mernog kabla. U ovom slučaju, antena nema usklađenu impedansu (približno 50 Ω) za posmatrane tačke M1, M2 i M3. Parametar S_{11} u tački M1 ima vrednost od -10 dB, dok za je za preostale dve tačke M2 i M3 njegova vrednost iznad -10 dB, kao što se može videti na slici 10. Zbog toga je bio potreban redizajn antene.



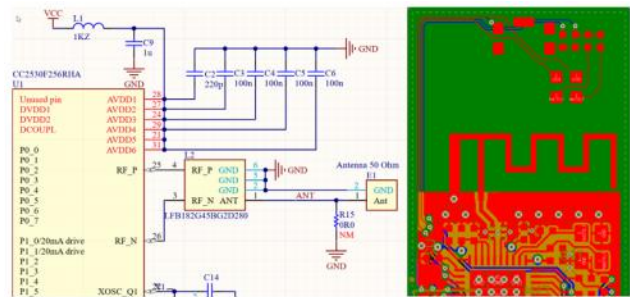
Slika 9. Kompletno kolo sa projektovanom antenom i merni kabl V.FL-2LPG-062N2D-A(35)



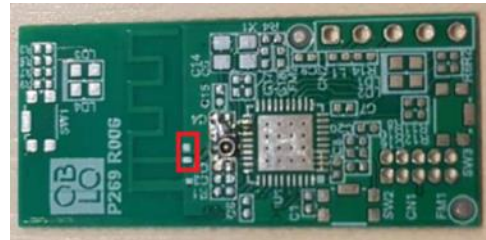
Slika 10. Rezultati merenja za kompletno kolo sa projektovanom antenom

5. REDIZAJN ANTENE (P269 R006)

Da bi poboljšali karakteristike antene, dodat je bakarni sloj na štampanoj ploči tj. sloj sa masom (GND), kao što se može videti na slici 11. Ostavljeno je jedno slobodno mesto (*place holder*) za komponentu, koja će biti dodata da se podese parametri antene na štampanoj ploči P269 R006 (slika 12).



Slika 11. Deo šeme uređaja i redizajn štampane ploče



Slika 12. Prikaz praznog mesta ostavljenog za dodavanje korekcionih komponenti na štampanoj ploči

Da bi se uradilo merenje, potrebno je predvideti pogodno mesto za postavljanje konektora za merenje parametara antene na štampanoj ploči. Mesto je tako izabrano da bude što bliže pristupnoj tački (*feed point*) antene, kao što je prikazano na slici 13. Nakon redizajna uređaja ponovljeno je merenje parametara antene.



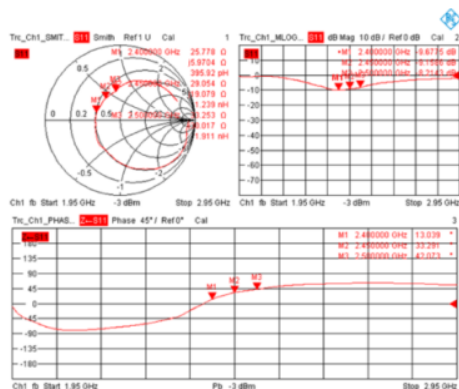
(a)

(b)

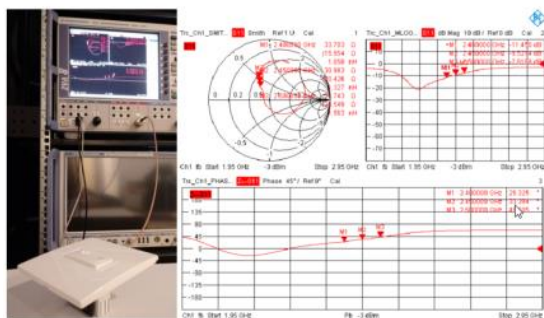
Slika 13. Prikaz štampane ploče P269 R006 (a) bez postavljenih komponenti i (b) sa komponentama

6. MERENJE PARAMETARA REDIZAJNIRANE ANTENE (P269 R006)

Izmerene karakteristike antene na štampanoj ploči P269 R006 pokazuju da se tačke M1 ($25,778 + j \cdot 5,97 \Omega$), M2 ($29,054 + j \cdot 19,079 \Omega$) i M3 ($33,253 + j \cdot 30,017 \Omega$) nalaze u odgovarajućem opsegu i da su grupisane, kao što se može videti na slici 14. Parametar S_{11} u svim tačkama ima vrednost blisku, nešto višu od -10 dB. Pošto su merne tačke grupisane, može se zaključiti da je karakteristiku takve antene jednostavno popraviti. Nakon toga, uređaj sa dodatim napajanjem je oklopljen i izvršeno je novo merenje (slika 15). Deo sa napajanjem je potrebno zalemiti na svim lemnim mestima za štampanu ploču da ne bi došlo do odstupanja u merenju. Postavljanjem konektora bliže pristupnoj tački antene dobijamo tačnije rezultate merenja. Treba napomenuti da je na projektovanoj ploči ostavljeno mesto za dodavanje korekcionih komponenti, za slučaj da je potrebno popraviti karakteristike antene.



Slika 14. Rezultati merenja redizajnirane štampane ploče P269 R006 bez dodatih komponenta



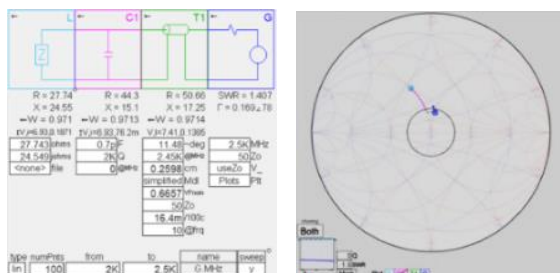
Slika 15. Rezultati merenja za redizajniranu štampanu ploču sa komponentama postavljenim u kućište, bez dodatne komponente za korigovanje parametara antene

Redizajnirana antena ima znatno bolje karakteristike od prve verzije. Markeri su skoro spojeni u jednoj tački bez obzira da li se antena nalazi u kućištu (slika 15), što znači da je takvoj anteni znatno lakše korigovati karakteristike. Čak i bez dodatne korekcionne komponente, kada se uređaj nalazi u kućištu, tačka M2 pokazuje vrednost S_{11} parametra nižu od -10 dB.

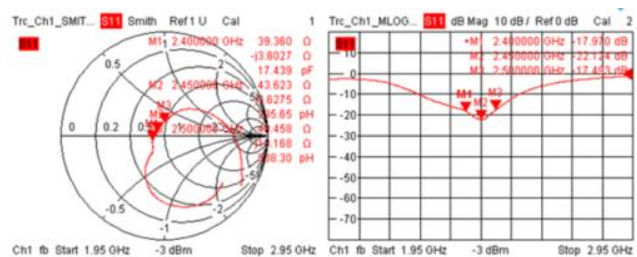
Za poboljšanje karakteristika uređaja, korišćen je programski alat SimSmith [10]. Njime je određena korekciona komponenta (slika 16), kondenzator $C_1 = 0,7$ pF, koji daje najbolje rezultate (slika 17).

Tačka M2 pokazuje vrednost S_{11} parametra nižu od -20 dB. Izmerena vrednost RSSI parametra na razdaljini od 15 m ima opseg od -73.5 dBm do -83 dBm, što je dovoljno dobro.

Domet antene je do 80 m, na toj razdaljini je primetno slabljenje signala ali uređaj još uvek radi. Zaključujemo da ovako projektovana antena odgovara zahtevima Zigbee uređaja.



Slika 16. Podešavanja parametara balun kola u SimSmith programskom alatu



Slika 17. Rezultati merenja kompletnog uređaja, nakon dodavanja korekcionog kondenzatora $C_1 = 0,7$ pF

7. ZAKLJUČAK

Cilj ovog rada je bio da se predstavi projektovanje antene, merenje njenih parametara i podešavanje karakteristika. Prikazane su dve antene. Prva projektovana antena, P119 R002, ima parametar $S_{11} = -15$ dB, što je odgovarajuće, međutim, nakon oklapanja njene karakteristike nisu bile zadovoljavajuće, $S_{11} = -9$ dB. Urađen je redizajn antene i dodavanje korekcionog kola koje je određeno pomoću simulatora SimSmith. Antena P269 R006 sa korekcionim kolom pokazuje vrednost za parametar $S_{11} = -20$ dB, čak i nakon dodavanja kućišta. Može se uočiti podudaranje merenih parametara sa proračunima koja su vršena pomoću softverskog alata na računaru. Tačke M1, M2 i M3 su bolje grupisane, poboljšan je domet antene. Bolje karakteristike antene bi se mogle postići dodavanjem korekcionog kola sa dva ili više kondenzatora, podešavanjem debljine štampane ploče (stackup), kao i podešavanjem impedanse antene što bliže 50Ω .

8. LITERATURA

- [1] Connectivity Standard Alliance: "Zigbee", <https://csa-iot.org/all-solutions/zigbee/>, pristupljeno: 1.12.2023.
- [2] Standard „IEE Std 802.15.4-2003“, ieeexplore.ieee.org/document/1237559, pristupljeno: 1.12.2023.
- [3] Airgain Inc. „Embedded Antenna model N2024“, www.arcantenna.com, pristupljeno 1.12.2023.
- [4] Agilent Technologies: „Antenna Test Selection Guide“, 2012.
- [5] Texas Instruments: „Antenna Selection Quick Guide“, www.ti.com/lit/pdf/swra351, pristupljeno 1.12.2023.
- [6] A. Đugova, J. Radić, M. Videnović-Mišić, L. Nad: „Metoda za određivanje karakteristika širokopojasnog nisko-šumnog pojačavača realizovanog u CMOS tehnologiji“, FTN Novi Sad, tehničko rešenje, www.ftn.uns.ac.rs/n4670087, pristupljeno: 1.12.2023.
- [7] A. Andersen: „Application Note AN043“, Texas Instruments, [AN043](http://www.ti.com/lit/an/an043), pristupljeno: 1.12.2023.
- [8] Hirose Electric: „Ultra-Small Surface Mount Coaxial Connectors“, www.mouser.com, 1.12.2023.
- [9] Upustvo „Vector Network Analyzer RS ZVA50“, www.testequipmenthq.com, pristupljeno: 1.12.2023.
- [10] Program „SimSmith“, www.ae6ty.com, pristupljeno: 1.12.2023.

Kratka biografija:

Dražen Stojšin rođen 1988. godine. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – na studijskoj grupi Primenjena elektronika odbranio je 2024. godine.

Kontakt: stojsin.drazen@gmail.com

ФОРЕНЗИКА РАЧУНАРСКИХ МРЕЖА**NETWORK FORENSICS**Јелена Стојаковић, *Факултет техничких наука, Нови Сад***Област – ЕЛЕКТРОТЕХНИЧКО И РАЧУНАРСКО ИНЖЕЊЕРСТВО**

Кратак садржај – Рад представља истраживање у једној од подобласти дигиталне форензике – форензици рачунарских мрежа. Циљ рада је преглед и анализа методологија и алата који се користе у процесу форензике рачунарских мрежа. Сходно рапидном напретку технологије, рачунарске мреже су погодно тло за различите видове злоупотреба. У циљу приказивања потенцијалног случаја у оквиру форензике рачунарских мрежа, спроведена је студија случаја кроз све фазе које су неопходне за доношење конкретног закључка.

Кључне речи: дигитална форензика, рачунарске мреже

Abstract – This work presents research in one of the subfields of digital forensics – computer network forensics. The aim of the work is to review and analyze the methodologies and tools used in the forensics process of computer networks. Due to the rapid advancement of technology, computer networks are a fertile ground for various types of abuse. In order to present a potential case within the forensics of computer networks, a case study was conducted through all the stages that are necessary to reach a concrete conclusion.

Keywords: digital forensics, computer networks

1. УВОД

Дигитална форензика је грана форензичке науке која се фокусира на идентификацију, прикупљање, обраду, анализу и извештавање о потенцијалним доказима који су представљени у дигиталном облику. Неопходно је да све спроведене активности користе добре форензичке технике како би се осигурало да су докази прихватљиви на суду [1].

Мрежна форензика је подобласт дигиталне форензике која се бави праћењем и анализом саобраћаја рачунарске мреже у сврху прикупљања информација, форензичких доказа или откривања упада. За разлику од неких других области дигиталне форензике, предмет мрежне форензике су обично ефемерне информације [2].

1.1. Рачунарске мреже

Рачунарска мрежа представља систем који се састоји од активних и пасивних елемената. Активни елементи

су мрежни и крајњи уређаји, док пасивне елементе представљају различите врсте каблова нпр. UTP или оптички кабови чија је сврха преношење података између рачунара. Дакле, рачунарске мреже омогућавају међусобно комуницирање више уређаја, те представљају комуникациони систем.

1.2 Врсте рачунарских мрежа

Рачунарске мреже се могу делити према више различитих критеријума. Једна од њих по подручју распрострања. По тој особини мреже се деле на:

- Local Area Network (LAN)
- Wide Area Network (WAN)
- Internet

По типу комуникационог медијума мреже се деле на:

- Жичне
- Бежичне

По распореду уређаја и веза у мрежи, мреже се деле на:

- Потпуно повезану мрежу
- Магистралу
- Звезду
- Прстен

1.3 OSI модел

OSI модел (енг. Open Systems Interconnection Model) је модел који се користи за описивање функција мрежног система. Он као такав карактерише рачунарске функције у универзални скуп правила и захтева како би подржао интероперабилност између различитих производа и софтвера [3].

Према овом моделу комуникација између рачунара се дели на седам слојева апстракције:

- | | |
|--------------------|----------------------|
| • Физички слој | • Слој сесије |
| • Слој везе | • Презентациони слој |
| • Мрежни слој | • Апликациони слој |
| • Транспортни слој | |

Сваки ниво OSI модела обавља одређени посао и комуницира са слојевима изнад и испод себе. Конкретне технологије и комуникациони протоколи могу да одговарају једном или више OSI нивоа. OSI модел пре свега може да помогне приликом изолације извора проблема.

Физички ниво је први и најнижи ниво OSI модела, бави се преносом сирових података преко мреже од физичког слоја уређаја за слање до физичког слоја уређаја за пријем.

У слоју везе, односно другом слоју OSI модела постоје директно повезани чворови који се користе за

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Стеван Гостојић, ред. проф.

обављање преноса података од чвора до чвора где се подаци пакују у такозване оквири (енг. frame).

Мрежни слој је одговоран за олакшавање преноса података између две мреже.

Транспортни слој је задужен за контролу протока и контролу грешака.

Слој сесије је одговоран за отварање и затварање комуникације између два уређаја.

Слој презентације је задужен за припрему података тако да их може користити слој апликације.

У **слоју апликације** дешава се директна комуникација са подацима корисника.

Hyper-Text Transfer Protocol (HTTP) је комуникациони протокол петог, шестог и седмог нивоа OSI модела осмишљен и дизајниран тако да преноси информације између умрежених уређаја.

Улога HTTP је комуникација између клијента и сервера. Типичан ток путем HTTP-а укључује клијентску машину која шаље захтев серверу који потом шаље одговор.

У оквиру HTTP протокола постоје HTTP захтев и HTTP одговор. Помоћу HTTP захтева веб претраживачи захтевају информације које су им потребне да би учили одређени вебсајт.

HTTP одговор је оно што веб клијенти добијају од сервера као одговор на HTTP захтев. Ови одговори пружају информације које су тражене путем HTTP захтева. Сваки HTTP одговор се састоји од статусног кода и од заглавља HTTP одговора и опционо од тела HTTP одговора.

Simple Mail Transfer Protocol (SMTP) је такође протокол петог, шестог и седмог нивоа OSI модела. Његова сврха је размена електронске поште између корисника на истим или различитим рачунарима, а поред тога служи и за:

- слање поруке једном или више прималаца
- слање текстуалног, аудио или видео садржаја
- слање поруке на мрежама ван интернета

Post Office Protocol 3 (POP3), је највише коришћени протокол за пријем електронске поште преко интернета. Овај стандардни протокол користи се за примање електронске поште са удаљеног сервера и слање локалном клијенту.

2. МЕТОДЕ ФОРЕНЗИКЕ РАЧУНАРСКИХ МРЕЖА

Процес форензике рачунарских мрежа представља истраживање дигиталних података прикупљених путем различитих мрежа или разних мрежних уређаја. Током самог процеса истраге прати се процедура корак по корак у којој истражитељи идентификују, прикупљају, чувају, прегледају и анализирају доказе.

Свака истрага у оквиру дигиталне форензике се састоји од наредних корака [4]:

2.1 Идентификација доказа

Први корак форензичке истраге, углавном укључује информације о доказима који су присутни, где се чувају и како се чувају тј. у ком су формату.

Извори доказа се могу поделити у следеће две категорије:

- Докази који се могу добити унутар мреже
- Докази који се могу добити ван мреже

2.2 Прикупљање доказа

Након што се успешно идентификују, доказе је неопходно прикупити на исправан начин помоћу одређених процедура или смерница како би остали ваљани. Ова фаза се бави обезбеђењем места које је предмет форензичке истраге од неовлашћених приступа и очувањем интегритета доказа како не би постали компромитовани и услед тога неупотребљиви.

Прикупљање доказа се сврстава у две категорије [5]:

- Прикупљање променљивих доказа
- Прикупљање непроменљивих доказа

2.3 Чување доказа

Врло је важно сачувати доказе тако да они у што већој мери остану непромењени.

Кад год се докази преносе од особе до особе, неопходно је очувати информације о такозваном ланцу доказа (енг. „chain of custody“) [6].

Ланац доказа је једна од три методе коју форензичари треба да користе како би сачували доказе пре него што започне фаза анализе. Поред ове методе, постоје још и „DriveImaging“ и „Hash Values“.

2.4 Прегледање доказа

Прикупљени подаци се класификују и групишу у одређене групе како би даљи процеси анализе били једноставнији и како би управљање самим доказима било једноставније.

Главни задатак ове фазе форензичке истраге јесте уклонити сувишне информације и неповезане податке и издвојити потенцијално најрелевантније доказе за истрагу. Такође, у оквиру ове фазе истраге потребно је повратити податке које је нападач потенцијално покушао да сакрије или камуфлира.

2.5 Анализа доказа

Главни циљ ове фазе је издвојити потенцијално релевантне податке и испитати их.

Евиденција догађаја на мрежи је најважније средство за форензичку анализу проблема у систему.

Фаза анализе се може извршити на два начина. Прва је стратегија „ухвати што можеш“ (енг. „Catch it as you can“), која подразумева снимање целокупног мрежног саобраћаја ради анализе.

Стратегија „заустави, погледај и слушај“ (енг. „Stop, look and listen“) подразумева анализу сваког пакета података који путује кроз мрежу и прикупљање само онога што се сматра сумњивим и вредним даље истраге.

Један од веома корисних извора за анализу саобраћаја на мрежи је и NAT механизам (енг. *Network Address Translation*).

3. АЛАТИ ФОРЕНЗИКЕ РАЧУНАРСКИХ МРЕЖА

Користећи алате за мрежну форензику, могуће је пратити е-пошту, открити лозинке, одредити веб странице које неко посећује, чак и шпијунирати садржај корпе за онлајн куповину. Огромна моћ ових система над данашњим мрежама чини их предметом злоупотребе.

Алати за мрежну форензику омогућавају надгледање мреже и прикупљање информација о саобраћају, али такође помажу у истрази разних врста злочина на мрежи.

Сваки алат форензике рачунарских мрежа у идеалном случају би требало да има следеће особине:

- Ефикасна обрада великих датотека за чување података о мрежи
- Екстраховање информација високог нивоа или индексирање кључних речи
- Могућност поткрепљивања сваког извученог закључка

3.1 Autopsy

Једна од карактеристика Autopsy-ја је тзв. „plug-in“ архитектура. Оваква архитектура сваком програмеру даје могућност да креира и додаје сопствене прилагођене модуле или да бира између неколико унапред направљених.

Постоје три опште групације модула:

- анализа фајлова
- модул за извештаје
- графички приказ аналитике

Неки од засебних модула које је могуће су:

- модул недавне активности,
- модул за идентификацију типа датотеке,
- модул за претрагу кључних речи и
- модул за парсирање електронске поште.

3.2 Wireshark

Овај алат је анализатор мрежних протокола. Омогућава хватање односно снимање и прегледање саобраћаја који се одвија на мрежи.

Корисне карактеристике које поседује овај алат:

- Филтрирање
- Снимање података уживо и „offline“ анализа
- Читање или писање из различитих датотека за снимање
- Декрипција

Wireshark такође омогућава идентификацију различитих врста напада на мрежу. Једна врста напада коју је могуће идентификовати помоћу овог алата је такозвано „малициозно преузимање“, помоћу ког нападач може са интернета да преузме датотеке у систем.

Друга врста напада је позната под скраћеницом „DDOS“ тј. *Distributed Denial Of Service*. У овој врсти напада нападачи ускраћују ресурсе у оквиру једног система или чак на нивоу читаве мреже.

Последња врста напада је такозвано скенирање порта. Нападаци се користе овом методом како би пронашли осетљиве уређаје и како би скенирали портове односно „пронашли отворена врата кроз која могу лако да уђу“.

3.3 Xplico

Циљ овог алата је да из интернет саобраћаја сними податке који су на нивоу апликације.

Фокус овог алата нису сами мрежни протоколи па Xplico сврставамо у групу алата за форензичку анализу мрежа, чији је фокус реконструкција података који су се слали преко самих протокола.

Xplico такође нуди могућност „извожења“ податка и информација у SQLite или MySQL базу података.

3.4 NetworkMiner

NetworkMiner је алат отвореног изворног кода који спада у групу алата за форензичку анализу мреже. Овај алат се може користити као пасивни „network sniffer“ или као алат за хватање пакета у циљу откривања оперативног система, сесија, имена хостова, отворених портова итд. без стављања било каквог саобраћаја на мрежу.

Још једна веома корисна карактеристика овог алата је то што корисник може да врши претрагу кључних речи у сачуваним или такозваним „снифованим“ подацима.

3.5 NetDetector

NetDetector континуирано бележи сав саобраћај мрежа на које је повезан. Док се саобраћај снима, овај алат у исто време анализира снимљене податке у циљу откривања аномалија у самом саобраћају и одмах даје упозорења ако се одређени прагови премаше. Ови прагови се могу програмирати од стране корисника, а кад се испуне, брзи механизми NetDetector-а обезбеђују континуирано снимање/чување и анализу тј. упозорења без пропуштања пакета, тока или протока на мрежи.

3.6 Xtractor

Овај алат је посебно дизајниран за издвајање података и информација из скоро свих врста датотека електронске поште, па тако може да се користи за екстракцију адреса електронске поште, бројева телефона и порука. Са друге стране, Xtractor може послужити за пренос електронске поште са e-mail клијента директно ка различитим провајдерима који пружају услуге везане за електронску пошту.

Кључне карактеристике овог алата су:

- Преглед електронске поште
- Чување метаподатака
- Одржавање хијерархије фолдера
- Неограничена конверзија

4. СТУДИЈА СЛУЧАЈА

Кроз студију случаја приказан је ток истраге кроз фазе које су неопходне у свакој истрази када је у питању дигитална форензика.

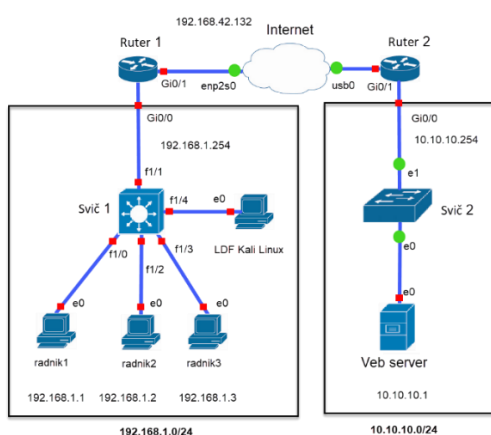
На слици 1 је представљена шема рачунарске мреже која ће бити предмет истраге.

Претпоставка је да је у десном делу представљен веб сервер неке компаније који хостује одређени сервис, а у левом делу рачунарске мреже појединачни уређаји запослених људи у другој компанији.

Циљ истраге јесте утврђивање ко је од запослених приступао веб сервису од интереса.

4.1 Идентификација доказа

На основу приложене шеме рачунарске мреже можемо да идентификујемо потенцијалне изворе доказа који ће бити од значаја за ток истраге. Кључни актери у овој истрази ће бити веб сервер и рутер који је конфигурисан у оквиру компаније.



Слика 1. Шема рачунарске мреже

4.2 Прикупљање доказа

Помоћу одређених алата неопходно је прикупити лог фајлове поменутих уређаја од значаја у оквиру рачунарске мреже на основу којих ће се даље вршити процес анализе. Најчешће компоненте у мрежи које су одговорне за NAT лог фајлове су рутери и firewall-ови. Начин на који је могуће прикупити ове лог фајлове зависи од самог уређаја, неки уређаји имају одговарајући веб интерфејс док други захтевају приступ преко командне линије.

4.3 Чување доказа

Најбитнији аспект чувања доказа је очување интегритета истих.

Смернице очувања доказа су:

- Документовање ланца доказа
- Коришћење хеш алгоритама
- Генерисање резервне копије

4.4 Прегледање доказа

У фази прегледања доказа неопходно је да се форензичари упознају са конфигурацијом мрежних уређаја. Циљ фазе прегледања доказа је заправо филтрирање свих прикупљених доказа како би се припремили само они докази који ће бити анализирани у наредној фази истраге.

4.5 Анализа доказа

Увидом у лог фајл веб сервера који хостује поменути сервис могуће је издвојити јавне IP адресе компаније са којих је приступано веб сервису. Уколико се погледа шема рачунарске мреже (слика 1) може се видети да је јавна IP адреса рутера компаније 192.168.42.132. Након увида у лог фајл веб сервера видеће се да се ова IP адреса појављује неколико пута уз различите портове.

Даље, прегледом лог фајла NAT сервиса који је конфигуриран на рутеру компаније могуће је одредити приватне IP адресе запослених људи у фирми који су приступили одређеном веб сервису. Након што су докази прикупљени, на одговарајући начин сачувани и прегледани потребно је анализирати их и донети одговарајуће закључке.

4.6 Презентација доказа

Након што су извршени сви неопходни кораци форензичке истраге почевши од идентификације

доказа завршно са анализом истих, неопходно је креирати налаз и мишљење који је даље потенцијално потребно приложити и бранити на суду.

Докази од значаја који су коришћени у оквиру истраге су:

- Шема рачунарске мреже
- Лог фајлови веб сервера
- Лог фајлови NAT сервиса конфигурираног на рутеру саме компаније

5. ЗАКЉУЧАК

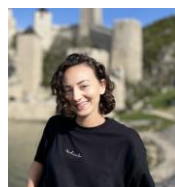
Са појавом интернета, годинама уназад, број и обим рачунарских мрежа се повећава великом брзином. Сходно томе расте и вероватноћа за потенцијална кривична дела путем рачунарских мрежа. Као одговор на такве претње јавља се потреба за константним развојем форензике рачунарских мрежа.

Мотивација овог рада јесте представљање процеса и значаја форензике рачунарских мрежа као и техника и алата у њеној примени. Изазов који потенцијално може да омета форензику рачунарских мрежа јесте чињеница да познавање начина функционисања алата и процедура које се користе у форензици, омогућавају нападачима да манипулишу потенцијалним доказима и смером истраге.

6. ЛИТЕРАТУРА

- [1] Interpol Digital forensics, доступно на <https://www.interpol.int/How-we-work/Innovation/Digital-forensics>. [посећено април 2021].
- [2] Network Forensics Encyclopedia, Science News & Research Reviews, доступно на <https://academic-accelerator.com/encyclopedia/network-forensics>. [посећено април 2023].
- [3] NetworkWorld, доступно на <https://www.networkworld.com/article/3239677/the-osi-model-explained-and-how-to-easily-remember-its-7-layers.html> [посећено април 2023]
- [4] Милана Писарић, Elektronski zapisi kao dokazi u krivičnom postupku, Правни факултет, Нови Сад, фебруар 2009
- [5] Mr. Ankit Agarwal, Ms Megha Gupta, Mr Saurabh Gupta, Prof Subhash Chandra Gupta, „Systematic Digital Forensics Investigation Model“
- [6] R. C. Joshi, Emmanuel S. Pilli, Fundamentals of Network Forensics

Кратка биографија:



Јелена Стојаковић рођена је 1997. године у Суботици.

Мастер рад на Факултету техничких наука из области Електротехнике и рачунарства – Форензика рачунарских мрежа одбранила је 2024. године.

контакт: jekibp@gmail.com

OPTIMIZACIJA UPOTREBE RESURSA U REALIZACIJI EMBEDDED SISTEMA ZA DALJINSKO UPRAVLJANJE I MONITORING**RESOURCE OPTIMIZATION IN THE IMPLEMENTATION OF EMBEDDED SYSTEM FOR REMOTE CONTROL AND MONITORING**

Jelena Marinković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom članku opisana je realizacija mehanizma za optimizovano logovanje na primeru embedded sistema. Ideja rada jeste da se zbog ograničenih resursa kod embedded uređaja, smanji veličina log poruka, na način da se umesto teksta poruke čuva samo njen identifikacioni broj. Uz pretpostavku da postoji datoteka koja sadrži parove broj poruke - tekst poruke, uz pomoć parsera moguće je rekonstruisati sadržaj log-a. Opisani log mehanizam je implementiran na embedded sistemu za čuvanje eksponata u muzeju. Logovi se po potrebi sa uređaja šalju na veb server, koji ih dekoduje, smešta u bazu podataka, i prikazuje potrebne informacije na veb stranici.

Ključne reči: *Embedded, Python, MySQL, JavaScript, Node.js*

Abstract – This article describes the implementation of an optimized logging mechanism for embedded systems. The idea of the project is to reduce the size of log messages, by logging only message identification number instead of message text. This is important because embedded devices have limited resources, especially memory. Assuming that there is a file containing pairs of message number - message text, it is possible, by using a log parser, to decode the content of the log. Log mechanism is implemented on the embedded system for exhibit protection. Logs are sent from the embedded device to the web server. The web server decodes them, stores data to a database, and displays information from the embedded system on the web page.

Keywords: *Embedded systems, Python, MySQL, JavaScript, Node.js*

1. UVOD

Ugrađeni sistemi (eng. *embedded systems*) predstavljaju spoj hardvera i softvera. Kao što im ime kaže, najčešće su ugrađeni u neki veći sistem, u okviru kog imaju jasno definisanu namenu. Prilikom razvoja embedded uređaja posebna pažnja se posvećuje optimizaciji, sa ciljem da se smanji veličina i cena proizvoda, a povećaju pouzdanost i performanse.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Predrag Teodorović, docent.

Zbog masovne proizvodnje, mala ušteda u pogledu potrebnih resursa može imati veliki uticaj na ekonomsku isplativost.

Embedded sisteme karakteriše *cross-platform* razvoj softvera. To znači da se softver razvija na jednom uređaju (*host*), a izvršava na drugom (*target*). Pored mnogih prednosti koje takva metoda donosi, javlja se i problem kako uspešno testirati sistem na konkretnoj (*target*) platformi. Kako nije svaki uređaj tokom rada moguće povezati na *host* mašinu i posmatrati tok izvršavanja, potrebno je imati mehanizam koji će omogućiti sistemu da beležiti i čuva događaje. Osim u svrhu testiranja, ovaj mehanizam bi bio koristan i za nadgledanje rada sistema pri uobičajenoj upotrebi.

Ideja rada je da se obezbedi efikasan mehanizam koji bi omogućio čuvanje potrebnog broja poruka bez da se naruše performanse sistema. Cilj je pre svega da se postigne ušteda memorije, na način da se umesto teksta poruke, čuva samo njen identifikacioni broj. Mehanizam za logovanje je implementiran na embedded sistemu za čuvanje eksponata u muzeju. Za prikaz informacija, parsiranje poruka i komunikaciju sa bazom podataka koristi se veb server.

2. LOGOVANJE

U ovom poglavlju će biti predstavljeno na koji način je izvršena optimizacija logovanja i koje je to rezultate donelo.

2.1. Sadržaj log poruke

Da bi prenela sve potrebne informacije, jedna log poruka treba da sadrži sledeće podatke:

- Poruku
- Naziv modula koji je poruku poslao
- Važnost poruke (eng. *log level*)
- Vreme nastanka poruke

Analiziranjem logova primećeno je da kada se poruka čuva, u većini slučajeva najveći deo njenog sadržaja ostaje nepromenjen. To dovodi do zaključka da je umesto skladištenja cele poruke dovoljno čuvati samo njen identifikator i onaj deo koji je podlozan promenama. Na osnovu te dve informacije moguće je rekonstruisati log poruku, uz pretpostavku da postoji nezavisna datoteka koja povezuje identifikator poruke sa njenim tekstom.



2.2. Implementacija logovanja

U slučaju da funkcija ne prima parametre imala bi sledeći izgled: *LOGE 0(ui16 module, ui16 id)*. Ista funkcija sa

2.3. Dekoder log poruka

2.4. Upis u bazu podataka

Slika 2. Prikaz tabele $\text{esp} \log t$ u koju su smeštene poruke

2.5. Zauzeće memorije

```
Used Flash size : 727220 bytes
    .text      : 602283 bytes
    .rodata    : 124681 bytes
Total image size: 831174 bytes (.bin may be padded larger)
```

Slika 3. Zauezetost flash-a u slučaju tekstualnih poruka

```

Used Flash size : 726116 bytes
.text          : 602171 bytes
.rodata        : 123689 bytes
Total image size: 830070 bytes (.bin may be padded larger)

```

Slika 4. Zauzetost flash-a pri optimizovanom logovanju

Poređenje je vršeno nakon implementacije tridesetak log poruka. Binarni kod aplikacije nalazi se u .text delu (IROM), dok se u .rodata nalaze konstantni podaci koji ne mogu da se modifikuju (DROM). U slučaju optimizovanog logovanja, kao posledica skladištenja manjeg broja informacija, smanjilo se zauzeće oba dela flash-a, za ukupno 1104 bajta. Deljenjem dobijene razlike sa ukupnim brojem implementiranih logova, dolazi se do podatka da se po jednoj poruci uštedi oko 37 bajtova. Ušteda u okviru jednog loga znatno varira od dužine poruke koja se čuva, što je poruka duža, ušteda je veća. Kako sistem postaje kompleksniji, očekuje se povećanje broja logova, samim tim i ušteda koju ovaj mehanizam pruža postaje značajnija.

3. EMBEDDED PROJEKAT

U ovom poglavlju predstavljen je realizovani embedded projekat i korišćeni hardver. Projekat je podeljen na dve celine, informativni i bezbednosni deo sistema.

3.1. Informativni deo sistema

Zadatak informativnog dela sistema je pružanje podataka o tome da li su uslovi u kojima se eksponat čuva zadovoljeni. Posmatra se temperatura i pritisak u okolini objekta, kao i da li posetioci poštuju minimalnu dozvoljenu distancu od objekta.

ESP32

Sistem je realizovan uz pomoć mikrokontrolera ESP32. Za razvoj je korišćena platforma Espressif SDK, zvanično razvojno okruženje za datu porodicu mikrokontrolera. Primeri koje okruženje nudi preuzeti su sa zvaničnog GitHub naloga [2], i uz određene izmene i adaptacije se koriste u projektu. Zahvaljujući FreeRTOS operativnom sistemu, zadaci koje mikrokontroler treba da vrši podeljeni su na taskove (eng. tasks). Pored mikrokontrolera u projektu se koriste i senzori za udaljenost, senzor za temperaturu i pritisak i alarm. Periferije su povezane putem GPIO pinova.

Ultrazvučni senzor razdaljine i alarm

Za merenje rastojanja do eksponata koristi se ultrazvučni senzor HC-SR04. Ukoliko je izmerena distanca manja od 30 cm, aktivira se alarm, kada se osoba udalji, alarm se sam deaktivira. Na server se šalje samo podatak o ukupnom broju aktivacija alarma. Alarm MH-FMD aktivira se na nizak logički nivo, a deaktivira na visok.

Senzor temperature i pritiska

Praćenje atmosferskih uslova vrši se pomoću digitalnog senzora BMP280. Odabran je da se koristi jer je malih dimenzija, velike preciznosti i male potrošnje. Sa mikrokontrolerom komunicira putem I2C protokola. Merenje se vrši jednom u minutu, a podaci se šalju na server.

3.1. Bezbednosni deo sistema

Bezbednosni deo sistema zadužen je da detektuje namerne prekršaje, pri kojima je bezbednost eksponata ugrožena. Za realizaciju se koristi kamera i senzor pokreta.

ESP32-CAM

ESP32-CAM je pločica koja sadrži mikrokontroler ESP32-S i kameru. Kamera OV2640 ima rezoluciju 2 megapiksela, malih je dimenzija i nije integrisana već je naknadno spojena sa pločicom. ESP32-CAM-MB adapter služi da putem USB porta poveže pločicu sa računarom odnosno napajanjem.

Senzor pokreta

Pasivni infracrveni senzor HC-SR501 koristi se za detekciju pokreta. Rad mu se zasniva na beleženju infracrvenih talasa. Pokret se detekuje čitanjem nivoa GPIO pina. Ukoliko se očitava nizak logički nivo, smatra se da do pokreta nije došlo, u suprotnom pokret je detektovan, na osnovu čega ESP32-CAM zna da treba da napravi fotografiju i prosledi je veb serveru.

4. VEB SERVER

U poglavlju je reč o realizaciji veb servera, njegovim ulogama i potrebnim alatima. Poglavlje je podeljeno na dve celine, back-end i front-end deo, to jest serversku i klijentsku stranu veb servera.

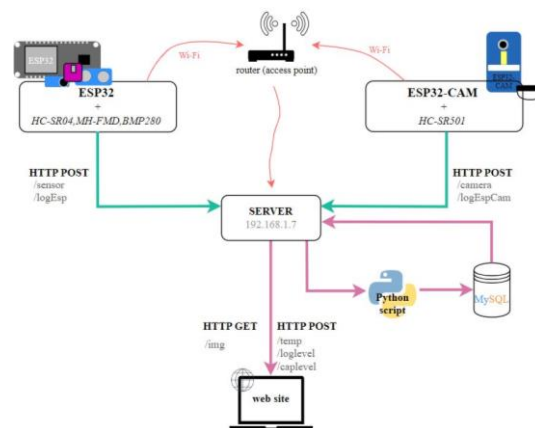
4.1. Back-end

Back-end se odnosi na delove programa koji omogućavaju rad servera, to jest implementira ponašanje veb stranice. Sastoji se od veb servera i baze podataka. Najvažnije funkcionalnosti koje implementira su: pristup podacima o aplikaciji, upravljanje bazom podataka, bezbednost, autentifikacija, autorizacija i tako dalje. Za potrebe rada koristi se domen lokalne mašine (localhost adresa), stoga je domenu moguće pristupiti samo u okviru lokalne mreže (eng. local area network, LAN).

Back-end je realizovan koristeći funkcije koje pruža Express.js. Express.js je minimalan i fleksibilan Node.js framework koji pruža širok spektar funkcija za razvoj aplikacija, te omogućava lako i jednostavno kreiranje API-ja (eng. Application Programming Interface). Node.js je cross-platform runtime environment za izvršavanje JavaScript koda van pretraživača [3].

Izgled back-end dela veb sajta

Server ima ulogu da prihvata i obrađuje zahteve sa ESP32 i ESP32-CAM. Podaci se na server šalju putem HTTP post zahteva. Na slici 5. šematski prikaz veb servera.



Slika 5. Šematski prikaz veb servera

Pored prihvatanja podataka sa uređaja, server ima ulogu i da log poruke parsira, pomoću Python skripte, smesti

podatke u bazu (MySQL) i po potrebi dostavlja informacije *front-end* delu veb sajta.

4.2. Front-end

Front-end označava razvoj korisničkog interfejsa. To uključuje i dizajn i funkcionalnost interfejsa. *Front-end* naziva se još i korisnička strana jer predstavlja deo veb sajta sa kojim korisnik stupa u interakciju. U *front-end* jezike spadaju, *HTML*, *CSS* i *JavaScript*.

Front-end jezici

HTML (eng. *Hypertext Markup Language*) jezik namenjen je opisu veb stranice, definiše njen sadržaj i strukturu. Veb pregledač (eng. *web browser*) preuzima *HTML* dokumente sa servera i prikazuje ih na sajtu u multimedijalnoj formi [4]. Strukturu dokumenta čine elementi, koji su označeni tagovima. U projektu je korišćen *PUG template engine* za *Node.js*, koji ima pojednostavljenu sintaksu u odnosu na *HTML*.

CSS (eng. *Cascading Style Sheet*) je jezik za formatiranje kojim se definiše izgled veb stranice. Određuje stil u kom se *HTML* elementi prikazuju.

JavaScript je skriptni, imperativni programski jezik visokog nivoa. Služi za komunikaciju sa *back-end* stranom sajta, reagovanje na događaje, proveru podataka pre slanja serveru, modifikaciju i čitanje sadržaja *HTML* elementa i tako dalje.

Izgled front-end dela veb sajta

Na slici 6. prikazan je deo veb stranice, zadužen za prikaz logova, izbor log nivoa i podešavanja kapaciteta bafera za skladištenje poruka.



Slika 6. Veb stranica log deo

Bira se jedan od šest log nivoa (*none*, *error*, *warning*, *info*, *debug*, *verbose*) i jedan od četiri moguća kapaciteta (256, 512, 1024, 2048 bajtova). Podatak o promeni se pritiskom na odgovarajuće dugme šalje na *back-end*. Pritiskom na dugme *Show log file* ispisuju se log poruke iz baze podataka, posredstvom serverske strane.



Slika 7. Veb stranica deo vezan na kameru

Na slici 7. prikazana je tabela sa veb stranice koja sadrži informacije o fotografijama koje su dobijene sa *ESP32-CAM* modula. Pritiskom na dugme *view* ispod tabele prikaže se izabrana slika.

5. ZAKLJUČAK

Cilj rada bio je implementacija mehanizma za optimizovano logovanje i njegovo testiranje na konkretnom *embeded* sistemu. Ideja je bila unaprediti log mehanizam, tako da log poruke zauzmu što manje mesta u memoriji. Tokom testiranja funkcija i celokupnog sistema, kao ni tokom korišćenja veb servera nije se naišlo na probleme ili greške.

U nekim budućim nadogradnjama, bilo bi interesantno da se umesto ručnog dodavanja jedinstvenog broja poruke koriste *hash* funkcije, što bi podrazumevalo prelazak sa programskog jezika C na C++. Ukoliko bi *hash* funkcija bila *constexp* tipa, računanje *string hash-a* bilo bi moguće uraditi u toku kompajliranja. U tom slučaju *string hash* vrednost bi predstavljala identifikacioni broj poruke [5].

Takođe, moguće je nadograditi funkcije za logovanje tako da se osim na veb server podaci loguju i na *SD* karticu. Ovakvo rešenje bilo bi dobro kod uređaja koji nemaju pristup internetu.

Server se može unaprediti dodavanjem više funkcija za upravljanje logovima, slikama i drugim podacima.

Nadogradnja *embeded* sistema moguća je u pogledu dodavanja većeg broja senzora za kontrolu atmosferskih uslova ili senzora za detekciju nekih spoljašnjih faktora poput požara.

6. LITERATURA

[1] <https://www.ascii-code.com/>, pregledano januar 2024.

[2] <https://github.com/espressif/esp-idf>, pregledano januar 2024.

[3] Shah, Dhruvi . *Node.JS Guidebook*. BPB Publications, 2018.

[4] <https://www.elektronika.ftn.uns.ac.rs/umrezeni-embeded-sistemi/wp-content/uploads/sites/176/2018/03/UES-04-Front-end-i-Back-end.pdf>, pregledano januar 2024.

[5] https://www.youtube.com/watch?v=Dt0vx-7e_B0, pregledano januar 2024

Kratka biografija:



Jelena Marinković rođena je 1997. godine. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnika i računarstvo – *Embeded* sistemi i algoritmi odbranila je 2020. god.

kontakt: jelenamarpet@gmail.com

**IMPLEMENTACIJA I VERIFIKACIJA SISTEMA ZA PROCESIRANJE PAKETA
PRIMENOM FORMALNIH METODA****IMPLEMENTATION AND VERIFICATION OF PACKET PROCESSING SYSTEM
UTILIZING FORMAL METHODS**

Aleksa Đoković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je predstavljena implementacija sistema za procesiranje paketa koja podrazumeva izgradnju paketa na osnovu ulaznih podataka i konfiguracije, i parsiranje dolaznih paketa kako bi se ekstrahovale potrebne informacije i utvrdilo postojanje grešaka prilikom prenosa. Sistem je verifikovan FPV formalnom tehnikom (eng. *Formal property verification*), dok je otpornost na greške testirana primenom FSV formalne tehnike (eng. *Formal Safety Verification*).

Ključne reči: procesiranje paketa, formalna verifikacija, FPV, FSV, Hemingov kod, otpornost na greške

Abstract – The paper presents the implementation of the packet processing system, which includes the construction of packets based on input data and configuration, and the parsing of incoming packets in order to determine the existence of errors during transmission. The system was verified using the FPV formal technique (*Formal property verification*), while the fault tolerance was tested using the FSV formal technique (*Formal Safety Verification*).

Keywords: packet processing, formal verification, FPV, FSV, Hamming code, fault tolerance

1. UVOD

Kompleksni digitalni sistemi prožimaju skoro svaki aspekt modernog života. Ispravno funkcionisanje često se smatra podrazumevanim, iako predstavlja jedan od najtežih zadataka prilikom razvoja i fabrikacije. Posledice postojanja kvarova mogu biti katastrofalne, i zbog toga uspešna verifikacija digitalnih sistema ima ključnu ulogu u proizvodnji.

Tradicionalne metode verifikacije, poput simulacije, uključuju formiranje testbenč okruženja i pisanje specifičnih ili randomizovanih testova, a kasnije temeljnu analizu i popravku uočenih kvarova, kako bi se postigla funkcionalnost sistema. Međutim, ovakav pristup ima puno nedostataka. Pre svega, formiranje kvalitetnog okruženja iziskuje veliku količinu vremena, gde postoji verovatnoća da će se veliki broj kvarova nalaziti baš unutar testbenča. Formalne metode pružaju drugačiji pristup, verifikaciju na osnovu izlaza sistema. To podrazumeva verifikaciju određenih osobina, za koje formalni alat generiše test vektor kako bi dokazao ili opovrgnuo mogućnost sistema

da ispuni definisane osobine. Formalni alat matematički analizira potpun prostor svih mogućih simulacionih puteva. Zbog toga, ima sposobnost da dostigne bilo koje dostižno stanje sistema, a zatim generiše dugačku i kompleksnu sekvencu na ulazima sistema, za koju bi inače bili potrebni sati razvoja jednako efikasnog testbenča u verifikaciji pomoću simulacije [1].

Osnovni principi formalne metodologije detaljnije su opisani u sekciji 2. U sekciji 3, predstavljena je specifikacija i struktura implementiranog sistema. Sekcija 4 posvećena je detaljima projektovanja sistema za procesiranje paketa, dok je u sekciji 5 predstavljen postupak formalne verifikacije, kao i još jedne formalne tehnike koja dokazuje otpornost sistema na greške. Zatim, u sekciji 6, iznete su ideje za unapređenje sistema. Zaključno sa sekcijom 7, dat je konačan pregled svega urađenog.

2. FORMALNE METODE

Formalne metode predstavljaju efikasniji i sveobuhvatniji pristup, koji omogućava da se određeni aspekti sistema matematički istraže i dokažu. Neke od glavnih prednosti su [1]:

- Formalni alati analiziraju mogući prostor stanja sistema, odnosno uzimaju u obzir sve simulacione puteve odjednom.
- Pružaju potpunu pokrivenost, bez potrebe za kreiranjem kompleksnih verifikacionih okruženja i test vektora.
- Obezbeđuju kratke primere ili kontraprimere ponašanja sistema, koji u velikoj meri olakšavaju proces debugovanja.
- Bilo koje stanje sistema, koje direktno nije zabranjeno ograničenjima, smatra se dostupnim i stoga alat može vrlo lako otkriti interesantne ugaone slučajeve.
- Suštinska razlika u odnosu na simulaciju predstavlja analizu na osnovu internih stanja i izlaza sistema, gde nije potrebno specificirati ulaze kako bi sistem prešao u stanje od interesa. Uz postavljena ograničenja na ulazima alat će istražiti prostor stanja i pronaći najkraći simulacioni put koji pokazuje zahtevano ponašanje ili dokazati da to nije moguće.
- Analiza osobina sistema koje ne moraju biti vremenski ograničene, tj postavljanje pitanja da li je određeno ponašanje moguće bez specificiranja vremenskog ograničenja pojavljivanja.

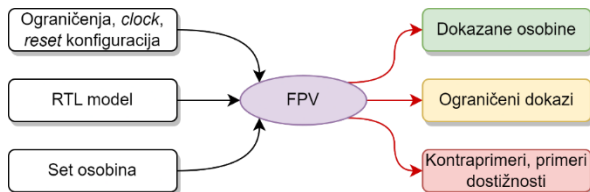
NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Vuk Vranjković, vanr. prof.

2.1. FPV TEHNIKA

FPV (eng. *Formal Property Verification*) je tehnika pomoću koje se dokazuje da određeni set osobina (eng. *Property*) važi za neki digitalni sistem. Suštinski to je potpuno drugačiji pristup od simulacije jer omogućava specifikaciju određenih vrednosti na izlazu sistema gde alat praktično trenutno generiše test vektor koji zadovoljava te uslove [1]. Ulazi i izlazi FPV modela prikazani su na slici 1. FPV matematički analizira sve moguće puteve izvršavanja za sistem koji se verifikuje unutar dozvoljenog prostora stanja. U procesu dokazivanja, nad osobinama se izvršavaju verifikacione radnje koje se dele na [1]:

- **tvrdnje** (eng. *assert*) – očekuje se da uvek važi, definišu određena stanja sistema kao nelegalna
- **tačke pokrivenosti** (eng. *cover*) – neophodno je da postoji bar jedan primer, definiše stanje sistema koje mora biti dostižno
- **pretpostavke** (eng. *assume*) – ograničenje koje isključuje određena stanja iz skupa svih dostupnih stanja

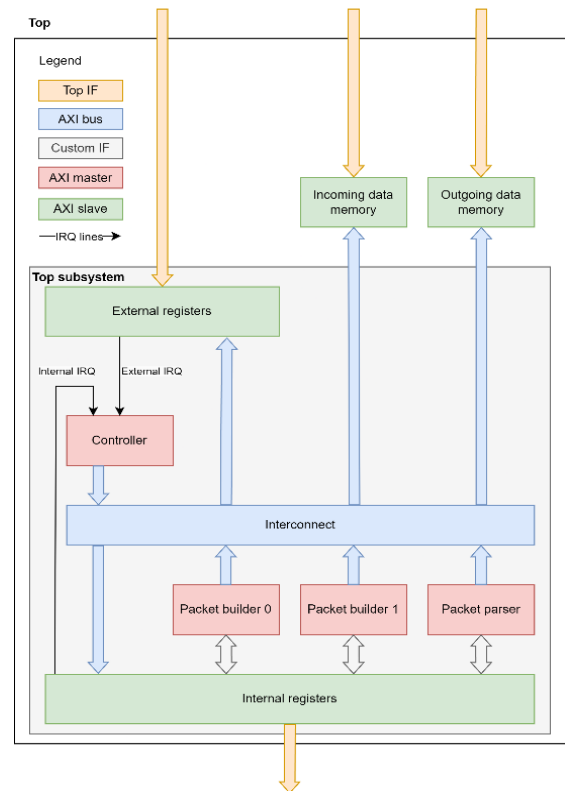


Slika 1. Ulazi i izlazi FPV alata

3 SPECIFIKACIJA SISTEMA

Osnovna namena sistema je izgradnja paketa na osnovu podataka koji pristižu u memoriju ili parsiranje paketa kako bi se izvukle potrebne informacije i eventualno detektovale greške tokom prenosa. Komunikacija svih komponenti u sistemu se bazira na AXI4 protokolu. Memorije su standardne komponente sa dva porta, širina podataka je 32 bita. Podržavaju *burst* (prenos podataka u blokovima) upis i čitanje, kao i bajt pristup. Glavni interfejs sistema čine interfejsi prema eksternim i internim (samo za čitanje) registrima i memorijama. Struktura podataka u sistemu prati *little-endian* šemu. Sistem se sastoji iz sledećih blokova (slika 2):

- Kontroler – obavlja konfiguraciju sistema, pokreće obradu paketa. AXI4 *Lite* interfejs.
- AXI4 interkonekt – povezuje memorijski mapirane master i slejv komponente.
- Memorija iz koje se čitaju podaci za obradu. (eng. *Incoming data memory*)
- Memorija u koju se upisuju formirani paketi. (eng. *Outgoing data memory*)
- Dva bloka za izgradnju paketa (eng. *Packet builder*)
- Blok za parsiranje paketa (eng. *Packet parser*)
- Interni registri – sadrže konfiguraciju trenutnog procesa obrade paketa.
- Eksterni registri – sadrže konfiguraciju trenutnog procesa obrade, koju kontroler treba da prepíše u interne registre.



Slika 2. Struktura sistema

3.1. MODOVI RADA

Sistem poseduje dva moda rada: izgradnja paketa (PB, eng. *Packet Building*) i parsiranje paketa (PP, eng. *Packet Parsing*). Za svaki od modova postoje eksterni prekidi (slika 2) koji pokreću izgradnju ili parsiranje. Signal *ext_irq[0]* indikuje da su podaci spremni u memoriji za izgradnju paketa, dok signal *ext_irq[1]* indikuje da je paket spreman za parsiranje. Ako se zahtevani zadatak ne može ispuniti, jer su blokovi zauzeti, kontroler šalje signal ka eksternim registrima, koji inkrementuje brojač o otkazanim procesima obrade, i završava zadatak.

Nakon dobijanja zahteva za pokretanjem određenog zadatka kontroler čita konfiguraciju iz eksternih registara, zatim upisuje u interne registre, pokreće zadatak i uklanja eksterni prekid. Odgovarajući blok za obradu započinje svoj proces u skladu sa definisanom konfiguracijom unutar internih registara. Izgrađeni paket se upisuje u memoriju u slučaju procesa za izgradnju paketa, dok u slučaju procesa za parsiranjem, ekstrahovane informacije paketa (podaci iz zaglavlja paketa, ECC i CRC greške) se upisuju u odgovarajuće interne registre. Svaki blok za obradu, na kraju, podiže signal prekida (*int_irq[2:0]* signal), kako bi označio kraj obrade. Kontroler uklanja interni prekid pre pokretanja novog procesa, čime se ciklus rada sistema završava.

3.2. STRUKTURA PAKETA

Svaki paket u sistemu se sastoji iz tri polja (slika 4), zaglavlje paketa, podaci i CRC8 kod. Zaglavlje paketa (slika 3) se sastoji od polja: **SOP** [15:13] – Indikator početka paketa, **P bit** [12:12] – dodatni MSB bit ECC koda, neophodan za detektovanje dvostrukih grešaka, **Tip paketa** [11:8] – može uzeti bilo koju vrednost, **Broj bajtova** [7:4] – broj bajtova podataka u paketu, **ECC** [3:0]

– Hemingov kod računa se na osnovu bita tipa paketa i broja bajtova. Može da detektuje dvostruke i ispravi jednostruke greške.

SOP	P	Tip paketa	Broj bajtova	ECC
-----	---	------------	--------------	-----

Slika 3. Struktura zaglavlja paketa

Polje	Zaglavlje paketa	Podaci	CRC8
Veličina u bajtovima	2	1-16	1

Slika 4. Struktura paketa

3.3. HEMINGOV KOD

U osnovi Hemingov (ECC) kod se sastoji od 4 bita koji se prenose zajedno sa bitima poruke koja sadrži potrebne informacije (broj bajtova u paketu i tip paketa). Svaki bit ECC koda formira se primenom XOR operacije između odgovarajućih bita poruke koji nose potrebne informacije. Da bi se greška detektovala neophodno je formirati korekcionni kod C, na osnovu čije vrednosti se zaključuje koji bit poruke je pogrešan. Shodno tom, pogrešan bit se invertuje, kako bi se dobila ispravna poruka [2].

Za detekciju dvostrukih grešaka neophodno je uvesti dodatan bit P, koji se formira primenom XOR operacije između svih bita koji se nalaze u poruci. U odnosu na kombinaciju vrednosti bita P i korekcionnog koda C razlikuju se slučajevi prikazani na slici 5.

C	P	Zaključak
0	0	Nema greške
0	1	Greška se nalazi u P bitu.
≠ 0	1	Greška na jednom bitu.
≠ 0	0	Greška na dva ili više bita.

Slika 5. ECC greške [2]

4. IMPLEMENTACIJA SISTEMA

Implementacija sistema podeljena je u tri faze: Faza komunikacije, faza konfiguracije i faza obrade. Prva faza uključuje razvoj interkonekt modula i kontrolera za AXI4 protokol. Ove komponente predstavljaju ključnu ulogu za komunikaciju ostalih blokova u sistemu. Osnovni cilj je implementacija funkcionalnog interkonekta, koji omogućava komunikaciju master-slejev komponenti uz primenu arbitracije po *Round-robin* principu, i master i slejev kontrolera koji poštuju specifikaciju AXI4 standarda.

Druga faza uključuje razvoj glavnog kontrolera u sistemu kao i eksternih i internih registarskih blokova. Ove komponente neophodne su za konfiguraciju blokova za izgradnju i parsiranje paketa i shodno tome osnovni cilj je postići ispravnu konfiguraciju u trenutku kada blokovi započinju zahtevanu obradu podataka.

Treća faza uključuje razvoj blokova za izgradnju i parsiranje paketa. Ove komponente predstavljaju suštinu rada sistema jer proizvode rezultat obrade podataka. Osnovni cilj jeste implementacija blokova koji su sposobni da izgrade paket u skladu sa definisanom konfiguracijom i zaključuje postojanje eventualnih kvarova u toku prenosa paketa.

5. VERIFIKACIJA SISTEMA

Verifikacija sistema prati faze razvoja sistema. Pre svega, neophodno je osigurati da komponente mogu međusobno da komuniciraju poštujući AXI protokol. Zatim sledi

verifikacija rada kontrolera koji treba da obezbedi ispravnu konfiguraciju komponenti za obradu podataka. Na kraju, potrebno je verifikovati integritet podataka prilikom izgradnje paketa u odnosu na svaki tip operacije, kao i sposobnost sistema za detekciju i korekciju grešaka.

Tokom svake faze primenjen je verifikacioni postupak koji se sastoji iz tri koraka. U prvom koraku neophodno je proveriti da li sistem uopšte ispunjava osnovne funkcionalnosti predviđene specifikacijom dizajna (eng. *Design Exercise*). To se postiže definicijom jednostavnih tački pokrivenosti koje daju izvesnu dozu samopouzdanja da sistem prati glavnu ideju i namenu. Očekivano je da se u ovoj fazi pronađe najveći broj kvarova u sistemu. Kako broj kvarova opada, verifikacija prelazi na drugi korak a to je traženje specijalnih ugaonih slučajeva u kojima se kriju neki neočekivani problemi (eng. *Bug Hunting*). U ovom trenutku postavljena su stroga ograničenja sa fokusom na specijalne slučajeve. Postepenim popuštanjem ograničenja prelazi se na treći korak, gde formalni alat uzima u obzir potpunu kompleksnost sistema kako bi postigao potpun dokaz (eng. *Full Proof*). Ovo predstavlja vrlo zahtevan zadatak, koji ako se ispuni, daje veliku dozu sigurnosti da sistem radi ispravno [1].

5.1. INTEGRITET PODATAKA

Verifikacija integriteta podataka bazira se na primeni slobodnih promenljivih (eng. *Non-Deterministic variables*). Vrednosti koje uzima slobodna promenljiva potpuno su randomizovane uz ograničenje da ostaju stabilne za jedan simulacioni put (test slučaj). Kada model sadrži određeni nivo simetrije, ove promenljive mogu značajno povećati efikasnost formalne verifikacije [1].

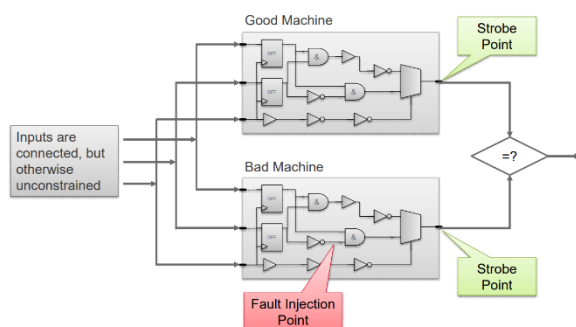
Struktura paketa, kao što je već pominjano u specifikaciji, sastoji se od zaglavlja paketa, podataka i CRC koda. Organizacija bajtova u paketu zavisi od broja bajtova podataka i operacije koja se primenjuje prilikom formiranja paketa.

Vrednost slobodne promenljive odnosi se na poziciju određenog bajta ulaznih podataka. U izgrađenom paketu, zbog dodavanja dodatnih bajtova zaglavlja paketa i tipa operacije koji se primenjuje, pozicija selektovanog bajta će imati drugu vrednost. Vrednosti izabranog bajta se registruju na ulazu i izlazu, a zatim porede kako bi se potvrdio integritet. Na ovaj način slobodna promenljiva omogućava generalizaciju formalne analize, jer će svi bajtovi podataka u paketu biti provereni odjednom.

5.2. FSV TEHNIKA

FSV (eng. *Formal Safety Verification*) aplikacija služi za dokazivanje sigurnosti sistema koji poseduju određenu redundantnu logiku sa ciljem detektovanja i korekcije kvarova. FSV funkcioniše tako što se definišu interni signali sa kvarom i tačke opservacije (eng. *strobe points*) koje mogu biti FO (eng. *functional output*) ili CO (eng. *checker output*) tipa, gde alat generiše tvrdnje u skladu sa definisanim tačkama kako bi dokazao sigurnost ili pronašao kontra-primer. U odnosu na FO sistem se smatra sigurnim ako se greška ne može propagirati do FO, zbog redundantne logike koja to sprečava, ili ako se greška uvek može detektovati na CO. Da bi se to utvrdilo FSV formira formalni model ispravne i mašine sa kvarom (slika 6), gde se porede definisani izlazi kako bi se utvrdila sposobnost sistema za detekciju i propagaciju kvara.

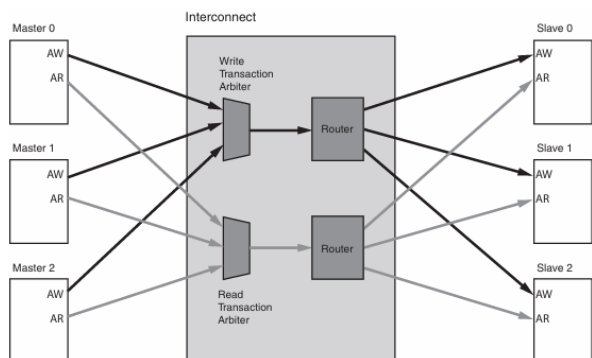
Sistem za procesiranje paketa poseduje logiku u zaglavlju paketa koja služi za detekciju i otklanjanje kvarova koji se odnose na informaciju o tipu i broju bajtova u paketu. ECC kod treba da bude sposoban da detektuje dvostruke i ispravlja jednostruke greške, što je dokazano pomoću FSV aplikacije [4].



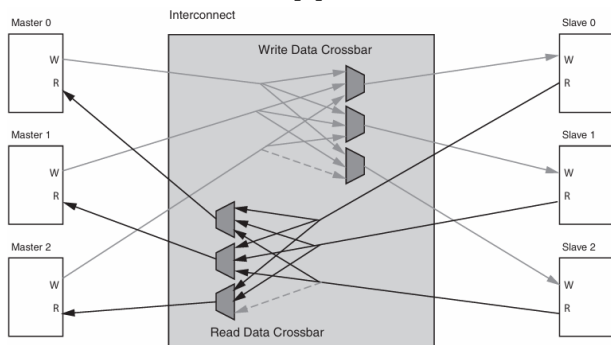
Slika 6. FSV model [4]

6. UNAPREĐENJE SISTEMA

Usko grlo u sistemu takođe predstavlja interkonekt koji podržava samo jednu transakciju u jednom trenutku. Deljeni pristup interkonekt (eng. *Shared Access Mode*) modulu, iako zahteva malo hardverskih resursa za implementaciju ne obezbeđuje dobre performanse sistema. Bolje, ali i skuplje, rešenje po pitanju resursa predstavlja interkonekt u *crossbar* režimu (slike 7 i 8). Struktura prati SAMD topologiju (eng. *Shared-Address Multiple-Data*). Nezavisni adresni kanali poseduju zasebne arbitre za transakcije upisa i čitanja. Master koji dobije *grant* može da zahteva određenu liniju za podatke prema slejv modulu. Kanali za prenos podataka organizovani su u *crossbar* topologiji koja omogućava da nezavisni parovi mogu da razmenjuju podatke u paraleli [3].



Slika 7. Deljeni pristup adresnih kanala za upis i čitanje [3]



Slika 8. Crossbar organizacija kanala za podatke [3]

7. ZAKLJUČAK

Implementacija sistema poseduje sve funkcionalnosti predviđene specifikacijom. Sistem je sposoban da na osnovu eksternog prekida, pokrene različite operacije izgradnje paketa, kao i da detektuje i ispravi eventualne greške nastale u procesu prenosa podataka. Svaka faza implementacije, kao i sistem u celosti, verifikovani su primenom formalne metodologije uz upotrebu efikasnih tehnika. Na kraju, razmatran je pravac unapređenja koji omogućava efikasniju komunikaciju komponentni u sistemu. Na osnovu svega navedenog u radu može se zaključiti da je sistem upešno projektovan i verifikovan primenom isključivo formalnih metoda.

8. LITERATURA

- [1] E. Seligman, E. T. Schubert и K. M. V. A. Kiran, Formal verification: An essential toolkit for modern VLSI Design, Elsevier/Morgan Kaufmann, 2015.
- [2] N. Raut, „tutorialspoint,“ [На мрежи]. Available: <https://www.tutorialspoint.com/hamming-code-for-single-error-correction-double-error-detection>. [Последњи приступ 22 1 2024].
- [3] „LogiCORE IP AXI Interconnect,“ [На мрежи]. Available: <https://www.xilinx.com/support>.
- [4] „JasperGold Functional Safety Verification,“ Cadence, [На мрежи]. Available: <https://support.cadence.com>.

Kratka biografija:



Aleksa Đoković rođen je u Prijepolju 1999. god. Diplomirao je na Fakultetu tehničkih nauka u Novom Sadu, na smeru Energetika, elektronika i telekomunikacije 2022. godine. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Energetska, elektronika i telekomunikacije odbranio je 2024. god.
Kontakt: djokovic64@gmail.com

ДИЗАЈН И ЕВАЛУАЦИЈА АЛГОРИТАМА ЗА МЕРЕЊЕ УДАЉЕНОСТИ ЗАСНОВАНЕ НА ВРЕМЕНУ ДОЛАСКА СИГНАЛА (TOA) ЗА IEEE 802.11AZ DESIGN AND EVALUATION OF TIME OF ARRIVAL (TOA)-BASED RANGING ALGORITHMS FOR IEEE 802.11AZ

Андреј Скала, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У овом раду је представљен алгоритам за процену долазног времена сигнала назван ЛЦЦ алгоритам, као алтернатива са ниском комплексношћу МУСИЦ алгоритму. Оба алгоритма су имплементирана у систему за позиционирање и њихови резултати су дискутовани за различите параметре.

Кључне речи: MUSIC, Wi-Fi, ToA естимација

Abstract – This paper presents a ToA estimation algorithm called LCC as a low complexity alternative to MUSIC. Both algorithms were implemented in a positioning system and their results were discussed in respect to different parameters.

Keywords: MUSIC, Wi-Fi, ToA estimation

1. УВОД

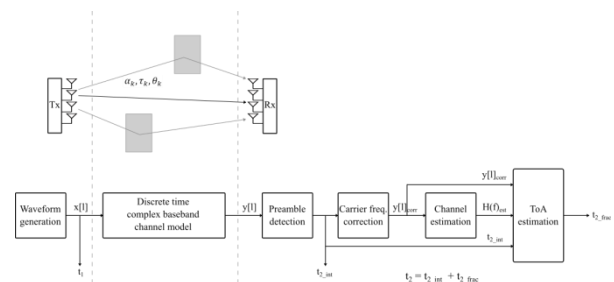
Са појавом Интернета Ствари (енг. *Internet of Things* IoT), прецизна локализација у затвореним просторијама је добила на значају [1]. Разни системи за локализацију су развијени, користећи различите безичне технологије и разне методе и алгоритме, као што су методе засноване на долазном углу сигнала (енг. *Angle of Arrival* AoA), времену доласка сигнала (енг. *Time of Arrival* ToA) и примљеној снази сигнала (енг. *Received Signal Strength* RSS). Главни проблем за ToA технике јесте прецизна естимација пропагационог кашњења радио сигнала из директног правца (енг. *direct line of sight* DLOS) [1]. Овај рад разматра популарни ToA алгоритам који се зове MUSIC [1] и уводи алгоритам ниске сложености заснован на корелацији назван LCC (енг. *low complexity correlation*).

2. МОДЕЛ СИСТЕМА ЗА ПРЕНОС СИГНАЛА

Модел система за пренос сигнала и процену ToA. се састоји из три главне компоненте: предајник са више антена, дискретни (комплексни) модел канала и пријемник са више антена. Предајник генерише Wi-Fi сигнал и бележи време почетка преноса. Модел канала моделира стварне сложености затвореног простора са више пропагационих путања (енг. *indoor multipath environment*), узимајући у обзир ефекте рефлексије и

кашњења сваког пута сигнала [2].

Коначно, пријемник, опремљен вишеструким антенама, детектује преамбулу, врши корекцију времена и фреквенције, као и процену одзива канала у фреквенцијском домену. На крају, користећи информације које је добио из примљеног пакета, процењује време доласка сигнала. Важно је напоменути да сваки део модела система уводи ограничење у укупну тачност процене ToA, од бележења времена, до детекције почетка преамбуле и естимације канала и ToA.



Слика 1. Блок дијаграм општег модела за пренос сигнала и естимацију ToA

3. LCC АЛГОРИТАМ

LCC (енг. *Low complexity correlation*) је једноставан алгоритам за естимацију времена доласка пакета који не користи естимирани фреквентни одзив канала као MUSIC, већ користи послат и примљене сигнале као улаз. Базира се на корелацији примљеног сигнала са послатим сигналом који је закашњен за τ из неког интервала, а резултат је псеудоспектар који ће имати максимум за τ који је најближи ToA. У зависности од тога да ли су сигнали који се корелишу у временском или фреквенцијском домену разликујемо две верзије алгоритма: TLCC верзија (у временском домену) и FLCC верзија (у фреквенцијском домену). Испоставља се да обе верзије имају сличну прецизност, али се могу разликовати у сложености. Фокус овог рада је на FLCC верзији.

Кораци FLCC алгоритма:

Корак 1: Израчунати брзу Фуријеову трансформацију (енг. FFT) послатих и примљених сигнала.

Корак 2: Конструисати псеудоспектар за сваку пријемну антену тако што се, за свако кашњење τ из одређеног интервала са кораком $\Delta\tau$, пропорционално промени фаза одбирака спектра послатог сигнала $Tx(n)$ и добијени спектри корелишу са спектром примљеног сигнала $Rx_m(n)$:

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Иван Мезен, ред. проф.


```

for  $\tau = \tau_a : \Delta\tau : \tau_b$ 
     $TxDelay(n, \tau) = Tx(n) \cdot e^{-j \cdot \tau \cdot g(n)}$ 
    for  $m = 1 : M$ 
         $P_m(\tau) = |TxDelay(n, \tau)^H Rx_m(n)|^2$ 
    end
end

```

Корак 3: Наћи локације глобалних максимума спектра за сваку антену ($i.e. \tau_m$).

Корак 4: ТоА је средња вредност тих локација.

$n = 1, 2, \dots, N$ је редни број одбирка спектра сигнала, $m = 1, 2, \dots, M$ је редни број антене пријемника.

Једна од важних карактеристика LCC алгоритма је то што корак $\Delta\tau$ може да буде било које величине, од реда величине целог одбирка до дела одбирка. Ово значи да се LCC алгоритам може користити и за процену целог дела и децималног дела кашњења. Обе могућности су анализирани у овом раду.

Још једна могућа оптимизација за обе верзије LCC алгоритма је да, ако су интервал претраге и корак унапред познати и послати сигнал се не мења, све вредности послатих сигнала са кашњењем ($tx(Tx)Delay$) је довољно израчунати само једном на почетку алгоритма. Овим се алгоритам своди само на рачунање псеудоспектра и тражење локација његових максимума.

3.1 Анализа комплексности

Ако је $r = \tau_b - \tau_a$ величина интервала претраге, $\Delta\tau$ је величина корака, N ће бити коришћен као број одбирака $tx(Tx)Delay$ и $rx_m(Rx_m)$ и M је број антена пријемника, онда се број операција множења за TLCC и FLCC верзије налази у табели 3.

Табела 1. Број операција множења коришћених у TLCC и FLCC верзијама, са и без оптимизације рачунања $tx(Tx)Delay$ на почетку

Верзија алгоритма	Број операција множења (без рачунања $tx(Tx)Delay$ на почетку)	Број операција множења (са рачунањем $tx(Tx)Delay$ на почетку)
TLCC	$\frac{r}{\Delta\tau} \cdot (2N \log N + N + MN)$	$\frac{r}{\Delta\tau} \cdot M \cdot N$
FLCC	$2N \log N + \frac{r}{\Delta\tau} \cdot (N + MN)$	$N \log N + \frac{r}{\Delta\tau} \cdot M \cdot N$

Може се видети да FLCC верзија има мање множења у случају без оптимизације рачунања $tx(Tx)Delay$ на почетку. Такође се може видети како та оптимизација може додатно да смањи комплексност. Један критични параметар јесте $\frac{r}{\Delta\tau}$, број временских јединица за које се рачуна псеудоспектар. Већи интервал претраге или већа жељена прецизност (мањи корак) повећавају време извршавања алгоритма. Други критични параметар јесте N , број одбирака сигнала који се користи у корелацији. За TLCC

верзију, N је број временских одбирака улазног таласног облика, а за FLCC је број одбирака FFT дискретног спектра улаза.

За довољно мале дужине улаза, ова два броја су сличних величина. У овом раду, цела дужина пакета заједно са његовим кашњењем је узета за улаз алгоритма, пошто је то кашњење краће од 20 одбирака, али у реалним условима оно може бити много дуже и треба се одстранити. Друга могућност је да се користе само делови преамбуле као улаз, али како ово може утицати на перформансе би требало посебно испитати.

4. СИМУЛАЦИОНА СТУДИЈА

Оба алгоритма су имплементирани у систему за позиционирање. Систем се састоји од станице (енг. *station* STA), чију позицију покушавамо да израчунамо и најмање три приступне тачке (енг. *accesspoint* AP) на познатим позицијама.

Поступак за позиционирање започиње мерењем удаљености између станице и сваке приступне тачке. Мерење се обавља у два смера за сваку везу: *uplink* и *downlink*, што значи је за мерење удаљености за сваку везу потребно 2 трансмисије и ТоА естимације описане у поглављу 2. Средње време две трансмисије се користи за рачунање удаљености. Када су познате све удаљености, позиција станице се естимира коришћењем нормалне једначине за линеарну регресију са функцијом цене за најмањи квадрат.

4.1 Параметри симулације

Параметри симулације могу да се поделе у параметре система, предајника, канала и пријемника. Више симулационих експериментана је обављено, где су системски параметри увек остали исти, док су се остали параметри могли мењати сваки експеримент. Симулација је рађена користећи Matlab 2022b и WLAN додатак.

4.1.1 Параметри система

Поставка сваког експеримента је велика соба, димензија 30m x 50m са три приступне тачке у три ћошка собе са координатама ((0,0), (30,0), (0,50)) у ху равни. У сваком експерименту се симулира позиционирање 100 станица које су распоређене на насумичним позицијама са униформном дистрибуцијом унутар собе. Ови значи да сваки експеримент симулира 600 преноса и ТоА естимација. Коришћена је SNR вредност од 25dB

4.1.2 Параметри предајника

Параметри предајника су:

- Предајник шаље 802.11az NDP (*Null Data Packet*)
- Носећа фреквенција је 5GHz
- Број понављања HE-LTF поља је 3
- Променљив пропусни опсег (40MHz и 80MHz)
- Променљив број антена предајника (од 1 до 8)

4.1.3 Параметри канала

За модел канала са више путања је коришћен TGax модел типа "B" (B) [3]. Овај модел имитира реалистично понашање у стамбеним просторијама.

4.1.4 Параметри пријемника

За ТоА естимацију су коришћени MUSIC и FLCC алгоритми. Како би се тестирали различити аспекти FLCC алгоритма, његов блок има више верзија:

FLCC - Ово је основна верзија где се цели део почетног одбирка t_{2_int} рачуна у блоку за детекцију преамбуле који се базира на аутокорељацији примљеног сигнала, а FLCC естимира децимални део почетног одбирка t_{2_frac} тако што пролази кроз интервал од једног одбирка око t_{2_int} ($[t_{2_int} - 0.5, t_{2_int} + 0.5]$) са кораком 0.01 (укупно 101 корак).

У овој верзији, FLCC зависи од несавршене детекције преамбуле да сузи простор претраге за њега. Ова верзија се може директно поредити са MUSIC алгоритмом.

FLCCcorrection - У овој верзији, циљ је био да се испитају могућности FLCC за грубу процену кашњења пакета. FLCC блок сада има два дела. У првом делу се FLCC користи да покуша да исправи несавршену процену блока за детекцију преамбуле. Он пролази кроз више одбирака ([7,20] и [7,30] за фреквенције одабирања 40MHz и 80MHz) са кораком 0.5 одбирка (укупно 27 и 47 одбирка).

Након што сузи простор за претрагу, FLCC из прве верзије теста, са кораком 0.01, естимира децимални део кашњења t_{2_frac} .

FLCCideal - Ова верзија теста користи савршену детекцију преамбуле и претпоставља савршену процену целог дела кашњења. FLCC се поново користи за децимални део кашњења. Ова верзија тестира теоретски најбоље могућности FLCC алгоритма.

Ниједна FLCC верзија не користи оптимизацију да израчуна закашњење послате сигнале (tx(Tx)Delay) једном на почетку. Што се тиче осталих параметара, пријемник може имати променљив број антена (од 1 до 8).

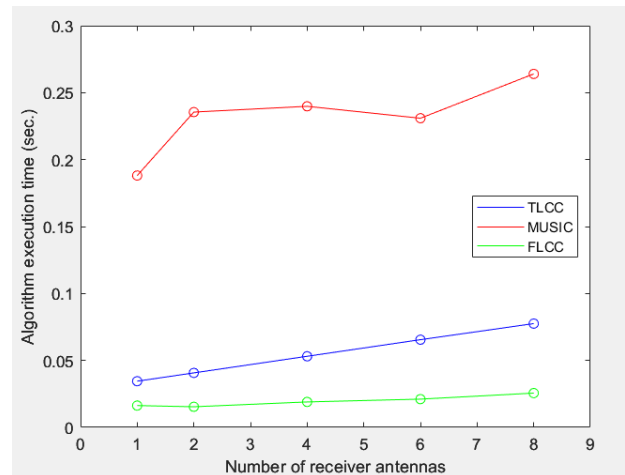
5. АНАЛИЗА РЕЗУЛТАТА

Циљ студије је био поређење алгоритама помоћу две метрике: средње време извршавања алгоритма и просечну апсолутну грешку мерења удаљености.

5.1. Просечно време извршавања алгоритма

Три алгоритма која су тестирана су MUSIC, LCC у временском домену (TLCC) и LCC у фреквенцијском домену (FLCC). Време извршавања сваког алгоритма је узето као време рада блока за ТоА естимацију.

Обе LCC верзије су коришћене за рачунање децималног дела кашњења и имале су интервал претраге од једног одбирка са кораком од 0.01 (укупно 101 корака) Резултати се налазе на слици 17. FLCC треба најмање времена за извршавање, 2-3 пута је бржи од TLCC и око 10 пута бржи од MUSIC алгоритма.

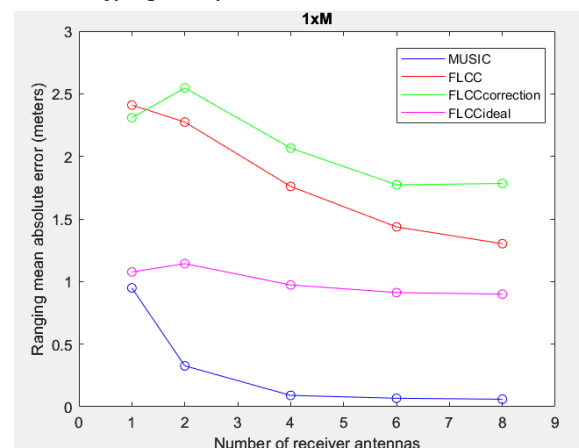


Слика 2. Резултати за средње време извршавања алгоритма за MUSIC, TLCC и FLCC за различит број пријемних антена

5.2. Средња апсолутна грешка мерења удаљености

1 xM антена и пропусни опсег 40MHz

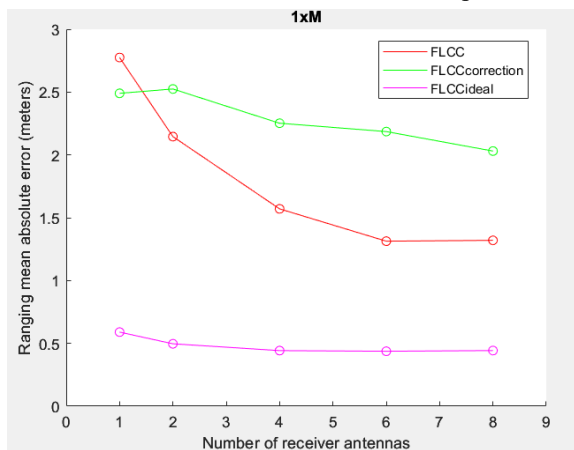
Овај експеримент пореди различите алгоритме и верзије из главе 4.1.4: MUSIC, FLCC, FLCCcorrection и FLCCideal, када предајник користи само једну антену. Види се да је MUSIC јасан победник са грешком која почиње од око 1m и брзо опада до реда величине неколико центиметара. Постоје два фактора који могу да утичу на грешку: процена целобројног дела кашњења и процена децималног дела кашњења. Моћ MUSIC алгоритма је да и поред лоше процене целобројног дела кашњења, може да исправи ову грешку и да прецизан коначни резултат. FLCC пати од овога јер се ослања на грубу процену кашњења да му сузи простор претраге и има средњу грешку 2.5m до 1.5m. FLCCcorrelation међутим, имао је још горе резултате. Када је целобројни део кашњења савршено познат, као код FLCCideal, алгоритам ради значајно боље, са средњом грешком од око 1m. Гледајући резултате на други начин, ако не бисмо имали блок за ТоА естимацију, средња грешка би била барем 7.5m, што значи да све верзије FLCC представљају унапређење у прецизности. Такође се може видети да, када поредимо FLCC и FLCCideal, да већи број антена пријемника има већи ефекат на блок за детекцију преамбуле него на FLCC.



Слика 3. Средња апсолутна грешка рачунања удаљености за сценарио 1xM антена

1 xM антена и пропусни опсег 80MHz

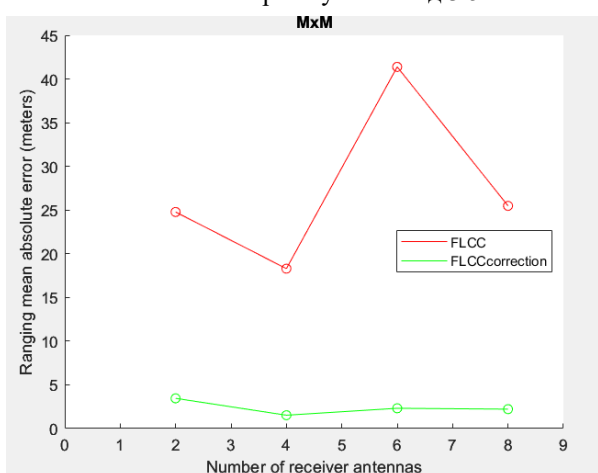
Када се ширина пропусног опсега повећа на 80MHz, експериментима са MUSIC алгоритмом је требало сувише да се заврше, али на основу тестова са мањем бројем станица може се очекивати да опет надмаши све FLCC верзије и има криву сличног изгледа као за пропусни опсег 40MHz која почиње од око 0.5m. Изненађујуће је да су FLCC и FLCCcorrection имали сличне резултате као у претходном експерименту, али се FLCCideal показао два пута боље са просечном грешком сада од око 0.5m, што значи да повећавање ширине пропусног опсега побољшава процену децималног дела кашњења код FLCC алгоритма.



Слика 4. Ефекат повећавања ширине пропусног опсега на средњу апсолутну грешку мерења удаљености

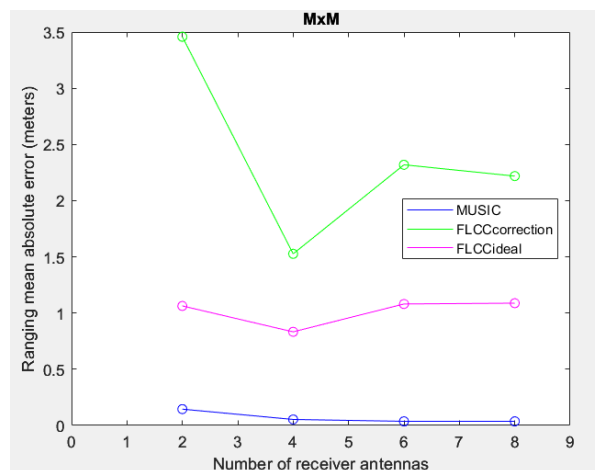
MxM антена и пропусни опсег СВ40MHz

Више антена предајника представља нешто сложеније окружење, пошто сада примљени сигнал представља комбинацију више сигнала. Заиста, може се видети велики негативни ефекатна блок за детекцију преамбуле јер је сад средња грешка за FLCC реда величине десетине метара. FLCCcorrection ипак, успева да побољша ово и смањи грешку на испод 3.5m.



Слика 5. Ефекат више антена предајника на неке FLCC варијанте

FLCCideal има средњу грешку око 1m, слично као и код експеримента са 1xM антена. Може се видети да су са више предајних антена FLCC верије имале сличне или горе резултате него са једном предајном антену. Са друге стране, MUSIC је имао значајно побољшање у прецизности.



Слика 6. Ефекат више антена предајника на MUSIC и остале FLCC варијанте

6. ЗАКЉУЧАК

Циљ рада је био да се испитају перформансе LCC алгоритма као алтернативе MUSIC алгоритму са нижом комплексношћу. Резултати су потврдили да LCC има значајно краће време извршавања, где је најбржа варијанта, звана FLCC, била 10 пута бржа од MUSIC алгоритма. Што се тиче прецизности, MUSIC се показао као очигледан победник у свим тестираним сценаријима, са просечном апсолутном грешком процене удаљености добро испод 1m до реда величине неколико центиметара.

FLCC се није лоше показао у грубој процени кашњења, али је боље резултате имао у процени децималног дела кашњења и, заједно са добрим проценитељем целобројног дела кашњења, је имао грешку испод 1m или мање у зависности од фреквенције одабирања.

Корелација је операција која се већ често користи у Wi-Fi уређајима за синхронизацију, што значи да би се исти ресурси могли потенцијално искористити и за LCC. Додатне предности LCC су нижа комплексност и чињеница да му није потребна естимација канала, па се препоручује за уређаје где је нижа потрошња или уштеда ресурса битнија од велике прецизности позиционирања.

7. ЛИТЕРАТУРА

- [1] Xinrong Li, K. Pahlavan., "Super-resolution TOA estimation with diversity for indoor geolocation", IEEE Transactions on Wireless Communications, vol 3, no. 1, pp. 224-234, Jan. 2004, doi: 10.1109/TWC.2003.819035.
- [2] Tarik Kazaz, Gerard J. M. Janssen, Jac Romme, Alle-Jan van der Veen, "Delay Estimation for Ranging and Localization Using Multiband Channel State Information" in IEEE Transactions on Wireless Communications, vol 21, no 4, Apr. 2022
- [3] Jianhan, L., Ron, P. et al., "TGax Channel Models" IEEE 802.11-14/0882r4, September 2014.

Кратка биографија:

Андреј Скала рођен је у Новом Саду 1997. год. Дипломирао је 2022. год. на Факултету техничких наука из области Електротехнике и рачунарства, смер Ембедед системи и алгоритми, на којем исте године уписује мастер академске студије.
контакт: skalaandrej7@gmail.com

UTICAJ PARAMETARA DISTRIBUTIVNE MREŽE I DISTRIBUIRANOG GENERATORA NA VREDNOST DIFERENCIJALNE STRUJE PARALELNIH VODOVA**IMPACT OF DISTRIBUTION NETWORK PARAMETERS AND DISTRIBUTED GENERATOR ON THE VALUE OF PARALLEL LINE DIFFERENTIAL CURRENT**

Marko Čančar, *Fakultet tehničkih nauka, Novi Sad*

Oblast - ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu predmet razmatranja jeste analiza uticaja parametara distributivne mreže i distributivnog generatora na vrednost diferencijalne struje paralelnih vodova. Cilj rada jeste da se proceni koliko značajno promene tipa voda (vazdušni/kablovski), nejednakost dužina paralelnih voda i postojanje distributivnog generatora na sabirnicama paralelnih vodova utiču na vrednost diferencijalne struje paralelnih vodova.

Ključne reči – Diferencijalna struja, Parametri mreže, Distributivni generator

Abstract – In this paper, the subject of consideration is the analysis of the influence of parameters of the distribution network and distribution generator on the value of the differential current of parallel lines. The aim of the paper is to assess how significant changes in the type of line (overhead/cable), the inequality of parallel line lengths and the existence of a distribution generator on parallel line bus bars affect the value of the differential current of parallel lines.

Keywords – Differential current, Network parameters, Distribution Generator

1. UVOD

Povećanje potreba za energijom, povećanje prisustva distributivnog generatora (DG) i povećanje potreba za prenosom energije u distributivnoj mreži dovode do povećanja broja elemenata mreže i do paralelnog rada elemenata. Za zaštitu paralelnih vodova distributivne mreže se koriste poprečne diferencijalne zaštite [1,2]. Te zaštite nisu predmet razmatranja u ovom radu nego su to struje koje se pri kvaru pojavljuju na paralelnim vodovima – preciznije razlika struja po paralelnim vodovima. Postojanje diferencijalne struje ukazuje na pojavu kvara. Upravo predmet razmatranja ovog rada je analiza kako promene elemenata i parametara distributivne mreže kao i (dodavanje) novog DG utiču na diferencijalne struje na paralelnim vodovima.

U glavi 2 je vrlo kratko objašnjena poprečna diferencijalna zaštita samo da bi se mogao steći kontekst zbog čega se obrađuje ova problematika diferencijalnih struja u ovom radu.

U glavi 3, opisan je matematički model za proračune na test mreži sa uključenim DG. Ovaj model je iskorišćen za proračun diferencijalnih struja na paralelnim vodovima.

U glavi 4 je data analiza tipa (vazdušni/kablovski) tip voda i promene dužine paralelnih vodova i DG na vrednost diferencijalne struje paralelnih vodova. Cilj je bio da se dobije uvid koliko varijacije parametara vodova – a izabrana je analiza efekta nejednakih dužina paralelnih vodova utiče na promenu vrednosti poprečne diferencijalne struje. Dodatna analiza je izvedena za slučaj kada se na sabirnice na kojima završavaju vodovi priključi jedan distributivni generator.

U poslednje dve glave dat je zaključak rada i spisak korišćene literature za pisanje ovog rada.

2. DIFERENCIJALNA ZAŠTITA VODOVA

U ovom delu su date osnovna razmatranja vezana za diferencijalnu zaštitu paralelnih vodova. Ova zaštita se poprečnome diferencijalne zaštite. Podužna diferencijalna zaštita nije predmet ovog rada, pa je izostavljena.

2.1. Poprečna diferencijalna zaštita

Poprečna diferencijalna zaštita se koristi za zaštitu paralelnih vodova sa približno jednakim parametrima. Potrebno je da oba voda polaze i da se završavaju na zajedničkim sabirnicama.

Ako su dva voda dovoljno slična, da se mogu smatrati istim (ako su impedanse paralelnih vodova jednake ili približno jednake), onda u normalnom pogonu i kod kvarova van zone štice struje kroz vodove su približno iste, pa je njihova razlika vrlo mala ili jednaka nuli. Međutim, situacija se menja ako se na jednom od vodova dogodi kvar. U takvim slučajevima počinje da se razlikuje struja po vodovima, tj. razlika struja (diferencijalna struja) je različita od nule i to se koristi kao kriterijum za delovanje zaštite.

Postoji dve varijante poprečne diferencijalne zaštite: obična i selektivna diferencijalna zaštita.

Obična poprečna diferencijalna zaštita ima prekidače ispred i iza paralelnih vodova. Međutim, ako se kvar dogodi samo na jednom vodu, nema potrebe da se isključuju oba voda, već samo vod sa kvarom i tada se koristi selektivna poprečna diferencijalna zaštita, koja će isključiti samo vod sa kvarom [1-5].

NAPOMENA:

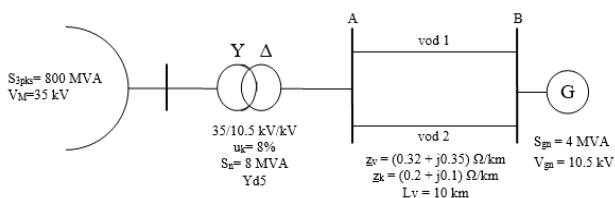
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Duško Bekut, red. prof.

3. MATEMATIČKI MODEL ZA PRORAČUN DIFERENCIJALNE STRUJE NA PARALELNIM VODOVIMA

U ovoj glavi je predstavljen matematički model za proračun diferencijalne struje na paralelnim vodovima za primer distributivne mreže prikazan na slici 3.1. Proračun je urađen za trolpolni kratak spoj. Razmatraju se proračuni samo diferencijalne struje vodova kod sabirnica A pri čemu se ne razmatra i podešavanje same poprečne diferencijalne zaštite za šta su potrebni proračuni i tokova snaga što izlazi iz okvira razmatranja ovog rada.

3.1. Postavka problema

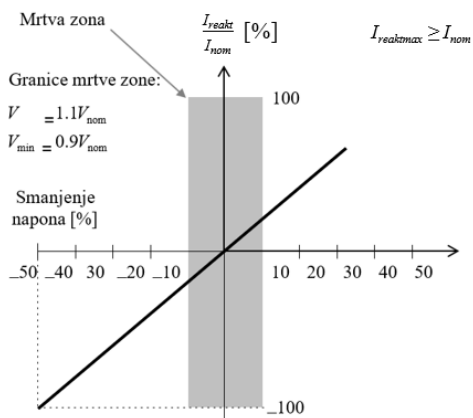
Za analizu vrednosti diferencijalne struje izabran je primer dva paralelna voda 10 kV (vazdušni i kablovski tip). Potrebno je simulirati trolpolni kratak spoj na vodu 1 i izračunati kolika se razlika struja meri na početku vodova. S obzirom da napajanje paralelnih sabirnica dolazi preko sabirnica A, ovaj slučaj je kritičniji za analizu od slučaja na kraju vodova kod sabirnica B. Parametri elemenata mreže su dati na slici 3.1.



Slika 3.1 Test distributivna mreža

3.2. Doprinos distributivnog generatora struji kratkog spoja

U radu je analiziran i uticaj DG na vrednost diferencijalne struje. Tokom trajanja kratkog spoja, postoji zahtev da DG injektiranja reaktivnu snagu u cilju očuvanja (podrške) napona čvora na koji je priključen u unapred definisanim granicama (podrazumevaju se DG koji su preko odgovarajuće energetske elektronike povezani na distributivnu mrežu [6]). Vreme trajanja naponske podrške DG za vreme kvara, kao i vrednost injektiranja, definisani su standardima. U ovom radu je korišćen nemački standard (Nemački grid kod) [6] koji definiše dve karakteristike: vreme trajanja naponske podrške DG tokom kvara i zavisnost injektiranja reaktivne struje DG od sniženja napona. Zavisnost injektiranja struje generatora od sniženja napona je prikazana na slici 3.2 [6].



Slika 3.2 Zavisnost injektiranja struje generatora od sniženja napona

Sa slike 3.2 se na osnovu sniženja napona priključnog čvora može odrediti vrednost injektiranja struje DG. Sa ove krive se može očitati da za svakih 1% sniženja napona injektiranje struje DG je 2% (struja koja se injektira ima fazni stav koji je kapacitivno pomeren za ugao od 90 stepeni u odnosu na fazni stav struje pre kvara). Najveće strujno injektiranje ne može da premaši 150% nominalne struje DG. Takođe, označena je i mrtva zona strujnog injektiranja DG, koja pokazuje da ukoliko je sniženje napona $\pm 10\%$, nema promene injektiranja generatora.

$$I_g = \frac{2}{100} \Delta U_{gen} I_{gn}, \quad (3.1)$$

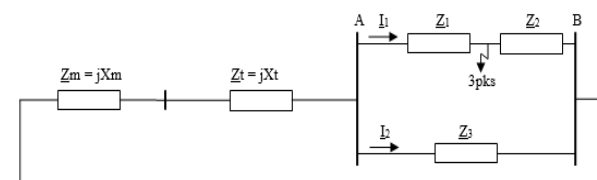
gde su:

ΔU_{gen} – sniženje napona priključnog čvora,

I_{gn} – nominalna struja generatora.

3.3. Izvođenje matematičkog modela

Prvi korak izvođenja matematičkog modela predstavlja formiranje šeme direktnog redosleda za mrežu, a u ovom slučaju je to mreža prikazana na slici 3.3, i izračunavanje struje kvara po vodovima (I_1 i I_2).



Slika 3.3 Pogonska šema test distributivne mreže bez generatora

Relacije za izračunavanje struja kvara po vodovima su:

$$I_1 = \frac{Z_2 + Z_3}{Z_1 + Z_2 + Z_3} I_{3pks}, \quad (3.2)$$

$$I_3 = I_{3pks} - I_1, \quad (3.3)$$

gde su:

Z_1 – impedansa voda 1 između sabirnica A i mesta kvara,

Z_2 – impedansa voda 1 između sabirnica B i mesta kvara,

Z_3 – impedansa voda 2,

I_{3pks} – nominalna snaga transformatora.

Zatim, potrebno je izračunati procentualno sniženje napona DG (na sabirnicama B) u odnosu na njegovu nominalnu vrednost:

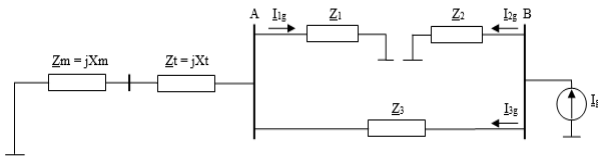
$$\Delta U_{gen} = \left(1 - \frac{\sqrt{3} |U_{gen}|}{V_{gn}} \right) \cdot 100, \quad (3.4)$$

gde su:

U_{gen} – fazni napon na mestu priključenja generatora za vreme kvara,

V_{gn} – nominalni linijski napon generatora.

Sada treba odrediti doprinos DG strujama I_1 i I_3 . DG će se modelovati preko idealnog strujnog izvora priključenog na sabirnicama B, kao na slici 3.4.



Slika 3.4 Pogonska šema test distributivne mreže sa DG

Strujno injektiranje DG (I_g) na mestu priključenja izračunava se pomoću relacije 3.1. Doprinos generatora strujama kvara po vodovima se dobija preko strujnog razdelnika (I_{1g} i I_{3g}).

$$I_{3g} = \frac{Z_2}{Z_2 + Z_3 + Z_1 || (Z_M + Z_T)} I_g, \quad (3.5)$$

$$I_{1g} = \frac{Z_M + Z_T}{Z_1 + Z_M + Z_T} I_{3g}, \quad (3.6)$$

gde su:

Z_M – impedansa napojne mreže,
 Z_T – impedansa napojnog transformatora.

Diferencijalna struja kod sabirnica A, na mestu ugradnje diferencijalne zaštite, sa uključenim DG je:

$$\Delta I_{gen} = (I_1 + I_{1g}) - (I_3 - I_{3g}). \quad (3.7)$$

Ukoliko se u relaciji za izračunavanje ΔI_{gen} izostave struje DG (I_{1g} i I_{3g}), dobija se izraz za izračunavanje diferencijalne struje za isključeno stanje DG:

$$\Delta I = I_1 - I_3. \quad (3.8)$$

Procentualna razlika diferencijalnih struja sa i bez DG u odnosu na diferencijalnu struju bez DG izračunava se na sledeći način:

$$\Delta I_{proc} = \frac{\Delta I - \Delta I_{gen}}{\Delta I} \cdot 100. \quad (3.9)$$

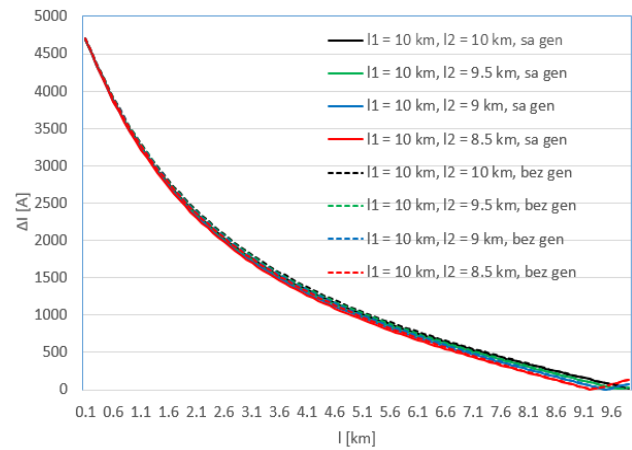
4. ANALIZA REZULTATA PRORAČUNA DIFERENCIJALNIH STRUJA PARALELNIH VODOVA

U ovom delu su prikazani rezultati proračuna diferencijalne struje za mrežu sa vazдушnim/kablovskim vodovima, pri promeni dužine paralelnih vodova. Analizira se primer za mrežu sa slike 3.1 gde su vodovi vazdušni, odnosno, kablovski.

Dužina voda 1 je konstantna u svim razmatranjima dok su za dužinu voda 2 korišćene sledeće vrednosti: 10, 9.5, 9 i 8.5 km (izbor ovih dužina odgovara slučajevima istih, vrlo sličnih, malo različitih i dosta različitih dužina vodova).

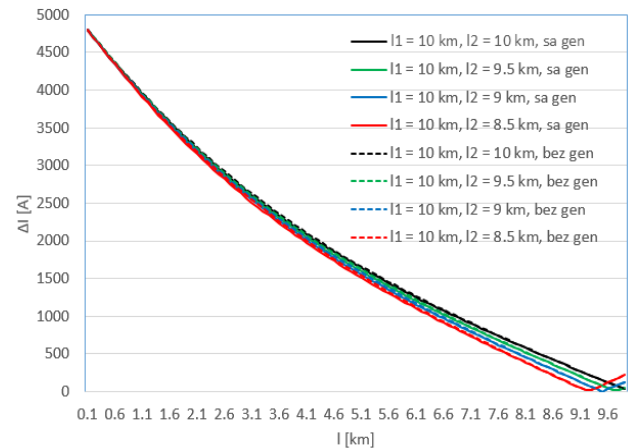
Duž voda 1 se simulira trolpolni kratak spoj od 0.1 do 9.9 km dužine voda merene od sabirnica A, sa korakom 0.1 km. Ostali parametri elemenata mreže su fiksni. Rezultati proračuna vrednosti diferencijalnih struja sa i bez DG za slučaj vazдушnih vodova su prikazani na slici 4.1, a na slici 4.2 za slučaj kablovskih vodova.

Sa slike 4.1 vidi se da je diferencijalna struja za kvar na početku voda 1 najveća.



Slika 4.1 Zavisnost razlike struja od dužine voda 2 kod vazдушnih vodova

Razlog za to je što su struje kratkih spojeva veće za kvar bliži izvoru napajanja. Iz istog razloga se diferencijalna struja smanjuje sa povećanjem rastojanja između mesta kvara i sabirnica A. Iz dobijenih rezultata se može uočiti da je uticaj promene diferencijalne struje za različite dužine vodova relativno mali i da više dolazi do izražaja za mesta kvara kod sabirnica B jer tu najviše dolazi do izražaja nejednakost impedansi vodova. Važno je primetiti da nejednake dužine vodova utiču da se diferencijalna struja na vodu 1 smanjuje za mesto kvara koji odgovara dužini voda 2 što znači da nejednake dužine paralelnih vodova skraćuju dužinu štice. Takođe, rezultati sa uključenim DG i bez njega su slični, uticaj DG je zanemarljiv.



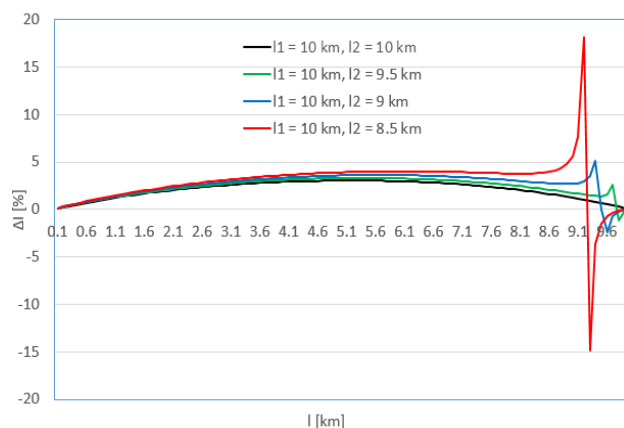
Slika 4.2 Zavisnost razlike struja od dužine voda 2 kod kablovskih vodova

Sa slike 4.2 se vidi da su diferencijalne struje kablovskih vodova slične odgovarajućim strujama vazдушnih vodova sa slike 4.1, što je posledica samog menjanja vrednosti podužne impedanse kablovskih vodova. Ostali efekti su isti kao i kod vazдушnih vodova.

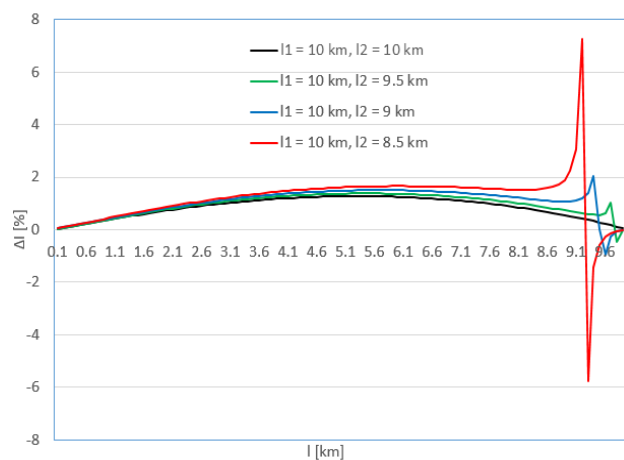
Procentualna razlika diferencijalnih struja za uključeno i isključeno stanje DG za vazdušne vodove prikazana je na slici 4.3, dok je na slici 4.4 prikazan slučaj za kablovske vodove.

Analizom rezultata proračuna sa slike 4.3 može se uočiti da je procentualna razlika diferencijalnih struja najveća za kvar na sredini voda, reda veličine do oko 4%, pa do 18%

za kvarove bliže sabirnicama B. Takođe, vidi se da je za veću razliku dužina vodova veća i procentualna razlika diferencijalnih struja. To je posledica činjenice da je mesto kvara bliže mestu gde se nalazi električna sredina mesta kvara na paralelnim vodovima (diferencijalna struja koja potiče od sistema (u ovom primeru to je mreža 35 kV) teži ka nultoj vrednosti). Zato strujno injektiranje DG i razlike struja po vodovima od DG više dolaze do izražaja jer na njihovu raspodelu utiču samo impedanse vodova. Zato se za mesta kvara pri kraju voda 2 pojavljuju pikovi na slici 4.3. kao posledica činjenice da vrednost diferencijalne struje teži ka malim vrednostima.



Slika 4.3 Procentualna razlika diferencijalnih struja za uključeno i isključeno stanje generatora, za različite dužine voda 2 za vazdušne vodove



Slika 4.4 Procentualna razlika diferencijalnih struja za uključeno i isključeno stanje generatora, za različite dužine voda 2 za kablovske vodove

Sa slike 4.4 vidi se da su procentualne razlike diferencijalnih struja kablovskih vodova manje od odgovarajućih procentualnih razlika diferencijalnih struja vazdušnih vodova sa slike 4.3 (oko 2% za kvarove na sredini voda, pa do 7.5% za kvar na sabirnicama B). Ostali efekti su isti kao i kod vazdušnih vodova.

5. ZAKLJUČAK

U ovom radu analizirani su uticaji parametara mreže i priključenja DG na vrednost poprečne diferencijalne struje paralelnih vodova.

Analizirani su različiti slučajevi promene parametara distributivne mreže: tip (vazdušni/kablovski) paralelnih

vodova i nejednake dužine ovih vodova. Konstatovano je da su promene najviše izražene u slučaju vazdušnih vodova, a da su manje izražene kod kablovskih vodova. Uticaj nejednakih dužina vodova najviše dolazi do izražaja na kraju voda suprotno od mesta proračuna diferencijalnih struja. Nejednake dužine paralelnih vodova utiču da se diferencijalna struja na vodu 1 smanjuje za mesto kvara koji odgovara dužini voda 2 što znači da nejednake dužine skraćuju dužinu štice. Konstatovano je da je uticaj DG na ove promene zanemariv.

Na kraju je konstatovano da je promena diferencijalnih struja ipak relativno mala i da na taj način ne može biti bitnije uticati bitno na podešenje poprečne diferencijalne zaštite.

6. LITERATURA

- [1] D. Bekut: Relejna zaštita, FTN izdavaštvo, Novi Sad, 2009.
- [2] M. Forcan, Z. Stojanović: A standby protection scheme to complement transverse differential protection of double circuit lines in the case of one parallel line tripped, Electric Power Systems Research, Volume 201, 2021, ISSN 0378-7796, <https://doi.org/10.1016/j.epsr.2021.107570>.
- [3] M. Kletsel, B. Masharapov: Differential protection of three and four parallel lines of idling current control, PRZEGLĄD ELEKTROTECHNICZNY, ISSN 0033-2097, R. 93 NR 10/2017, doi:10.15199/48.2017.10.26.
- [4] L. Shilong, C. Wei, Y. Xianggen, C. Deshu, O. Malik: Integrated Transverse Differential Protection Scheme for Double-Circuit Lines on the Same Tower, IEEE Transactions on Power Delivery, ISSN: 0885-8977, Volume: 33, October 2018, <https://doi.org/10.1109/TPWRD.2017.2765762>.
- [5] L. Xiaohua, Y. Xianggen, Z. Zhe, C. Deshu: Department of Electrical Engineering, Huazhong – A New Novel of transverse differential protection Scheme University of science and technology, Wuhan Hubei, 430074, P.R. China, International Conference on Power Systems Transients – IPST 2003, https://www.ipstconf.org/papers/Proc_IPST2003/03IPST05b-04.pdf.
- [6] L. Strezoski, I. Stefani, D. Bekut: Novel method for adaptive relay protection in distribution systems with electronically-coupled DERs, International Journal of Electrical Power & Energy Systems, 2020, Vol. 116, pp. 1-10, ISSN 0142-0615.

Kratka biografija:



Marko Čančar rođen je u Sremskoj Mitrovici 1994. godine. Osnovne studije završio je na Fakultetu tehničkih nauka 2017. godine. iz oblasti Elektrotehnika i računarstvo, smer Elektroenergetski sistemi. Master rad, na istom fakultetu, odbranio je 2024. godine.

РАЗВОЈ БЕКЕНД ПОДСИСТЕМА БЕЗ СЕРВЕРА ЗА ВЕБ 3.0 ПЛАТФОРМУ ЗА УПРАВЉАЊЕ КУРСЕВИМА**DEVELOPMENT OF A SERVERLESS BACKEND FOR A WEB 3.0 PLATFORM FOR COURSE MANAGEMENT**Марија Петровић, *Факултет техничких наука, Нови Сад***Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО**

Кратак садржај – Рад се бави развојем бекенд подсистема без сервера (енгл. *serverless*) апликације за управљање курсевима применом рачунарства у облаку. Поред три главна модела услуга (IaaS, PaaS, SaaS) обрађени су и све популарнији концепти *serverless* и *FaaS* услуга. Задатак рада је да представи једно од могућих архитектуралних решења које смањује недостатке традиционалних система користећи предности облака и интеграцију са блокчејн компонентом. Детаљно је описан начин имплементације и како се ресурси на Амазоновом облаку (Amazon Web Services) могу креирати помоћу *IaC* (Infrastructure as Code) концепта.

Кључне речи: *serverless*, платформа, облак, AWS, Ethereum

Abstract – This paper focuses on the development of a *serverless* backend subsystem for a cloud-based application managing courses. In addition to the three main service models (IaaS, PaaS, SaaS), it explores the increasingly popular concepts of *serverless* and *FaaS* services. The task of the thesis is to present one of the possible architectural solutions that addresses the drawbacks of traditional systems by leveraging the advantages of the cloud and integration with the *blockchain* component. The implementation method is detailed, along with how resources on Amazon Web Service (AWS) can be created using the *Infrastructure as Code* (IaC) concept.

Keywords: *serverless*, platform, cloud, AWS, Ethereum

1. УВОД

Традиционалне компаније су се ослањале на локалне сервере и инфраструктуру унутар својих простора за складиштење података и обраду информација, што је довело до све веће потребе за одржавањем сервера који опслужују захтеве.

Такође, појавили су се чести захтеви за велика улагања у набавку хардвера, проширивање меморијског капацитета, али и особља задуженог за управљање овим системима.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Душан Гајић, ванр. проф.

Све претходно наведено створило је добар темељ за развој рачунарства у облаку (енгл. *Cloud Computing*), који је донео револуцију омогућавајући компанијама да складиште податке и користе ресурсе путем удаљених центара (енгл. *data centers*) који се налазе ван њихове инфраструктуре. [1].

Следећи корак ка оптимизацији употребе ресурса јесте појава архитектуре без сервера (енгл. *serverless*) где је провајдер задужен за алокацију и бригу о ресурсима.

Међутим, иако су предложена бројна решења и стандарди, још увек су се јављали проблеми централизоване архитектуре. Као решење за то, настају децентрализовани системи и блокчејн технологија (енгл. *Blockchain*) као једна њихова имплементација *P2P* (*peer-to-peer*) мреже [2].

Овај рад се бави демонстрацијом имплементације *serverless* бекенд (енгл. *backend*) решења чија архитектура обухвата и блокчејн као компоненту.

2. ТЕОРИЈСКЕ ОСНОВЕ**2.1. Рачунарство у облаку**

Дистрибуирани системи имплементирани у стварном свету су често комплексни делови софтвера чије су компоненте размештене на више различитих, физички одвојених машина [3].

Рачунарство у облаку је један од најзначајнијих примера оваквих система и односи се на механизме који корисницима обезбеђују услуге и ресурсе преко Интернета [4].

Услуге облака нуде разни провајдери, од којих су свакако најпопуларнији Amazon Web Services (AWS), Microsoft Azure, и Google Cloud Platform (GCP).

2.2. Основе блокчејн технологије

Блокчејн технологија представља дистрибуирани систем који омогућава сигурну размену података путем криптографски повезаних блокова. Када се формира, блок се повезује са претходним стварајући непрекидан ланац (енгл. *chain*) који се не може променити без сагласности свих учесника у мрежи [5]. Свака трансакција (блок), обезбеђена је криптографским методама и валидирана од стране сваког овлашћеног члана мреже, коришћењем консензус алгоритама. Најпознатији алгоритми су: доказ посла (енгл. *Proof of Work* - PoW), доказ улога

(енгл. *Proof of Stake - PoS*), доказ протеклог времена (енгл. *Proof of Elapsed Time - PoET*) ...

2.3. Ethereum блокчејн

Ethereum је децентрализована, јавна блокчејн платформа заснована на *peer-to-peer* мрежи, која омогућава безбедно извршавање и верификацију паметних уговора. Њен креатор је Виталик Бутерин, објавивши 2013. године Ethereum *whitepaper* који поставља основу за развој децентрализованих платформи и апликација које би подржавале паметне уговоре. Стога, у Ethereum екосистем се уводи рачунарски механизам назван Ethereum виртуелна машина који је централни део Ethereum блокчејна. Она представља извршно окружење за паметне уговоре и управља стањем овог блокчејна.

Сматра се да је Ethereum направио револуцију и када је у питању развој Интернета, уводећи Веб 3.0 генерацију, и тиме подржао децентрализоване апликације (DApps), децентрализоване финансије (DeFi) и децентрализоване размене (DEXs) [6].

Крипто валута која се користи у Ethereum мрежи се назива етер (енгл. *ether - ETH*). Етер има и неколико фракција, тј. мањих јединица, а најважнији су Wei и Gwei.

Ethereum је дуго користио доказ посла (енгл. *Proof of Work*) као консензус алгоритам, међутим од 2022. године прелази на доказ улога (енгл. *Proof of Stake*), који је енергетски ефикаснији и сигурнији механизам.

3. РАЧУНАРСТВО У ОБЛАКУ И SERVERLESS АРХИТЕКТУРА

3.1. Историја рачунарства у облаку

Рачунарство у облаку креће са својим значајнијим развитком шездесетих година прошлог века када су постављени фундаментални концепти – испоручивање сервиса тако да га други могу користити, могућност дељења компјутерских ресурса између више особа и приступ сервисима (ресурсима) путем мреже [7].

Међутим, рани почеци имплементације облака озбиљније крећу тек 1970. године када је компанија IBM лансирала своју прву виртуалну машину – VM/370. Тада је настала виртуализација као фундаментална технологија која стоји иза сваког облака.

Деценију касније појавом виртуалних приватних мрежа (енгл. *Virtual Private Network - VPN*) постаје могуће повезивање рачунара у просторијама како би људи могли да приступе дељеним подацима. Све већом производњом компјутерских ресурса цене су почеле да падају, а услуге да расту, тако да се створио темељ да почетком 21. века изађе и први јавни провајдер облака – Amazon Web Services (AWS).

3.2. Сервисни модели облака

Како је рачунарство у облаку временом добијало све већу популарност издвојило се неколико различитих модела који задовољавају специфичне потребе

корисника, а то су инфраструктура као сервис (енгл. *Infrastructure as a Service - IaaS*), платформа као сервис (енгл. *Platform as a Service - PaaS*) и софтвер као сервис (енгл. *Software as a Service - SaaS*).

3.2. Serverless рачунарство и архитектуре

Коришћењем услуга провајдера рачунарства у облаку смањује се проблем бриге о хардверу али и даље се захтева одређени ниво управљања осталим софтверима на серверу, скалирањем и ценом [8].

Serverless рачунарство представља један модел рачунарства у облаку који функционише тако што провајдер алоцира ресурсе на захтев, водећи бригу о серверима уместо корисника. Сви ресурси се плаћају онолико колико се користе (енгл. *pay-as-you-go*).

3.3. Функција као сервис (Function as a Service – FaaS)

Иако се често користе у истом контексту, функција као сервис је у ствари један вид имплементације *serverless* модела који омогућава покретање функција написаних у различитим програмским језицима у облаку. Графички приказ уобичајеног тока догађаја приликом рада са *serverless* функцијама је дат на слици 1.



Слика 1. Приказ како *serverless* функција ради [9].

Пошто FaaS представља један подскуп *serverless* модела од њега наслеђује предности и мане.

3.4. Пример облака: Amazon Web Services (AWS)

AWS представља обимну и једну од најзаступљенијих платформи за рачунарство у облаку, насталу под окриљем компаније Амазон, и обухвата широк спектар услуга који подржавају раније поменуте моделе облака – IaaS, PaaS, SaaS, понуђене у форми различитих сервиса [10]. Такође, један је од првих провајдера који су увели *pay-as-you-go* модел коришћења облака.

AWS данас има центре података распрострањене на преко 87 зона доступности (енгл. *Availability Zone - AZ*) у регионима широм света који AWS чине скалабилном и високо доступном платформом.

У свом екосистему, AWS нуди велики број сервиса за различите примене и архитектуре, а неки од њих су: Amazon Elastic Compute Cloud (EC2), Amazon Simple Storage Service (S3), Amazon Relational Database Service (RDS), Amazon Lambda, Amazon API Gateway ...

3.5. AWS Cloud Development Kit (AWS CDK)

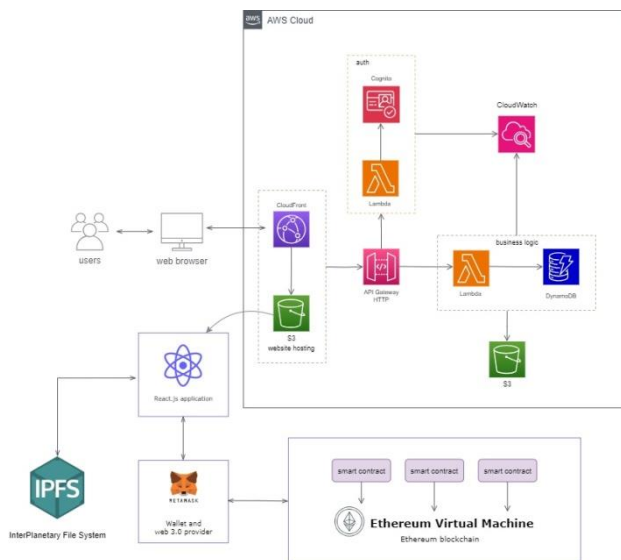
Када се поставља инфраструктура неког система важно је организовати компоненте на добар начин

како би се ефикасније и агилније могло управљати њоме. Како су временом комплексност и размере инфраструктуре расле било је јасно да традиционални приступ са својим недостацима више није погодно решење. То је био моменат када почиње да се развија концепт под називом инфраструктура као код (енгл. *Infrastructure as Code – IaC*) који омогућава управљање читавом инфраструктуром помоћу конфигурационих фајлова написаних у коду [11].

AWS Cloud Development Kit представља радни оквир за дефинисање и обезбеђивање инфраструктурних AWS ресурса користећи познате програмске језике.

4. АРХИТЕКТУРА СИСТЕМА

Платформа је развијена као децентрализована веб 3.0 апликација, са одвојеним клијентским и серверским делом који остварују међусобну комуникацију путем HTTPS протокола. И клијентска и серверска апликација су постављене на AWS облак – S3 сервис је употребљен за хостовање статичког веб сајта, док је бекенд развијен као serverless апликација (слика 3).



Слика 3. Дијаграм архитектуре решења.

Клијентска страна је имплементирана коришћењем React.js библиотеке у оквиру које је употребљен web3.js скуп библиотека захваљујући коме је омогућена интеграција са дигиталним новчаницима и веб 3.0 провајдерима (у овом случају MetaMask). Апликација комуницира и са IPFS (енгл. *InterPlanetary File System*) мрежом, ради складиштења датотека које је корисник унео, као и са чворовима блокчејн мреже, како би позивала функције паметног уговора за креирање и добављање корисникових незамњивих токена.

Бизнис логика серверског дела је представља Lambda функцијама које комуницирају са NoSQL базом података, DynamoDB. Када се захтев упуту серверу он прво стиже на API Gateway који ће на основу детаља из захтева окинути одговарајућу Lambda функцију. Ауторизација и аутентификација корисника је имплементирана коришћењем Cognito сервиса. За

мониторинг Lambda функција у решење је интегрисан и AWS CloudWatch сервис чији је задатак да прати и прикупља метрике коришћених ресурса.

Компонента која чини апликацију децентрализованом јесте Ethereum блокчејн мрежа над којом се извршавају паметни уговори.

5. ИМПЛЕМЕНТАЦИЈА РЕШЕЊА

Обе апликације (клијентска и серверска) су смештене у AWS окружењу, а инфраструктурни ресурси су подигнути уз помоћ AWS Cloud Development Kit радног оквира са којим је веза остварена кроз програмски језик Go. Сви ресурси се налазе у eu-central-1 региону.

5.1. Клијентска апликација

Клијентска страна представља *single-page* апликацију (енгл. SPA – *Single page application*) и доступна је на Интернету захваљујући интеграцији два сервиса – Amazon S3 и Amazon CloudFront. Комбинација две поменуте платформе је изузетно честа и може се рећи да представља стандард за испоруку фронтенд апликација.

5.2. Серверска апликација – serverless решење

Lambda функције представљају језгро инфраструктуре серверске стране јер са собом носе код специфичан за одређени део пословне логике. Пројекат је организован тако да је свака функционалност представљена са по једном Lambda функцијом, у засебном .go фајлу. На листингу 1 је дата генеричка структура коју поштује сваки од фајлова из lambda пакета.

```
func Handler(ctx context.Context, req
events.APIGatewayV2HTTPRequest)
(events.APIGatewayV2HTTPResponse, error) {
    // business logic
}

func main() {
    lambda.Start(Handler)
}
```

Листинг 1. Структура фајла једне Lambda функције.

Одговарајућа Lambda ће се окинути када са клијентске стране стигне захтев на API Gateway намењен за њу. Након што функција заврши процесирање вратиће на API Gateway одговор који ће бити прослеђен крајњем кориснику. Како би обавила своју функционалност, Lambda функција позива методе из одговарајућих сервиса који комуницирају са слојем намењеним за приступ бази података.

Да би се ресурси сместили на AWS облак то десило потребно је покренути следећу команду у терминалу:

cdk deploy

Да би ова команда имала ефекта и да би уопште могла да се изврши, мора да постоји конфигурациони фајл cdk.json. Пример фајла је дат на листингу 2.


```

{
  "app": "go mod download && go run
  backend.go",
  "watch": {
    "include": [
      "*"
    ],
    "exclude": [
      "cdk*.json",
      "go.mod",
      "go.sum"
    ]
  },
  "context": {
    "@aws-cdk/aws-
    lambda:recognizeLayerVersion": true,
    "@aws-cdk/core:checkSecretUsage": true

    // the rest of the code is omitted
  }
}

```

Листинг 2. Пример *cdk.json* датотеке.

Датотека се генерише при иницијализацији CDK пројекта (уз команду **cdk init**) и садржи наредбе неопходне да се инфраструктура покрене и испоручи (енгл. *deploy*) у AWS окружењу.

6. ЗАКЉУЧАК

Рад се бавио применом концепата рачунарства у облаку и serverless технологије као део њега, како би се имплементирао систем за управљање курсевима. Применом једне од популарних платформи облака (AWS) као и serverless архитектуре главне предности система јесу ефикасност трошкова и оптимизација ресурса. Основни недостатак система из блокчејн перспективе јесте присуство традиционалне серверске апликације која има посредничку улогу између клијентског интерфејса и паметних уговора. Са стране облака може да се истакне безбедност као главна брига, јер су подаци поверени провајдерима услуга, те снажне безбедносне мере морају постати императив. Појава праксе развоја у облаку је постала данас битан фактор у модерном развоју софтвера. Пројекат описан у раду је само један од примера трансформације традиционалних архитектуралних решења под утицајем рачунарства у облаку. Ово поље рачунарства је у константном процесу еволуције, а нове технологије попут рачунарства на рубу (енгл. *edge computing*), serverless архитектуре, блокчејна и услуга покретаних вештачком интелигенцијом су у стању да преобликују хоризонт могућности.

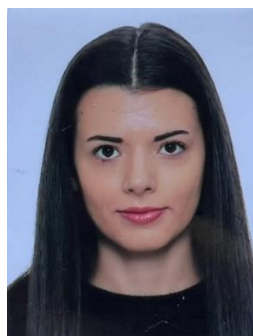
7. ЛИТЕРАТУРА

- [1] S. J. Bigelow, "TechTarget," 15 November 2022. [На мрежи]. Доступно: <https://www.techtarget.com/whatis/feature/The-history-of-cloud-computing-explained>. [Последњи приступ децембар 2023].
- [2] "modex.tech," 6 December 2021. [На мрежи]. Доступно: [https://www.modex.tech/blog/centralized-vs-](https://www.modex.tech/blog/centralized-vs-decentralized-vs-distributed-systems)

[decentralized-vs-distributed-systems](#). [Последњи приступ децембар 2023].

- [3] M. v. Steen, Distributed Systems (4th edition).
- [4] Montazerolghaem, Ahmadreza, Yaghmaee, M. Hossein, Leon-Garcia and Alberto, Green Cloud Multimedia Networking: NFV/SDN Based Energy-Efficient Resource Allocation, 2020.
- [5] A. S. Ravikiran, "simplelearn.com," October 2023. [На мрежи]. Доступно: <https://www.simplilearn.com/tutorials/blockchain-tutorial/blockchain-technology>. [Последњи приступ децембар 2023].
- [6] "Evolution of Blockchain Technology," 2 November 2022. [На мрежи]. Доступно: <https://www.nationthailand.com/business/corporate/40021641>. [Последњи приступ децембар 2023].
- [7] B. Varghese, "History of the cloud," 19 March 2019. [На мрежи]. Доступно: <https://www.bcs.org/articles-opinion-and-research/history-of-the-cloud/>. [Последњи приступ децембар 2023].
- [8] "What is serverless computing?," [На мрежи]. Доступно: <https://www.cloudflare.com/learning/serverless/what-is-serverless/>. [Последњи приступ јануар 2024].
- [9] "Serverless Architecture Overview," [На мрежи]. Доступно: <https://www.datadoghq.com/knowledge-center/serverless-architecture/>. [Последњи приступ јануар 2024].
- [10] N. Barney, "Amazon Web Services," [На мрежи]. Доступно: <https://www.techtarget.com/searchaws/definition/Amazon-Web-Services>. [Последњи приступ јануар 2024].
- [11] "What Is Infrastructure as Code?," [На мрежи]. Доступно: <https://stackify.com/what-is-infrastructure-as-code-how-it-works-best-practices-tutorials/>. [Последњи приступ јануар 2024].

Кратка биографија:



Марија Петровић је рођена 9. априла 1999. године у Новом Саду. Факултет техничких наука у Новом Саду, студијски програм Рачунарство и аутоматика уписала је 2018. године. Након завршених основних студија, 2022. године, уписала је мастер академске студије из исте области на смеру Рачунарство високих перформанси. контакт: petrovic.ma9@gmail.com

RAZVOJ BLOKČEJN PODSISTEMA ZA VEB 3.0 PLATFORMU ZA UPRAVLJANJE KURSEVIMA**DEVELOPMENT OF THE BLOCKCHAIN SUBSYSTEM FOR A WEB 3.0 PLATFORM FOR COURSE MANAGEMENT**

Stefan Ljubović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Rad se bavi razvojem veb 3.0 aplikacije za upravljanje kursovima primenom računarstva u oblaku. Sadrži opis primena blokčejna za plaćanje i kreiranje NFT sertifikata pomoću Solidity-a, IPFS-a kao i AWS CDK serverless- infrastrukture. Rad uključuje i opis implementacije ovih tehnika izvršene u programskom jeziku Go za backend i komunikaciju sa IPFS veb 3 memorijom kao i korišćenje Solidity-a za pametne ugovore.

Ključne reči: blokčejn, pametni ugovori, solidity, AWS CDK, serverless

Abstract – The work deals with the development of a course management application using cloud computing. It contains a description of blockchain applications for payment and creation of NFT certificates using Solidity, IPFS as well as AWS CDK serverless-infrastructure. The paper includes a description of the implementation of these techniques in the Go programming language for the backend and communication with IPFS web3 storage, as well as the use of Solidity for smart contracts.

Keywords: blockchain, smart contracts, Solidity, AWS CDK, serverless

1. UVOD

Tehnologija blokčejna je donela revoluciju u različite industrije, nudeći inovativna rešenja za dugotrajne izazove. U ovom radu dublje istražujemo praktične primene tehnologije blokčejna, fokusirajući se posebno na platne sisteme i kreiranje Non-Fungible Tokena (NFT). Razmatramo kako se Solidity, programski jezik koji se često koristi za razvoj pametnih ugovora na blokčejn platformama, može iskoristiti za ove svrhe. Takođe, koristimo snagu AWS CDK (Cloud Development Kit) i serverless infrastrukture kako bismo stvorili skalabilno okruženje za naša rešenja bazirana na blokčejnu.

Dok blokčejn napreduje, važno je razumeti kako se može primeniti za unapređenje sistema za plaćanja i uvođenje novih koncepta poput NFT-ova. U ovom radu ćemo ne samo razgovarati o teoriji već i detaljno opisati kako ove tehnike možemo primeniti koristeći programski jezik Go. Ova kombinacija tehnologija pruža uzbudljive mogućnosti za inovacije.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dušan Gajić, vanr. prof.

Na kraju rada, čitaoci će bolje razumeti praktične primene blokčejn tehnologije i kako je mogu iskoristiti u različitim oblastima.

2. POJAM BLOKČEJN TEHNOLOGIJE

Blokčejn predstavlja poseban oblik distribuiranih sistema, tačnije, jednu od mogućih implementacija tehnologije poznate kao Distribuirana glavna knjiga (DLT). Pametni ugovori su karakterističan deo blokčejn tehnologije i značajno unapređuju njenu upotrebu. U nastavku ćemo pružiti pregled ključnih pojmova koji su od suštinskog značaja za razumevanje blokčejna.

Pored ovih osnovnih konceptata, važno je napomenuti da AWS CDK (Cloud Development Kit) može biti koristan alat u kontekstu razvoja blokčejna, pojednostavljujući kreiranje i upravljanje beskrajinim infrastrukturnim resursima za blokčejn aplikacije. Takođe, pomenućemo i kreiranje Truffle projekta i proces dizanja pametnih ugovora pomoću ovog popularnog razvojnog okvira.

2.1. Distribuirani sistemi

Distribuirani sistemi predstavljaju arhitektonski model u kojem računarski resursi i funkcionalnosti nisu centralizovani na jednom mestu, već su raspoređeni i povezani preko mreže [1]. Ova decentralizovana struktura omogućava veću otpornost na kvarove, skalabilnost i efikasniju upotrebu resursa. Distribuirani sistemi se često koriste u različitim oblastima, uključujući cloud računarstvo, mreže, bankarstvo i naravno, blokčejn tehnologiji. Ovaj koncept igra ključnu ulogu u razumevanju i implementaciji tehnoloških rešenja današnjeg doba.

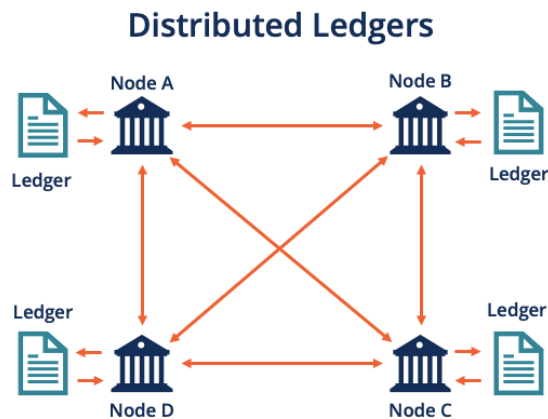
2.2. Distribuirana glavna knjiga

Tehnologija distribuirane knjige (DLT) omogućava visoko dostupnu bazu podataka održavanu fizički razmeštenim uređajima, tzv. čvorovima, u nepouzdanom okruženju. DLT obećava povećanje efikasnosti i transparentnosti u saradnji između pojedinaca i organizacija, nudeći otpornost na neovlašćeno korišćenje podataka, sigurnost, nepromenljivost i anonimnost [2].

Primene DLT-a uključuju lanac snabdevanja, finansije i zdravstvenu zaštitu. Na primer, u lancu snabdevanja, DLT se koristi za stvaranje sigurne skladišne platforme, koja se replicira preko više čvorova, omogućavajući sigurno skladištenje podataka i automatizaciju poslovnih procesa. Međutim, implementacija aplikacija na DLT-u suočava se s kompromisima između različitih karakteristika, poput

dostupnosti i doslednosti. Izbor odgovarajućeg DLT dizajna postaje ključan, s obzirom na specifične zahteve aplikacije, ali i uzimajući u obzir održivost aplikacije tokom vremena.

U suštini, DLT služi kao zajednička digitalna infrastruktura za visoko dostupne, distribuirane baze podataka u nepouzdanom okruženju, gde se podaci prenose i dodaju knjizi u obliku transakcija. Validacija transakcija se vrši putem digitalnih potpisa i kriptografije javnog ključa.



Slika 1. Distribuirana glavna knjiga

2.3. Blokčejn kao implementacija DLT-a

Blokčejn funkcioniše kao distribuirana knjiga, sastavljena od niza šifrovanih blokova koji sadrže informacije o validnoj mrežnoj aktivnosti. Svaki blok je vezan za prethodni, stvarajući nepromenljiv i transparentan lanac podataka. Ova tehnologija omogućava bezbednu razmenu vrednosti između entiteta putem interneta, sa čim eliminiše potrebu za posrednicima. Decentralizovana priroda blokčejna omogućava postizanje saglasnosti i dogovora u mreži bez centralnog upravljanja, što doprinosi transparentnosti i poverenju. Rudarenje, proces dodavanja blokova u lanac putem validacije od strane čvorova u mreži, igra ključnu ulogu u održavanju integriteta podataka. Rudari su nagrađeni za dodavanje validnih blokova, što ih motiviše da održavaju konzistentnost podataka i obezbede sigurnost mreže [3].

2.4. Pametni ugovori

Pametni ugovori, predloženi od strane Nika Szabo-a u 1990-ima, koriste tehnologiju blokčejna za automatsko izvršavanje ugovornih klauzula kada se ispune unapred definisani uslovi. Ovi ugovori smanjujući rizik, administrativne troškove i povećavajući efikasnost poslovnih procesa. Njihova nepromenljivost u blokčejnu pruža transparentnost i smanjuje zlonamerne radnje. Ipak, postoje izazovi poput pitanja privatnosti izvršenja ugovora i potrebe za obezbeđivanjem ispravnosti ugovora zbog mogućih grešaka u računarskim programima. [4]

2.5 Ethereum virtuelna mašina

Analogija "distributed ledger" često se koristi za opisivanje blokčejna, poput Bitcoina, koji koristi osnovne alate kriptografije za decentralizovanu valutu. Registar

vodi evidenciju aktivnosti uz pravila koja određuju šta se može ili ne može uraditi da bi se izmenio registar. Ethereum, sa svojom kriptovalutom Ether, funkcioniše kao raspodeljena mašina stanja koja omogućava izvršavanje pametnih ugovora. Ethereum-ova virtuelna mašina (EVM) definiše pravila za promenu stanja iz bloka u blok. [5]

EVM se opisuje kao funkcija prelaska stanja $Y(S, T) = S'$, gde S predstavlja staro stanje, T su transakcije, a S' je novo stanje. Ethereum koristi veliku strukturu podataka, modifikovani Merkle Patricia Trie, za čuvanje računa povezanih heševima.

Transakcije su kriptografski potpisane instrukcije od strane računa, a postoje dva tipa: one koje rezultuju pozivima poruka i one koje rezultuju kreiranjem ugovora. Ugovori sadrže Merkle Patricia skladišni trie i izvršavaju se kao niz EVM operacija koje obavljaju standardne operacije steka, poput XOR, AND, SABIRANJE, ODUZIMANJE. EVM takođe implementira blokčejm-specifične operacije steka, kao što su ADRESA, SALDO, BLOKHEŠ.

Implementacije EVM-a uključuju Py-EVM (Python), evmone (C++), ethereumjs-vm (JavaScript), eEVM (C++), revm (Rust), a sve moraju poštovati specifikaciju iz Ethereum Yellowpaper-a. Tokom devetogodišnje istorije Ethereum-a, EVM je prošao kroz revizije, a različite implementacije postoje u različitim programskim jezicima.

2.6 IPFS

IPFS (InterPlanetary File System) je modularni set protokola za organizaciju i prenos podataka, osmišljen sa principima adresiranja sadržaja i peer-to-peer mrežnog povezivanja. IPFS ima više implementacija, ali njegova glavna svrha je decentralizovano objavljivanje podataka, uključujući datoteke, direktorijume i web lokacije. [6]

IPFS se sastoji od implementacije protokola, decentralizovane mreže čvorova i modularnog skupa protokola za organizaciju i prenos podataka. Ipak, važno je napomenuti da IPFS nije sam provajder za čuvanje podataka, niti provajder cloud usluga.

IPFS funkcioniše tako što podatke predstavlja blokovima sa jedinstvenim identifikatorom, poznatim kao Content Identifier (CID), koji se generiše kombinovanjem heša podataka sa njihovim kodikom. Za rad sa CID-ovima i adresiranim sadržajem, IPFS koristi InterPlanetary Linked Data (IPLD) i Merkle DAG (usmeren aciklički graf). Za adresiranje fajlova, direktorijuma i simboličkih veza, IPFS koristi UnixFS. [4]

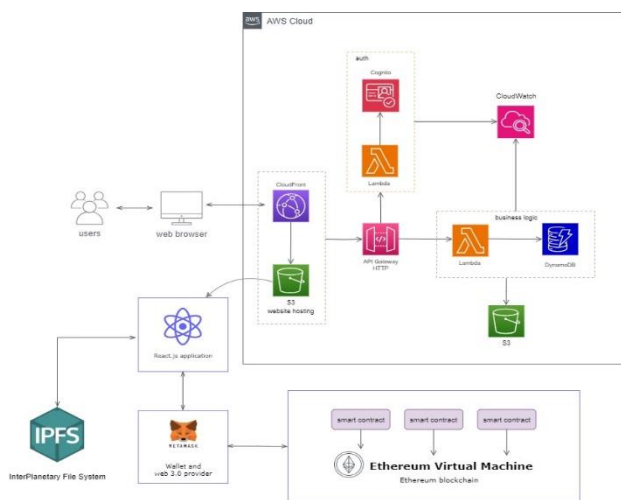
IPFS koristi Kademliju, distribuiranu tabelu sa heširanim ključem (DHT), kako bi pronašao čvorove u mreži koji čuvaju tražene podatke. Kademlija je efikasan sistem otporan na promene čvorova. IPFS čvorovi koriste Bitswap, porukama baziran, peer-to-peer mrežni protokol za razmenu podataka.

Ukratko, IPFS omogućava decentralizovano objavljivanje podataka kroz adresiranje sadržaja, peer-to-peer mrežno povezivanje i efikasne protokole za organizaciju i prenos podataka.

3. ARHITEKTURA SISTEMA

Platforma je koncipirana kao decentralizovana aplikacija za veb 3.0, s jasno odvojenim klijentskim i serverskim segmentima koji ostvaruju međusobnu komunikaciju putem HTTPS protokola.

Obe aplikacije, kako klijentska tako i serverska, smeštene su na AWS oblaku – S3 servis koristi se za hostovanje statičkog veb-sajta, dok je backend razvijen kao serverless aplikacija (vidi Sliku 2).



Slika 2. Dijagram arhitekture rešenja

Na klijentskoj strani, aplikacija je implementirana pomoću React.js biblioteke, koja se izvršava u veb pregledaču. Takođe, unutar nje koristi se web3.js biblioteka za integraciju s digitalnim novčanicima i veb 3.0 provajderima (u ovom slučaju, MetaMask). Aplikacija takođe ostvaruje komunikaciju s IPFS (InterPlanetary File System) mrežom za skladištenje korisničkih datoteka i s čvorovima blokčejn mreže za pozivanje funkcija pametnih ugovora.

Poslovna logika serverskog dela ostvarena je Lambda funkcijama, koje komuniciraju s NoSQL bazom podataka DynamoDB. DynamoDB sadrži podatke o entitetima sistema koji nisu deo decentralizovane mreže. Kad se zahtev uputi serveru, prvo prolazi kroz API Gateway, koji na osnovu detalja iz zahteva okida odgovarajuću Lambda funkciju. Autorizacija i autentifikacija korisnika implementirane su putem Cognito servisa, koristeći Cognito User Pool koji sadrži sve informacije o registrovanim korisnicima. Za praćenje Lambda funkcija, u rešenje je integrisan AWS CloudWatch servis, koji prati i prikuplja metrike korišćenih resursa.

Centralna komponenta koja čini aplikaciju decentralizovanom jeste Ethereum blokčejn mreža, nad kojom se izvršavaju pametni ugovori. U ovom kontekstu, klijentska strana je odgovorna za komunikaciju s čvorovima Ethereum mreže. Pametni ugovori su kreirani i deployovani na Ethereum platformu pomoću Truffle radnog okvira.

4. IMPLEMENTACIJA REŠENJA

4.1 Klijentska strana

Za implementaciju klijentske strane korišten je Reakt.js (engl. React.js) radni okvir, zajedno sa veb3.js (engl. web3.js) skupom biblioteka koje omogućavaju integraciju sa digitalnim novčanicima. Na primer, kao što je MetaMask (engl. MetaMask) novčanik ugrađen u veb pretraživač, a omogućava da korisnici povežu svoj Eterijum nalog sa nalogom na sistemu i na taj omogući da korisnici interaguju sa pametnim ugovorima. MetaMask je Eterijum provajder (engl. provider).

4.2 Truffle

Truffle je razvojno okruženje i radni okvir za testiranje blokčejn sistema na Ethereum virtuelnoj mašini (EVM). Instalacija Truffle paketa, pisanje pametnih ugovora u Solidity jeziku, i konfiguracija Truffle projekta ključni su koraci u procesu postavljanja pametnih ugovora na Ethereum mrežu.

Pametni ugovori se pišu u Solidity jeziku, a zatim kompajliraju naredbom `truffle compile`. Skripte za postavku ugovora smeštene su u `migrations/` direktorijumu, numerisane su i izvršavaju se kako bi ugovori bili postavljeni na mrežu. Konfiguracija se vrši u `truffle-config.js` datoteci, gde se koristi `hdwallet-provider` za potpisivanje transakcija.

Konačno, postavka pametnih ugovora na Sepolia test mrežu izvršava se komandom `truffle migrate --network sepolia`. Sve ove korake olakšava Truffle okruženje, omogućavajući brži razvoj i postavljanje pametnih ugovora na Ethereum mrežu. U nastavku sledi listing pametnog ugovora, kao i izvršavanje migracije:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.9;

import
"@openzeppelin/contracts/token/ERC721/ERC721.sol";
import
"@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/utils/Counters.sol";

contract CryptoCert is ERC721URIStorage, Ownable {
    constructor() ERC721("Crypto Cert", "CRC") {}
    using Counters for Counters.Counter;
    Counters.Counter private _tokenIds;

    function mintNFT(address recipient, string memory tokenURI)
        public onlyOwner
        returns (uint256)
    {
        _tokenIds.increment();
```



```

uint256 newItemId = _tokenIds.current();
_mint(recipient, newItemId);
_setTokenURI(newItemId, tokenURI);

return newItemId;
}
}

```

Listing 1. primer pametnog ugovora

4.3 Serverska strana

Serverska strana sistema koristi Amazonove serverless servise, a ključni deo je čuvanje podataka na decentralizovanom skladištu IPFS (InterPlanetary File System). IPFS je odabran kao rešenje za čuvanje datoteka, uključujući metapodatke NFT-ova (Non-Fungible Token) zbog efikasnosti i ograničenja kapaciteta skladištenja blokčejna.

Većina decentralizovanih aplikacija na Ethereum platformi kombinuje blokčejn i offchain skladište podataka poput IPFS-a. Datoteke i metapodaci o NFT-ovima čuvaju se na IPFS-u, dok se jedinstveni identifikatori (ID) tokena čuvaju na blokčejnu-u. Ovo poboljšava efikasnost mreže.

U radu se koristi kombinacija Golang i JavaScript klijentskih biblioteka za web3.storage servis. Kroz implementaciju funkcije ReceiveCertificate, sertifikati se obrađuju nakon završetka kursa. Slika sertifikata se preuzima sa S3 bucket-a, a zatim se kreira CID za sliku na Web3 Storage-u. Generišu se metapodaci za sertifikat, skladište na IPFS-u, i zatim se koristi za kreiranje NFT-a na frontend delu sistema. Na kraju, funkcija vraća URL gde se može pronaći finalni NFT sertifikat.

5. ZAKLJUČAK

Zaključak ovog rada sumira sveobuhvatno istraživanje i implementaciju blokčejn tehnologije, pametnih ugovora i decentralizovanih aplikacija. Kroz projekat Truffle, koji je poslužio kao osnova za razvoj Ethereum pametnih ugovora, istražena je serverless arhitektura koristeći Golang i AWS CDK, što naglašava prilagodljivost modernih tehnologija oblaka u podršci blokčejn aplikacijama.

Centralni deo istraživanja bio je fokus na korišćenju IPFS-a za skladištenje metapodataka i slika povezanih sa NFT sertifikatima u Web3 skladištu. Ovaj inovativni pristup obezbeđuje decentralizovanu i nepromenljivu prirodu podataka, integrišući efikasno IPFS i Web3 skladište. Kroz implementaciju frontend-a koristeći React i MetaMask, pružen je korisnički orijentisan pristup interakciji sa Ethereum blokčejnom.

U radu je naglašen značaj blokčejn tehnologije u stvaranju decentralizovanih i transparentnih sistema. Integracija Ethereum-a, IPFS-a, Web3 skladišta i AWS CDK-a u projektu predstavlja primer pristupa razvoju blokčejn aplikacija, kombinujući infrastrukturu blok lanca, logiku pametnog ugovora i interakciju frontend-a sa Ethereum platformom.

Ovaj rad ima za cilj da doprinese razumevanju i primeni aplikacija zasnovanih na blokčejnu, nudeći opipljiv plan za buduće projekte u brzo-razvijajućem okruženju decentralizovanih aplikacija.

6. LITERATURA

- [1] N. Barney, "Distributed ledger technology (DLT) TechTarget".
- [2] "Distributed Systems," March 2018. [Online]. Available: <http://www.dgma.donetsk.ua/docs/kafedry/avp/metod/van%20Steen%20-%20Distributed%20Systems.pdf>.
- [3] S. K. и. H. G. H. Natarajan, "Distributed Ledger Technology and Blockchain," FinTech Note, 2017.
- [4] N. Szabo, "Smart Contracts," [Online]. Available: <https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html>. [Accessed February 2024].
- [5] ethereum.org, "EVM," [Online]. Available: <https://ethereum.org/en/developers/docs/evm/>.
- [6] "How IPFS works," [Online]. Available: <https://ipfs.io/>. [Accessed 2 2024].

Kratka biografija:



Stefan Ljubović rođen je u Subotici 1999. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika odbranio je 2022.god.

kontakt: ljubovicstefan@gmail.com

**ПРЕДИКЦИЈА КАШЊЕЊА АВИОНСКИХ ЛЕТОВА КОРИШЋЕЊЕМ
АЛГОРИТАМА МАШИНСКОГ УЧЕЊА****PREDICTION OF FLIGHT DELAYS USING MACHINE LEARNING ALGORITHMS**Катарина Жерајић, Јелена Сливка, *Факултет техничких наука, Нови Сад***Област – РАЧУНАРСТВО И АУТОМАТИКА**

Кратак садржај – У овом раду се истражује проблем кашњења авионских летова. У развијенијим државама где кашњења авионских летова могу да представљају значајан финансијски губитак, постоје институције које се баве праћењем и анализом овог проблема. У циљу одређивања фактора који утичу на кашњење авионских летова, обучени су модели машинског учења. За предикцију су коришћени су подаци о летовима, авионима, аеродромима и временским условима у време лета. Кашњења су подељена у три класе: занемарљиво кашњење (до 15 минута), мало кашњење (између 15 и 60 минута) и велико кашњење (преко 60 минута).

Кључне речи: предвиђање кашњења летова, експлоративна анализа података, модел система, KNN, SVM.

Abstract – This paper tackles the problem of flight delays. In more developed countries, where flight delays can lead to significant financial loss, institutions are founded to monitor and analyze this problem. This paper analyses the factors that influence flight delays by training machine learning models on data on flights, planes, airports, and weather conditions at the time of flight. Flight delays are divided into three classes: negligible delays (up to 15 minutes), small delays (between 15 and 60 minutes), and long delays (over 60 minutes).

Keywords: flight delay prediction, exploratory data analysis, system model, KNN, SVM.

1. УВОД

У данашње време, кашњење авионских летова је постало честа појава, што може изазвати низ проблема за путнике и за авио-компаније.

За авио-компаније, кашњење летова може утицати на пад репутације уколико нису у могућности да адекватно управљају ситуацијом и пруже алтернативне аранжмане за путнике који су погођени кашњењем лета.

Уколико би имали приступ поузданим информацијама о потенцијалним проблемима који би могли изазвати кашњење летова, авио-компаније би могле да правовремено осмисле алтернативе за своје путнике,

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је била др Јелена Сливка, ванр. проф.

што би повећало њихово задовољство и позитивно утицало на углед авио-компаније.

Предикција кашњења летова би била од велике користи и за путнике, омогућавајући им да се припреме за могуће изазове и да испланирају алтернативе у случају да дође до кашњења летова.

С друге стране, ова врста предикције може имати и шире бенефите за индустрију, помажући у бољем управљању капацитетима и ресурсима на аеродромима, као и у побољшању целокупног транспортног система.

Циљ овог рада је да се, применом машинског учења, утврди веза између података о авионским летовима, временским условима и карактеристикама авиона са могућим кашњењем лета. У том циљу су прикупљени јавно доступни подаци о авионима, аеродромима и летовима, као и подаци о временским условима. Да би се утврдили фактори који имају највећи утицај на кашњење авиона, извршена је експлоративна анализа података. На основу резултата експлоративне анализе изабран је скуп обележја за тренирање модела за предикцију.

У наставку рада биће детаљно објашњени различити аспекти решавања проблема. У поглављу 2 се налази осврт на радове који се баве сличном тематиком. Поглавље 3 садржи опис скупа података, анализе спроведене над добијеним подацима и описује алгоритме коришћене за предикцију кашњења авионских летова. Поглавље 4 садржи дискусију на тему резултата добијених на основу формираног модела. На крају, поглавље 5 закључује овај рад.

2. ПРЕТХОДНА РЈЕШЕЊА

Претрагом радова на тему предикције кашњења авионских летова, пронађено је неколико радова са различитим приступом у решавању овог проблема.

Рад [1] се бави предикцијом одлагања авионских летова на територији САД-а на основу података о авиону, укрцавању путника и терета у комбинацији са подацима о лету. Модели машинског учења који су коришћени за предикцију су: *Decision Tree*, *Random Forest*, *Extra Trees*, *Bagging*, *Gradient Boosting* и *XGBoost*. За сваки модел су израчунате тачност, одзив и Ф-мера. *XGBoost* класификатор је дао најбоље резултате, праћен *Random Forest* алгоритмом. На основу најбоље остварених резултата одлучено је да се у овом раду користе *XGBoost* класификатор и *Random Forest* алгоритам. Поред тога, користиле се методе евалуације које су коришћене у раду [1].

Рад [2] је анализирао кашњења летова Делта авио-компаније. Коришћени су подаци о летовима, који укључују термин лета, аеродром поласка, аеродром доласка, пријеме доласка, разлика између постигнутог и жељеног времена доласка, број летова и удаљеност између аеродрома. Аутори рада [2] су експериментисали са следећим моделима машинског учења: *Gradient Boosting Classifier*, *Decision Tree*, *Naive Bayes*, *SVM*, *Random Forest*, *Extra Trees*, *Bagging*, *logistic regression*. За евалуацију модела су користили прецизност, тачност, одзив и специфичност. Најбољи резултати остварени су употребом *Gradient boosting* класификатора, што иде у прилог одлуци да се и у овом раду користи *XGBoost*.

Рад [3] је истраживао утицај временских услова на кашњење авионских летова. У ту сврху су аутори користили податке о летовима, временским условима и гужвама на терминалу. Експериментисали су са следећим моделима машинског учења: *Decision Tree*, *neural networks*, *SVM*. Решење је евалуирано поделом на три подскупа на основу параметара временских услова. Утврђено је да нема велике разлике у резултатима који су постигнути применом горе наведених алгоритама. С обзиром да су резултати за сва три алгорита слични, одлучено је да *SVM* буде један од алгорита за предикцију коришћених у овом раду.

3. МЕТОДОЛОГИЈА

У овом поглављу је представљена имплементација система за предикцију кашњења авионских летова.

Као улаз у систем коришћена је комбинација више скупова података. Међу њима су подаци о авионима, аеродромима и летовима, као и подаци о временским условима. Циљна варијабли, односно излаз из система је класа кашњења којој неки лет припада. Скуп података описан је у поглављу 3.1.

Након прикупљања података, извршена је експлоративна анализа како би се одредила адекватна обележја за предикцију кашњења лета. Поступак експлоративне анализе је описан у поглављу 3.2.

Над креираним скупом података су тренирани модели машинског учења приказани у поглављу 3.3.

3.1. Скупови података

Скуп података који је коришћен у овом раду је произведен од стране Завода за статистику саобраћаја (енг. *Bureau of Transportation Statistics*) [4] у Сједињеним Америчким Државама. Овај скуп садржи три различита скупа података о комерцијалним летовима из 2015. године, организованих у CSV датотеке. Први скуп података садржи информације о авио-компанијама, други скуп података садржи информације о аеродромима у Сједињеним Америчким Државама, а трећи и најзначајнији скуп података садржи информације о летовима који су организовани у току 2015. године. Овај сет садржи информације о више од пет милиона летова, а сваки лет је описан 31 атрибутом.

На основу радова анализираних у поглављу 2, изведен је закључак да би старост и тип авиона могли да имају

утицај на предикцију кашњења летова. Стога су прикупљени подаци са сајта о авионима [5] који на основу вредности *TAIL_NUMBER* обележја, враћа податке о карактеристикама авиона. Ови подаци подразумевају информације о типу и старости авиона.

Подаци о временским условима за полазни и долазни аеродром из 2015. године су прикупљени позивањем *API*-ја сајта о временским условима [6].

Након добављања података, следећи корак је чишћење података, што подразумева следеће фазе:

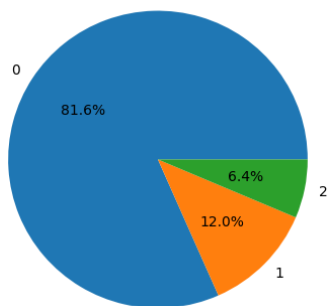
- **Уклањање недостајућих вредности.** Формирани скуп података о летовима има преко пет милиона редова, па је одлучено је да редови са недостајућим вредностима циљног обележја буду уклоњени.
- **Смањивање скупа података.** Одлучено је да се скуп података смањи са пет милиона на око 10 000 узорака због спорости прикупљања података о авионима и временским условима. При смањењу скупа података је коришћена стратегија узимања сваког *k*-тог узорака. Да би добијени скуп података био права слика већег скупа од којег је добијен, било је потребно сортирати полазни скуп по вредностима циљног обележја. Након сортирања скупа података, да би се добило 10 000 узорака, узет је сваки 500-ти узорак из сортираног скупа података.
- **Категоризација циљног обележја.** Након смањивања скупа података, примјењен је велики распон вредности циљног обележја (од -80 до 2000). Извршена је категоризација циљног обележја у три категорије:
 - 0 категорија - занемарљиво кашњење (мање од 15 минута)
 - 1 категорија - мало кашњење (од 15 до 60 минута)
 - 2 категорија - велико кашњење (више од 60 минута)
- **Уклањање аутлајера.** Након категоризације циљног обележја, примјењено је да постоје обележја чије вредности доста одскачу од осталих вредности. На основу графика вредности обележја је утврђено за која обележја да се уклоне вредности које одскачу (енг. *outliers*).

3.2. Експлоративна анализа и обрада података

Циљ експлоративне анализе података је да се разумеју и истраже подаци како би се откриле кључне карактеристике и трендови у скупу података.

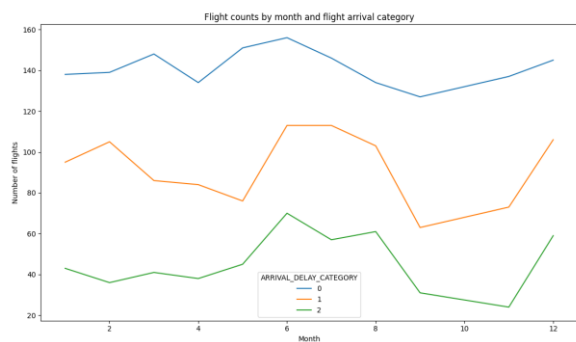
Први корак у фази експлоративне анализе био је утврђивање односа између категорија кашњења летова. На слици 1 се налази процентуална расподела летова према категоријама кашњења.

На слици 1 види се да највећи проценат скупа података чине летови који су имали кашњење мање од 15 минута. Ово значи да је скуп података неизбалансиран, што може негативно утицати на методе класификације.



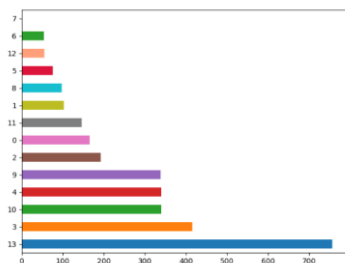
Слика 1. Расподела кашњења према категоријама

На слици 2 приказан је график на коме се види зависност између месеца у години и броја кашњења летова у свакој од категорија кашњења. Можемо закључити да је у летњим месецима (6 - 9 месеца) највећи број кашњења по свим категоријама, па је из тог разлога одлучено да обележје MONTH буде укључено у предикцију.



Слика 2. Расподела кашњења по категорији у односу на месец у години

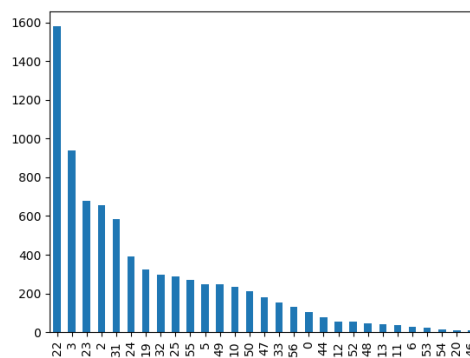
На слици 3 приказан је утицај авио-компаније на кашњење летова. На графику се види да једна компанија има убедљиво највише кашњења, самим тим је одлучено да се обележје AIRLINE узме у обзир приликом предикције кашњења авиона.



Слика 3. Расподела кашњења према авио-компанији

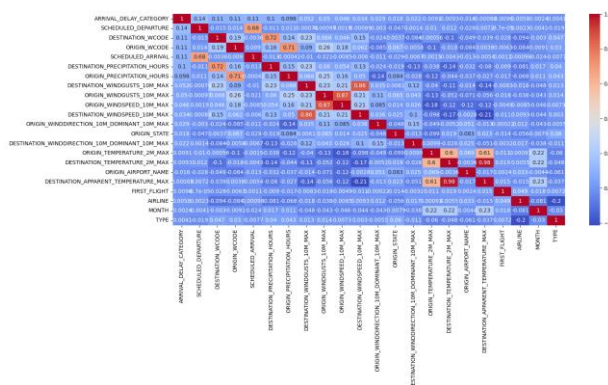
На слици 4 се може видети број летова који касне за сваки тип авиона. Такође је видљиво да се неколико типова авиона издвајају по броју кашњења. Због тога је обележје TYPE узето као једно од обележја за тренирање модела.

Као последњи корак у експлоративној анализи, се посматра зависност обележја, како би се утврдило која обележја се могу искористити за обучавање модела. Зависност обележја се може утврдити анализом матрице корелације почетног скупа обележја.



Слика 4. Расподела кашњења према типу авиона

На слици 5 је приказана матрица корелације за обележја која су изабрана за предикцију на основу експлоративне анализе и анализе нумеричких вредности корелација са циљним обележјем.



Слика 5. Матрица корелације одабраног скупа обележја

3.3. Коришћени модели машинског учења

Ово поглавље дискутује неколико модела машинског учења који су коришћени за предикцију кашњења у авионском саобраћају:

- К-најближих комшија (енг. *K-nearest neighbour* - KNN)
- Метод случајне шуме (енг. *Random forest* - RF)
- XGBoost класификатор (XGB)
- Машине потпорних вектора (енг. *Support Vector Machine* - SVM)

KNN (*K-Nearest Neighbors*) модел се користи за класификацију и регресију. Основна идеја модела је да, ако два објекта имају сличне карактеристике, они припадају истој категорији.

Random Forest је модел машинског учења који се састоји од више стабала одлучивања која се генеришу случајним избором узорка података и случајним избором карактеристика. Свако стабло даје своје предвиђање, а коначно предвиђање се добија агрегацијом предвиђања појединачних стабала одлуке.

XGBoost класификатор је базиран на идеји градијентног *boosting*-a, који је поступак тренирања више слабијих модела како би се постигла боља тачност предвиђања.

SVM има за основну идеју проналажење хипер-равни која најбоље раздваја различите класе података. Хипер-раван се бира тако да буде максимално удаљена од најближих тачака различитих класа.

4. РЕЗУЛТАТИ И ДИСКУСИЈА

Први корак у тренирању сваког од модела је подјела података на тренинг, валидациони и тест скуп. Скупови су подељени у односу 80% тренинг скуп, 10% валидациони скуп и 10% тест скуп.

Након оптимизације параметара, која је вршена на валидационом скупу података, извршено је обучавање модела помоћу тренинг скупа података. За евалуацију су коришћене следеће метрике: Ф1-мјера, одзив, тачност и прецизност. У табели 1 приказани су резултати предикције кашњења авионских летова над тест скупом података.

Табела 1 - Резултати предикције кашњења авионских летова

Модел	Тачност	Прецизност	Одзив	Ф1 - мјера
XGB	80%	65%	80%	72%
KNN	80%	74%	80%	72%
RF	80%	65%	80%	72%
SVM	80%	65%	80%	72%

Као што је видљиво у табели 1, сваки од модела који је коришћен за предикцију дао је сличне резултате - Ф1-мјера износи 72%, док тачност износи 80% за све моделе.

У раду [1] максимална вредност Ф-мере која је постигнута износи 58%, што је горе од резултата добијених у овом раду. У раду [2] Ф-мера није коришћена као метрика евалуације. Умјесто тога, коришћена је AUC (Area Under the Curve) метрика, где је добијена вредност од 72%.

Ако упоредимо почетни скуп података коришћен у научним радовима [1] и [2] са скупом података коришћеним у овом раду, можемо примјетити да они садрже мање обележја, што је главни разлог слабијих резултата.

У научном раду [3] коришћена је CSI (Critical Success Index) метрика и постигнута је вредност од 84%.

5. ЗАКЉУЧАК

Циљ овог рада била је анализа кашњења авионских летова. Одрађена је експлоративна анализа да би се утврдило која обележја имају највећи утицај на циљно обележје (класа кашњења). На основу резултата експлоративне анализе изабран је скуп обележја за тренирање модела за предикцију. Резултати који су постигнути приказани су у поглављу 4. Као што је видљиво у табели 1, резултати су слични за све моделе.

Иако су остварени резултати доста добри у поређењу са научним радовима [1], [2] и [3], остављен је простор за унапређење рада. Како 80% скупа података чине летови са занемарљивим кашњењем, потенци-

јално унапређење би било балансирање скупа података применом техника за *under-* и *over-sampling*.

Такође, контрола ваздушног саобраћаја (ATC - Air traffic control) може имати утицаја на кашњење авионских летова, па би значајно унапређење било укључити и информацију о стању ваздушног простора у тренирање модела за предикцију. Осим тога, за предикцију кашњења лета авиона би се могла искористити и информација о претходном лету на који се дати лет конектује, јер постоји могућност ланчане реакције, уколико је претходни лет каснио.

6. ЛИТЕРАТУРА

- [1] Mustafa Kurt (2019), MEF university, Flight delay prediction
<https://openaccess.mef.edu.tr/xmlui/bitstream/handle/20.500.11779/1217/MustafaKurt.pdf>
- [2] Enwew Chibuike Kenneth (2019), National College of Ireland, A Machine Learning approach predicting flight arrival delay reduction for Delta Airlines
<https://norma.ncirl.ie/4305/1/chibuikekennethenwere.pdf>
- [3] Deepak Kulkarni, Yao Wang and Banavar Sridhar (2013), Ames Research Center, Data mining for understanding and improving decision-making affecting ground delay programs
<https://permanent.access.gpo.gov/gpo52894/20140008891.pdf>
- [4] Kegggle, 2015 Flight Delays and Cancellations,
<https://www.kaggle.com/datasets/usdot/flight-delays>
- [5] FlightAware, Flight Tracker/Flight Status
<https://flightaware.com>
- [6] Open-Meteo, Free Open-Source Weather API
<https://open-meteo.com>

Кратка биографија:



Катарина Жерајић рођена је 20.10.1999. у Невесињу, БиХ. Године 2022 завршила је основне студије на Факултету техничких наука на коме брани и мастер рад 2024 године из области Електротехнике и рачунарства - Рачунарство и аутоматика. Школске 2022/23 године је радила као сарадник у настави на Факултету техничких наука. Године 2023 започиње рад као предавач на “ФТН информатици”, и као “јава девелопер” у фирми CAKE.

контакт: katarinazer6@gmail.com

МОДЕЛ ВРЕМЕНСКЕ СЕРИЈЕ ЗА ПРЕДВИЂАЊЕ ЦЕНА AWS СПОТ ИНСТАНЦИ TIME SERIES MODEL FOR PREDICTING AWS SPOT INSTANCE PRICES

Милица Прањкић, Јелена Сливка, *Факултет техничких наука, Нови Сад*

Област – РАЧУНАРСТВО И АУТОМАТИКА

Кратак садржај – Растућа потражња за рачунарским ресурсима подстакла је коришћење провајдера рачунарства у облаку. То је довело до развоја нове димензије у којој везе између употребе ресурса и трошкова морају бити разматране са организацијске перспективе. Као део своје EC2 (Elastic Compute Cloud) услуге, Амазон је увео спот инстанце као јефтину јавну инфраструктуру која долази са ценом непоузданости услуге. Цена спот инстанци је кључни фактор у управљању трошковима и оптимизацији ресурса у рачунарству у облаку. Динамичко одређивање цена може закомпликовати процес планирања и алокације ресурса. Стога, тачна предикција цена спот инстанци може бити од велике користи за кориснике облака, омогућавајући им да максимално искористе своје ресурсе и минимизирају трошкове.

Кључне речи: спот инстанце, временске серије, модел за предвиђање цене, SARIMA.

Abstract – The growing demand for computing resources has spurred the use of cloud computing providers. This has led to the development of a new dimension where the relationship between resource usage and cost must be considered from an organizational perspective. As part of its EC2 (Elastic Compute Cloud) service, Amazon introduced spot instances as a cheap public infrastructure at the price of service unreliability. The price of spot instances is a key factor in cost management and resource optimization in cloud computing. Dynamic pricing can complicate the process of planning and resource allocation. Therefore, accurate prediction of spot instance prices can greatly benefit cloud users, allowing them to maximize their resources and minimize costs.

Keywords: spot instances, time series, price prediction model, SARIMA

1. УВОД

У Амазоновом моделу спот инстанци (Spot Instance, SI), изнајмљене инстанце могу се нагло прекинути од стране провајдера услуга када је то потребно. Интерфејс за управљање SI базира се на стратегији лицитације која се базира на Амазоновим тајним ценовним стратегијама, што отежава корисницима примену стратегија распоређивања или пружања ресурса заснованих на таквим (јефтинијим) ресурсима.

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је била др Јелена Сливка, ванр. проф.

Иако се верује да употреба EC2 SI инфраструктуре може смањити трошкове за крајње кориснике, детаљни преглед литературе показује да њихове карактеристике и могућности још увек нису дубоко истражене.

AWS пружа могућност прегледа историјских података о промени цена спот инстанци у претходних 90 дана. Ова историја, корисницима пружа детаљан увид у флукуацију цена спот инстанци током одређеног периода. У оквиру овог истраживања разматран је процес формирања одговарајућег модела за одређивање цена SI на основу прикупљених историјских података. Циљ модела је да омогући предвиђање будуће цене SI током времена, како би се лоцирао SI ресурс доступан по ниској цени.

Овај рад презентује SARIMA (Seasonal Autoregressive Integrated Moving Average) модел временске серије за предикцију цена SI у будућности. Скуп података над којим је модел обучен су историјски подаци о ценама SI у претходних шест месеци.

Историјски подаци су подељени у тренинг и тест скуп, где се тренинг подаци користе за обуку модела, а тест подаци за његову евалуацију. За процену тачности предвиђања коришћене су различите метрике које пореде стварне вредности цене SI у датом временском тренутку са њеном вредношћу предвиђеном од стране модела временске серије.

Остатак овог рада организован је на следећи начин. Поглавље 2 даје преглед радова који се баве предикцијом цена SI. Поглавље 3 излаже теоријске основе примењених модела временских серија. Поглавље 4 описује решење за предикцију цена SI представљено у овом раду. Поглавље 5 описује експерименте извршене у циљу евалуације овог решења, а резултати евалуације су приказани и дискутовани у поглављу 6. Поглавље 7 закључује рад.

2. ПРЕГЛЕД СТАЊА У ОБЛАСТИ

Од објављивања спот инстанце на Амазону 2009. године, многе студије су се бавиле њиховим истраживањем. Лиу и други [1] су ове студије поделили у две главне категорије, на основу метода на којима се базирају:

- модели за предвиђање цена SI на основу статистичке временске анализе (поглавље 2.1.1) и
- модели за предвиђање цена SI применом машинског учења (поглавље 2.1.2).

2.1 Модели предвиђања цена на основу статистичке временске анализе

У раду [2], *SARIMA* модел се успоставља анализирајући историјске цене спот инстанце за претходних 11 месеци. Применом различитих хеуристика за моделовање тренинг скупа података, откривено је да *SARIMA* алгоритам постиже значајно боље перформансе приликом предвиђања цене *SI* у односу на размотрене алтернативе. *SARIMA* модел временске серије показује 17% већу тачност у поређењу са резултатима других предиктивних алгоритама.

2.2 Модели предвиђања цена на основу машинског учења

У једном од скоријих истраживања, Конг и други [3] износе став да методе машинског учења нису добре за предвиђање циљне варијабле за проблеме временских серија. *RNN* (*Recurrent Neural Network*) постиже добре перформансе на проблемима временских серија, али тренирање овог модела проблематично због појаве проблема експлозије градијента и нестанка градијента. *LSTM* (*Long Short-Term Memory network*) је позната варијанта *RNN*-а која решава проблем нестанка градијента, експлозије градијента и недостатка дугорочне способности меморије.

Стога се *LSTM* може ефективно користити за временске серије. *GRU* (*Gated Recurrent Unit*) је варијанта *LSTM*-а која поједностављује унутрашњу *LSTM* структуру, а постиже готово једнаке перформансе.

У раду [3], додато је неколико потпуно повезаних слојева у циљу побољшања тачности *GRU* модела у предвиђању нелинеарних података о ценама. Такође, коришћен је модел насумичне шуме (*Random Forest*), метода машинског учења, за предвиђање података о ценама *SI*, ради поређења.

2.2 Одабир модела за предвиђање цене спот инстанце

Преглед стања у области указује на значајне предности алгоритама машинског учења за предикцију временских серија. Међутим, статистички модели временских серија, попут *SARIMA* модела, имају боље перформансе у случајевима [6]:

- Када су подаци имају ограничен опсег вредности, при чему су те вредности ниске
- Када је количина података за обуку мала
- За краткорочна предвиђања
- У случају линеарних временских серија.

Због велике заступљености статистичких алгоритама машинског учења у предикцији цена *SI* у претходним радовима, овај рад ће се фокусирати на истраживање перформанси статичких модела. Такође, количина историјских података над којом ће се вршити обука није велика, правиће се краткорочна предвиђања и цене *SI* су релативно мале вредности, ограничене одређеним опсегом.

Стога, статички модели машинског учења представљају добар избор за овај проблем. Једини услов који историјски подаци не задовољавају, у овом случају, јесте линеарност. Историјски подаци о

ценама инстанци су нелинеарна структура, где сваки податак не може бити објашњен на основу претходних података.

Добијени резултати биће евалуирани на сличан начин као у радовима [3] и [5]. Такође, резултати овог истраживања ће бити упоређени са резултатима из наведених истраживања.

3. ТЕОРИЈСКИ ПОЈМОВИ И ДЕФИНИЦИЈЕ

Како се цене спот инстанци мењају током одређеног временског периода, класификоване су као временска серија (*time-series*). Временска серија је секвенцијални скуп података, мерених у узастопним временским периодима. Мерења прикупљена током догађаја у временској серији сортирају се у одговарајући хронолошки редослед.

Један од најпопуларнијих и најчешће коришћених стохастичких модела временских серија је модел *ARIMA* (*Autoregressive Integrated Moving Average*). За прогнозирање сезонских временских серија, *Бокс* и *Џенкинс* [4] су предложили прилично успешну варијанту *ARIMA* модела, *SARIMA* (*Seasonal ARIMA*).

Бокс-Џенкинс приступ изградњи модела временских серија састоји се од три фазе, које се понављају док се не добије задовољавајући модел: спецификација модела, процена параметара и дијагностичка провера.

У овом раду коришћен је *SARIMAX* модел. *SARIMAX* се разликује у односу на *SARIMA* модел по могућности укључивања утицаја спољних фактора на посматрани модел. Спољне или егзогене варијабле су оне на које се не може директно утицати, али могу имати ефекат на понашање посматраног модела.

4. МЕТОДОЛОГИЈА

Ово поглавље представља имплементацију модела временске серије за предикцију цена *AWS SI*. За формирање и обучавање модела коришћени су историјски подаци о цени *SI* у периоду од 6 месеци. Очекивани излаз из модела је будућа цена *SI*. Као модел машинског учења се користи *SARIMAX* модел.

У првом кораку, формира се скуп од почетних тренинг података, који обухвата првих неколико тачака временске серије и користи се за обуку модела.

Затим, тренирани модел предвиђа наредну вредност из временске серије, тј., вредност која следи одмах након последње вредности у подацима за обуку.

Након тога, вредност коју је модел покушао да предвиди, додаје се у скуп тренинг података – не додаје се предвиђена, већ стварна, измерена вредност.

Скуп тренинг података је тиме проширен да обухвата једну додатну тачку.

Овај поступак се понавља за сваку преосталу тачку у временској серији, чиме се добија итеративни поступак обуке који побољшава своје перформансе у сваком кораку (слика 1).



Слика 1. Дијаграм тока експеримента

5. ЕКСПЕРИМЕНТИ

5.1 Скупови података

Историјски подаци о ценама спот инстанци преузети су са AWS веб странице. Историја цена доступна је за период од 90 дана. Подаци су прикупљени у два наврата, у размаку од 3 месеца, те скуп података обухвата историју цена у периоду од 180 дана. Посматрани регион је: *us-east-1*, укључујући све његове подрегионе и све типове *SI* доступне у том региону.

Структура прикупљених података представљена је следећим обележјима: зона доступности, тип инстанце, платформа инстанце, цена инстанце и временска ознака узорковања. С обзиром на то да цена уме знатно да варира у зависности од сва три параметра, извршено је груписање скупа података према следећим параметрима: *us-east-1a* (зона доступности), *t3.2xlarge* (тип инстанце), и *Windows* и *Linux* (платформа). Као резултат овог груписања формирана су два засебна скупа података који садрже информације о промени цене спот инстанце која задовољава одабране параметре груписања.

Подаци из поменутих скупова су сортирани у растућем редоследу на основу временске ознаке узорковања, што значи да су историјске цене инстанци поређане хронолошки. Затим, подаци су груписани у тренинг (90%) и тест скуп (10%).

5.2 Експеримент 1

У оквиру овог експеримента, *SARIMAX* моделу допремљен је цео скуп тренинг података над којима је вршена обука. Примарни циљ овог експеримента је, биоло креирање модела предикције цена *SI*.

Секундарна ствар која се кроз овај експеримент може показати јесте начин на који се развија тачност предвиђања модела током времена. Другим речима, утврдити да ли се квалитет предвиђања побољшава или погоршава како време одмиче.

Након што је модел обучен, извршено је предвиђање цена за неколико тачака у будућности, тј., неколико наредних корака. Број корака је одабран насумично, али пракса је показала да, што се више тачака одабере, то ће се боље приказати и објаснити реално стање система и квалитета дугорочне предикције.

5.3 Експеримент 2

У оквиру овог експеримента, коришћена је техника померајућег прозора за предвиђање. Другим речима, предвиђање је вршено један корак унапред, након чега је резултат тог предвиђања увршћен у тренинг скуп, а модел је поново подвргнут поступку обуке. Овај процес је мање захтеван за меморију, али, обучавање модела траје знатно дуже. Изузетно је ефикасан за предвиђање временских серија и састоји се из неколико корака описаних у поглављу 4.

Након што је поступак обуке модела завршен, добијена су два скупа: скуп предвиђених вредности и скуп стварних вредности. Како би се установила ефикасност модела, ова два скупа су упоређена и над њима су примењене одређене метрике, наведене у наставку (поглавље 5.3).

6. РЕЗУЛТАТИ И ДИСКУСИЈА

У оквиру првог експеримента, модел је обучен над читавим тренинг скупом, а након завршетка процеса учења вршена је предикција будућних вредности. Предикција је вршена за 20 наредних корака.

На почетку, модел показује релативно добре способности предикције. Међутим, како се временски интервал предикције повећава, тако модел све више губи на тачности и показује све лошије перформансе.

Вредности средње апсолутне процентуалне грешке (*MAPE*), приказане у табели 1, представљају просечну грешку модела при предвиђању у односу на стварне вредности. Грешка је изражена у процентима, а почетна вредност код првог корака је 0.0869%, што означава релативно добру прецизност предвиђања у том тренутку. Међутим, како време одмиче и како се повећава број корака у предвиђању, модел почиње да губи своју прецизност. На пример, у 20. кораку процентуална грешка је достигла 2.6332%.

Табела 1. Вредности *MAPE* за различите кораке предикције у експерименту 1

Корак	<i>MAPE</i>
1.	0.0869
2.	0.0324
...	...
19.	2.2682
20.	2.6332

Експеримент 2с, који користи технику померајућег прозора описану у поглављу 4, извршава предикцију

за све вредности из тест скупа. На почетку, модел показује врло добре способности предикције (Табела 2). Тачност модела варира током овог поступка, али апсолутна вредност грешке остаје врло мала до самог краја. Шта више, грешка се смањује како време одмиче.

У експерименту 2, грешка коју модел прави врло је мала од самог почетка. Иако постоје одређене варијације *MAPE* вредности кроз итерације, модел је био у стању да одржи добру тачност у великом броју случајева. Поред тога, тачност овог модела се повећава како време одмиче, што може деловати обећавајуће када је реч о дугорочним предикцијама.

Табела 2. Вредности *MAPE* за различите кораке предикције у експерименту 2с

Корак	MAPE
1.	0.0869
2.	0.1195
...	...
61.	0.0174
62.	0.0398

6.1 Дискусија

Из Табеле 3 јасно се уочава да је техника померајућег прозора правила мање грешке у односу на модел обучен над тренинг који даје предикције за више временских корака у напред.

Табела 3. Вредности *MSE*, *RMSE*, *MAE* и *R-SQUARED* за дату модел

	MSE	RMSE	MAE	R ²
SARIMAX	1.2e-05	0.0035	0.00261	-14.4864
Техника померајућег прозора	2.6e-07	0.0005	0.00042	0.98588

Резултати овог истраживања потврђују теорију изложу у раду [5], која истиче губитак перформанси предикције SARIMAX модела приликом предвиђања дужег временског периода унапред.

Сходно чињеници да је SARIMAX статички модел, показао је веома добре перформансе приликом краткорочне предикције. То га чини конкурентним у односу на алтернативне моделе машинског учења.

7. ЗАКЉУЧАК

У овом раду представљен је систем за предикцију цена спот инстанци помоћу SARIMA модела временске серије. Мотивација проистиче из чињенице да се област рачунарства у облаку рапидно развија и постаје кључни део информатичке инфраструктуре бројних организација. Цена спот инстанци је кључни фактор у управљању трошковима и оптимизацији ресурса у рачунарству у облаку.

Динамичко одређивање цена може закомпликовати процес планирања и алокације ресурса. Стога, тачна предикција цена спот инстанци може бити од велике користи за кориснике облака, омогућавајући им да максимално искористе своје ресурсе и минимизирају трошкове.

У овом раду је за предикцију цена спот инстанци коришћен SARIMAX модел за предвиђање временских серија. Извршена су два експеримента која су пружила увид у перформансе модела приликом краткорочне и дугорочне предикције. Евалуација модела извршена је израчунавањем средње апсолутне процентуалне грешке (*MAPE*), која квантификује разлику између стварних и предвиђених вредности и омогућава да се тачно утврди ефикасност модела.

Када је реч о статичким моделима временске серије за предикцију, делује да је SARIMAX најбоља опција. Међутим, из јасно наведених недостатака, не мора значити да ће он бити адекватна опција приликом сусретања са различитим типовима проблема. Иако је модел показао добре перформансе приликом краткорочних предвиђања, пре његове употребе у продукцији, требало би истражити да ли је овакво ограничење прихватљиво за корисника.

8. ЛИТЕРАТУРА

- [1] Wenqiang Liu†, Pengwei Wang*†, Ying Meng, Caihui Zhao and Zhaohui Zhang, Cloud spot instance price prediction using kNN regression
- [2] Alkharif S, Lee K, Kim H (2018) Time-series analysis for price prediction of opportunistic Cloud computing resources. 2018 7th international conference on emerging databases. Springer, Singapore, pp 221–229
- [3] Dawei Kong, Shijun Liu, Li Pan, Amazon Spot Instance Price Prediction with GRU Network
- [4] G.E.P. Box, G. Jenkins, "Time Series Analysis, Forecasting and Control", Holden-Day, San Francisco, CA, 1970.
- [5] Vaia I. Kontopoulou 1, Athanasios D. Panagopoulos 2,* , Ioannis Kakkos 1 and George K. Matsopoulos 1, A Review of ARIMA vs. Machine Learning Approaches for Time Series Forecasting in Data Driven Networks
- [6] Paliari, I.; Karanikola, A.; Kotsiantis, S. A comparison of the optimized LSTM, XGBOOST and ARIMA in Time Series forecasting. In Proceedings of the 2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA), Chania Crete, Greece, 12–14 July 2021; pp. 1–7.

Кратка биографија:



Милица Прањкић рођена је 1997. године у Новом Саду. Основне академске студије завршила је 2020. године на Факултету техничких наука, на ком брани и мастер рад 2024. године из области Примењене рачунарске науке и информатика - Интелигентни системи. контакт: m.pranjkic@gmail.com

СУМАРИЗАЦИЈА НАУЧНИХ РАДОВА НА СРПСКОМ ЈЕЗИКУ ПРИМЕНОМ NLP МЕТОДА**SUMMARIZATION OF SCIENTIFIC PAPERS IN SERBIAN LANGUAGE USING NLP METHODS**

Наташа Ивановић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У раду је представљен систем за сумаризацију научних радова на српском језику применом NLP метода са циљем да се посао истраживача олакша кроз аутоматско генерисање апстракта. Решење је имплементирано кроз два модула – фаза екстракције (TextRank алгоритам) и фаза апстракције (Sequence-to-Sequence модели), где се трансформер модел показао као најбољи избор.

Кључне речи: Аутоматска сумаризација текста, NLP, Sequence-to-Sequence модели, Трансформер модели, TextRank алгоритам

Abstract – The paper presents a system for summarizing scientific papers in Serbian using NLP methods, aiming to facilitate researchers' work through automatic abstract generation. The solution is implemented in two modules – the extraction phase (TextRank algorithm) and the abstraction phase (Sequence-to-sequence models), for which the Transformer model has proven to be the best choice.

Keywords: Automatic text summarization, NLP, Sequence-to-Sequence models, Transformer models, TextRank algorithm

1. УВОД

Системи за сумаризацију текста представљају интегрални део различитих решења у оквиру обраде природног језика (NLP – енгл. *Natural Language Processing*). Задатак сумаризације текста има за циљ очување кључних информација и контекста, редукујући оригинални текст на краћу верзију [1]. Овај рад се фокусира на генерисање сажетака научних радова. Апстракт представља кратак опис научног рада који садржи кључне информације о истраживању [2] - има значајну улогу, јер истраживачи често на основу садржаја апстракта одлучују да ли ће наставити са читањем остатка рада.

Систем за генерисање апстракта има потенцијал да олакша посао истраживача и помогне у ефикаснијем и бржем комуницирању кључних аспеката рада. Аутоматски генерисани апстракти могу пружити објективнији приказ рада и побољшати конзистентност у стилу и садржају [2].

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Александар Ковачевић, ред. проф.

У овом раду представљено је решење за аутоматско генерисање апстракта на српском језику применом екстракције и апстракције текста. У првом кораку користи се TextRank алгоритам [3] за копирање, односно екстракцију најбитнијих реченица из рада, које служе као улаз у фазу апстракције. У сврхе апстракције користе се неуронске мреже засноване на енкодер-декодер архитектури [4], у циљу генерисања нових фраза које нису обавезно део оригиналног текста. У наставку биће анализирано стање у области, а затим ће бити дат преглед методологије, извршених експеримената и њихових резултата.

2. СТАЊЕ У ОБЛАСТИ

Интересовање за аутоматску сумаризацију текста датира још из половине 20. века. Ручно писање апстракта подложно је субјективној процени истраживача – аутори рада [5] предлажу решење које се заснива на копирању најзначајнијих реченица ранжираних на основу броја понављања често коришћених речи и кључних речи, речи у оквиру наслова и поднаслова, као и на основу саме позиција реченице у тексту. Старији приступи ослањају се на хеуристику и прецизно дефинисане параметре. У научним радовима се исте информације формулишу на различите начине у зависности од поглавља. Тешко је избећи дубликате у генерисаним апстрактима примењујући овакав приступ.

У раду [6] представљена је примена TextRank алгоритма у сврхе екстрактивне сумаризације текста писаног на српском језику. Аутори истичу да су досадашња истраживања у овом домену већински рађена над текстовима на енглеском и да је стога овај проблем изазван за решавање. Алгоритам рачуна косинусну сличност међу реченицама и генерише матрицу сличности, која се затим конвертује у граф, где су реченице чворови, а гране скорови сличности. Финални сажетак представља топ три ранжиране реченице. Решење је евалуирано применом ROUGE метрике.

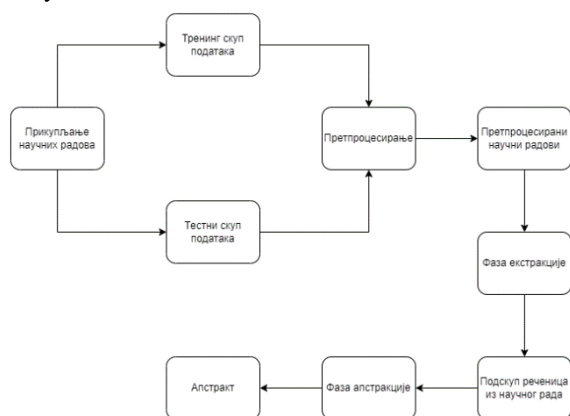
Рани приступи апстрактној сумаризацији укључивали су (1) *sentence compression*, чији циљ је да редукује дужину реченице уклањајући садржај који није есенцијалан; (2) *sentence fusion* где се комбинује неколико независних реченица у једну кохерентну целину и (3) *sentence revision* током ког се генеришу реченице које нису део оригиналног текста [7]. Данас се за потребе апстракције користе *sequence-to-*

sequence модели са енкодер-декодер архитектуром. Енкодери и декодери су најчешће имплементирани као *RNN* (енгл. *Recurrent Neural Network*). Енковање дугачких секвенци и даље је отворени проблем за ове моделе. Архитектура је унапређена увођењем *attention* механизма. Кључна идеја је идентификација значајних речи и фраза у документу и прослеђивање те информације декодеру [7].

У раду [8] представљен је комбиновани приступ сумаризацији текста, уз нагласак да овакав приступ значајно унапређује добијене резултате. Како би се избегао проблем предугачких докумената, примењује се корак екстракције. Користе се два модела који се заснивају на енкодер-декодер архитектури. Код првог модела се за енкодер користе два *Bi-LSTM*-а, а за декодер један *LSTM*. За други модел, користе се два *LSTM*-а. За потребе апстракције кориштен је *GPT* трансформер модел. Кориштена су два јавно доступна скупа података: *arXiv* и *PubMed*. Кориштена је *ROUGE* метрика за евалуацију. Комбиновани приступ аутоматској сумаризацији текста уз коришћење трансформер модела доноси обећавајуће резултате.

3. МЕТОДОЛОГИЈА

Сумаризација текста одвија се у две фазе. Прво се примењују екстрактивне методе за издвајање најзначајнијих реченица, а затим се тај међу-резултат прослеђује у фазу апстракције. Улаз у систем представља научни рад, а излаз из система представља апстракт. Слика 3.1 приказује груб преглед архитектуре решења. У наставку овог поглавља биће детаљније објашњени кораци обраде научних радова и модули система.



Слика 3.1. Преглед архитектуре решења

3.1. Прикупљање података

Скуп научних радова писаних на српском језику преузет је скрепљовањем званичне *web* странице Зборника радова Факултета техничких наука¹ помоћу *selenium* библиотеке. Разлог за овај одабир лежи у томе што већина радова прати предефинисан формат. Иницијално је прикупљено 2335 радова. Након анализе њихове структуре и форматирања, избачени су они који су одскакали од стандарда. После чишћења скупа података преостало је укупно 2215 радова који су подвргнути процесу екстракције и

апстракције. Због ограничења рачунарских ресурса креиран је и помоћни скуп података, меморијски мање захтеван, где је идеја да се сумаризује апстракт и генерише наслов рада. Циљ овог приступа је да се испрати у коликој мери дужина и комплексност текстуалног садржаја утиче на перформансе модела.

3.2. Претпроцесирање података

У процесу претпроцесирања података радови су конвертовани из *.pdf* формата у *.txt* формат. За сваки рад спроведено је издвајање апстракта, који се касније користе као референтни сажетци током тренирања и тестирања модела. Издвојене су кључне речи, чија сврха је да додатно помогну у евалуацији генерисаних апстракта, анализи грешака и провери да ли је модел успео да схвати главну идеју научног рада.

Апстракт и кључне речи написане на енглеском језику избачене су из текстуалне репрезентације рада. Уклоњено је заглавље рада, наслов, аутор, област истраживања, заједно са следећим поглављима: *литература*, *библиографија*, *биографија* и *захвалност*. Радови написани на ћирилици конвертовани су у латинично писмо.

Текст је очишћен од референци и специјалних карактера за набрајање, као и од *scientific* реченица које садрже математичке симболе и операторе. Знакови интерпункције и дијакритички знакови нису уклањани из текста, како би генерисани сажетак био што веродостојнији у граматичком смислу.

Сваки рад подвргнут је процесу издвајања наслова поглавља применом анализе стилова. Испоставило се да већина наслова прати форматирање *Times New Roman*, *Bold* и формат *број-тачка-наслов*. За анализу форматирања *.pdf* докумената кориштена је *fitz* библиотека. Мотивација за издвајање наслова лежи у томе што нису сва поглавља подједнако релевантна за апстракт. Анализа комплетног научног рада доприноси бољем схватању идеје и суштине тог рада, али приликом генерисања апстракта фокус треба да буде на најрелевантнијим поглављима за ту сврху. Истраживања су показала да 72% апстракта чине информације енкапсулиране у уводу, закључку и табелама [9].

3.3. Фаза екстракције

Фаза екстракције за циљ има да прекопира најзначајније реченице из научног рада за генерисање апстракта. Резултат ове фазе је редукована текстуална репрезентација научног рада, која служи као улаз за финалну фазу апстракције. Ова оптимизација у смањењу броја реченица значајно смањује време тренирања модела, димензионалност проблема, као и потребу за рачунарским ресурсима.

За потребе екстракције реченица имплементиран је *TextRank* алгоритам [3] који рачуна косинусну сличност између свих парова реченица и примењује се над сваким поглављем научног рада. Генерисани су сажетци од 5, 10, 15 и 20 реченица, где је идеја да се из увода и закључка копира дупло више реченица у односу на остала поглавља. Уколико се деси да алгоритам покуша да копира више реченица него што постоји у поглављу, онда се ради распоређивање броја

¹ <https://www.ftn.uns.ac.rs/ojs/index.php/zbornik>

реченица на преостала поглавља. Уколико је и то немогућ случај, реченице се преузимају из најдужег поглавља.

3.4. Фаза апстракције

Задатак ове фазе је да од текстуалне репрезентације креиране током екстракције генерише финални апстракт, одн. парафразиран текст. У оквиру решења примењена је енкодер-декодер архитектура, која је послужила као полазна структура за даљи развој. Претпроцесирање података за ову фазу извршено је коришћењем *keras*-ове *Tokenizer* библиотеке. На основу идентификованих речи у тренинг скупу података, *Tokenizer* гради речник где свака реч има свој јединствени индекс – кључ је реч из текста, вредност је фреквенција речи. Генерисан речник је кориштен је за конвертовање текста у нумеричку репрезентацију и касније за декодирање, одн. враћање речи у текстуалну репрезентацију. Трениран је *custom word embedding* фитовањем *Word2Vec* модела над целим скупом података, са циљем да се обогати репрезентација речи у *embedding* слоју током процеса тренирања модела за сумаризацију текста.

Имплементирана, истренирана и тестирана су три модела за сврху генерисања сажетака научних радова:

- *vanilla sequence-to-sequence* модел [10];
- *sequence-to-sequence* модел и *attention* [10];
- *google/mt5-small* трансформер модел [11].

Vanilla sequence-to-sequence модел се састоји из енкодера и декодера и представља дуални *RNN* систем. *Sequence-to-sequence* модел са *attention* механизмом представља проширење претходно представљеног модела, где енкодер користи бидирекциони *LSTM* (*BiLSTM*). Излаз енкодера обухвата информације о контексту из оба смера, чиме се добија шире разумевање улазног текста. Декодер такође користи *LSTM* и проширен је *attention*-ом који је дефинисан функцијом *one_step_attention*. Ова функција користи тежинске коефицијенте за одабир важних делова енкодованог текста. Током сваког корака декодирања користи се *attention* механизам.

Оптимизација модела током тренирања спроведена је коришћењем *sparse_categorical_crossentropy* функције губитка. Кориштен је *RMSprop* (енгл. *Root Mean Square Propagation*) оптимизатор за минимизацију *loss* функције током тренинга. Овај одабир није нужно једини прави избор – одлука је донета на основу већ постојећих истраживања која се баве проблемом сумаризације текста.

Трансформер модел *google/mt5-small* изабран је за решавање проблема сумаризације научних радова јер подржава нестандартне језике, укључујући и српски. Јавно је доступан на *Hugging Face* платформи. *Small* верзија одабрана је због меморијских ограничења. Архитектура овог модела није у потпуности позната, с обзиром на то да детаљи имплементације нису објављени, али може се претпоставити да се заснива на сличним принципима који важе за све трансформере. Примењује *self-attention* и *masked self-attention* механизме. Модел је дотрениран на тренинг скупу података научних радова писаних на српском језику. Како је одабрани трансформер предвиђен за

решавање проблема сумаризације текста, одлучено је да се за функцију губитка и оптимизатор користи оно што библиотека подразумевано нуди, а то су *cross-entropy loss* и *Adam optimizer*.

4. РЕЗУЛТАТИ И ДИСКУСИЈА

Како је *Text Rank* алгоритам ненадгледан, не постоје конкретни параметри који утичу на перформансе и успешност алгоритма. У овом раду фокус је на екстрактима који се састоје од 5 реченица - дужи екстракти испоставили су се као неподобни, јер су значајно продужавали време тренирања модела. Алгоритам је успевао да генерише кохерентне сажетке који се састоје од реченица релевантних за апстракт.

Vanilla sequence-to-sequence модел трениран је и тестиран над оба скупа података, где је фокус био на томе како број *LSTM* слојева, величина улазног *batch*-а и број епоха утичу на квалитет генерисаног сажетка. Примењено је да величина *batch*-а није значајно утицала на резултате. Емпиријски је закључено да већи број епоха током тренинг фазе доприноси томе да генерисани сажетци имају кохерентнију структуру и да су семантички и граматички коректнији. Са енкодером који садржи 1 *LSTM* слој и 2500 епоха постигнути су одлични резултати над тренинг скупом података (*rougeL (f1) = 0.7644*), док је током тестирања уочено да је модел генерисао сажетке које је научио током тренинг фазе и додељивао их научним радовима из тест скупа података. Дошло је до *overfitting*-а, али је ипак охрабрујуће што су током фазе тестирања генерисани сажетци барем били у сличном домену као циљни.

Модел са 2 *LSTM*-а слоја истрениран у 200 епоха није постигао боље резултате у односу на претходни модел. Генерисане реченице биле су неисправне и неретко се дешавало да дође до значајног понављања речи. Са повећањем броја *LSTM* слојева у енкодеру јавили су се и проблеми меморијских ограничења, те је стога за модел чији енкодер садржи 3 *LSTM* слоја одлучено да буде трениран над скупом података за генерисање наслова на основу апстракта. Нису добијени задовољавајући резултати – контекстуално нису имали везе са циљним насловом, али су донекле били семантички и граматички коректни. Проширење *sequence-to-sequence* модела *attention*-ом није донело значајне бенефите у погледу резултата, с обзиром на то да је фаза тренирања трајала веома дуго због рачунарске комплексности овог механизма и меморијских ограничења, па је одлучено да тренирање траје свега 30 епоха.

Трансформер модел истрениран у 150 епоха над скупом података који се састоји од научних радова показао се боље у односу на све претходно анализирани моделе. Може се приметити да модел има свест о контексту и да углавном генерише једну или две реченице које затим понавља више пута. Не постоје значајне разлике између тренинг и тест резултата и стиче се утисак да генерисани сажетци садрже део кључних речи научних радова и да је генерална идеја донекле очувана (Табела 4.1). Трансформер модел је такође истрениран током 100

епоха над скупом података који се састоји од наслова радова. Добијени су резултати који имају смисла и контекстуално су повезани са улазним апстрактном и циљним насловом (Табела 4.1). Не постоје значајне разлике у резултатима између тренинг и тест скупа података.

Google/mt5-small, Epochs 150, Scientific Papers Dataset	
Циљ	u ovom radu predstavljena je rekonstrukcija objekta doma vojske u subotici koji nije u funkciji dugi niz godina, a nalazi se u samom centru grada. predložene su konkretne mere transformacije, kako bi prostor postao kvalitetniji i atraktivniji, a samim tim i privlačniji za posetioce.
Израз	u ovom radu opisan je objekat doma vojske u subotici u subotici. opisan je postupak revitalizacije objekta doma vojske u subotici u subotici. opisan je postupak revitalizacije objekta u subotici u subotici u novom sadu.
Циљ	ovaj rad obuhvata analizu postojećeg stanja kao i potrebna istorijska istraživanja sa ciljem obnove vetrenjače i njenog plasiranja u širu javnost. projektom je obuhvaćeno i rešenje enterijera.
Израз	u ovom radu opisan je i opisana konstrukcija vetrenjače u novom sadu. opisan je i opisana konstrukcija koja je vršena na osnovu svih karakteristika i karakteristika koja se koriste za njihovu upotrebu.
Google/mt5-small, Epochs 100, Scientific Titles Dataset	
Циљ	analiza faktora uspešnosti projekata
Израз	analiza faktora na projektima
Циљ	arhitektonska studija multifunkcionalnog objekta na klisi
Израз	projekat multifunkcionalnog objekta na klisi u novom sadu
Циљ	grype skener za testiranje ranjivosti u kontejnerima
Израз	primena rizika od neželjenih napada aplikacije

Табела 4.1 Преглед примера резултата трансформер модела над тест скупом података

На основу претходно приказаних резултата може се закључити да је трансформер најбољи избор за решавање овог проблема. Једино у случају коришћења овог модела постоји конзистентна веза између улаза, генерисаног израза и циљног израза и модел се понаша слично над тренинг и тест подацима. Слободном проценом може се видети да је трансформер генерисао донекле квалитетне апстракте. Што се наслова тиче, може се приметити да постоје генерисани наслови који не одговарају у потпуности циљном, формулација је другачија, али контекст је и даље очуван. Постигнути су бољи резултати над скупом података који се тиче генерисања наслова (Табела 4.2) – модел се боље сналази када је димензионалност проблема мања.

Dataset - model	rouge1 (f1)	rouge2 (f1)	rougeL (f1)
Papers - 1 LSTM	0.1434	0.0156	0.1027
Titles - 3 LSTM	0.0626	0.0037	0.0584
Titles - Attention	0.1707	0.0228	0.1275
Papers – mt5	0.2127	0.0477	0.1688
Titles – mt5	0.2847	0.1641	0.2755

Табела 4.2 Преглед постигнутих резултата

5. ЗАКЉУЧАК

У раду је представљен систем за аутоматску сумаризацију научних радова писаних на српском језику. Решење се састоји од два модула – фазе екстракције (TextRank алгоритам) и фазе апстракције (sequence-to-sequence модели, од којих се најбоље

показао трансформер *google/mt5-small*). Модели су тренирани над два скупа података, од којих су оба креирана ручно, прикупљањем научних радова из ФТН-ове архиве. Како нумеричке вредности не осликавају у најбољој мери квалитет генерисаних сажетака, извршена је ручна евалуација. Закључено је да су апстракти и наслови семантички и граматички коректни и у вези са контекстом научног рада.

Фаза екстракције текста могла би бити унапређена тако да реченице које садрже кључне речи имају већу тежину. Постоји потреба за већим и стандардизованим скуповима података на српском језику. Претпоставља се да би са богатијим скупом података и бољим рачунарским ресурсима модели били боље истренирани. Без обзира на ограничења, може се рећи да добијени резултати представљају добру полазну основу за даља истраживања у домену сумаризације текста на српском језику.

6. ЛИТЕРАТУРА

- [1] Tas, O., & Kiyani, F. (2007). A survey automatic text summarization. *PressAcademia Procedia*, 5(1), 205-213.
- [2] Cachola, I., Lo, K., Cohan, A., & Weld, D. S. (2020). TLDR: Extreme summarization of scientific documents. *arXiv preprint arXiv:2004.15011*.
- [3] Mihalcea, R., & Tarau, P. (2004, July). TextRank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language proc.* (pp. 404-411).
- [4] Keneshloo, Y., Shi, T., Ramakrishnan, N., & Reddy, C. K. (2019). Deep reinforcement learning for sequence-to-sequence models. *IEEE transactions on neural networks and learning systems*, 31(7), 2469-2489.
- [5] Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2), 264-285.
- [6] Kosmajac, D., & Kešelj, V. (2019, March). Automatic text summarization of news articles in serbian language. In *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)* (pp. 1-6). IEEE.
- [7] Lin, H., & Ng, V. (2019, July). Abstractive summarization: A survey of the state of the art. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 33, No. 01, pp. 9815-9822).
- [8] Pilault, J., Li, R., Subramanian, S., & Pal, C. (2020, November). On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 9308-9319).
- [9] Altmami, N. I., & Menai, M. E. B. (2022). Automatic summarization of scientific articles: A survey. *Journal of King Saud University-Computer and Information Sciences*, 34(4), 1011-1028.
- [10] Shi, T., Keneshloo, Y., Ramakrishnan, N., & Reddy, C. K. (2021). Neural abstractive text summarization with sequence-to-sequence models. *ACM Transactions on Data Science*, 2(1), 1-37.
- [11] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Кратка биографија:



Наташа Ивановић рођена је 1. маја 1998. године у Сомбору. Мастер рад на Факултету техничких наука из области Електротехнике и рачунарства – Интелигентни системи одбранила је 2024. године.

VIZUALIZACIJA AUTOSAR LOGGING/TRACING PROTOKOLA VISUALIZATION OF AUTOSAR LOGGING/TRACING PROTOCOL

Mladen Planojević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu obrađeno je jedno rješenje alata za vizualizaciju AUTOSAR logging/tracing protokola. Rješenje omogućava prikaz izvršavanja izvršnih jedinica u okviru sistema u realnom vremenu koji je baziran na AUTOSAR arhitekturi, na osnovu čega je moguće izvršiti analizu i poboljšanje performansi sistema.

Ključne reči: Vizualizacija, AUTOSAR standard, TTEthernet protokol, logging, tracing

Abstract – This paper presents one of potential solutions for implementation of tool for visualisation of AUTOSAR logging and tracing protocol. The tool implements a graphical view of runnables execution within real-time system, based on AUTOSAR architecture. The visualization, generated by tool, helps in analysis conduction and improvement of system performance.

Keywords: Visualization, AUTOSAR standard, TTEthernet protocol, logging, tracing

1. UVOD

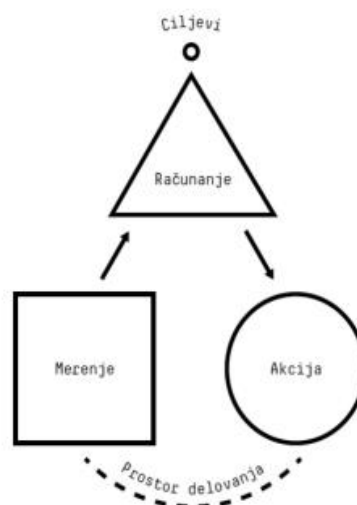
Automobilska industrija danas predstavlja jednu od najperspektivnijih grana industrije. Od svog nastanka, pretrpjela je brojne izmjene, a sve u cilju poboljšanja samog kvaliteta automobila, a time i postizanja sve većeg zadovoljstva putnika, kao krajnjih korisnika. Automobili današnjice predstavljaju veoma kompleksne sisteme, sastavljene od velikog broja komponenti, čiji su zadaci različiti – od pomoći vozaču pri svakodnevnoj vožnji pa sve do potpune autonomnosti i isključivanja vozača iz procesa upravljanja vozilom [1].

Rad pomenutih komponenti automobila zasniva se na principu takozvane kontrolne petlje. Osnovna kontrolna petlja, prikazana na slici 1, sastoji se od senzorskog sistema za prikupljanje informacija o sredini – mjerenje, računarskog sistema za donošenje odluka – računanje, kao i aktuatorskog sistema koji vrši uticaj nad sredinom – akcija [2]. Senzori, poput kamera, radara i lidara igraju ključnu ulogu u realizaciji kontrolne petlje, prikupljajući bitne podatke o različitim parametrima vozila, kao što su brzina, temperatura, pritisak, i slično. Vrijednosti pomenutih parametara se, zatim, obrađuju od strane moćnih procesnih jedinica kako bi se donijele odgovarajuće odluke i izvršile pravovremene akcije posredstvom aktuatora.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Bogdan Pavković, vanr. prof.

Dati proces se odvija neprestano, tokom trajanja cijele vožnje i u realnom vremenu, a razmjena podataka se ostvaruje posredstvom komunikacionih medijuma i protokola.



Slika 1. Osnovna kontrolna petlja [1]

Da bi sastavni elementi kontrolne petlje, prikazane na slici 1, koordinisano i efikasno funkcionisali, potrebno je implementirati pouzdan i efikasan komunikacioni sistem, koji će raditi u realnom vremenu i čija će jedna od glavnih odlika biti determinizam [3]. Pojam determinizma, koji se veoma često sreće u sistemima koji se izvršavaju u realnom vremenu, unosi predvidivost u takve sisteme, odnosno sugerise da se sve aktivnosti, tj. događaji u datom sistemu odvijaju po tačno definisanom rasporedu i u predefinisanoj redoslijedi [4].

Upravo je determinizam jedna od glavnih odlika koja je presudila da TTEthernet protokol postane opšte prihvaćena tehnologija u automobilske industriji [5], a sistem na kom je baziran ovaj rad svoje izvršavanje zasniva na pomenutom protokolu.

Osim determinizma, kao ključnog faktora u komunikacionom sistemu automobila, bitno je pomenuti i AUTOSAR koncept (eng. *Automotive Open System Architecture*), koji pruža standardizovani okvir za razvoj softvera u automobilima. Pomenuti koncept, kreiran od strane istoimenog konzorcijuma, predstavlja globalno priznati standard u automobilske industriji, koji danas u svojim rješenjima koriste neki od najpoznatijih proizvođača automobila i koji je nastao kao odgovor na potrebu da se definiše otvorena i standardizovana arhitektura softvera koja bi omogućila lakšu integraciju različitih

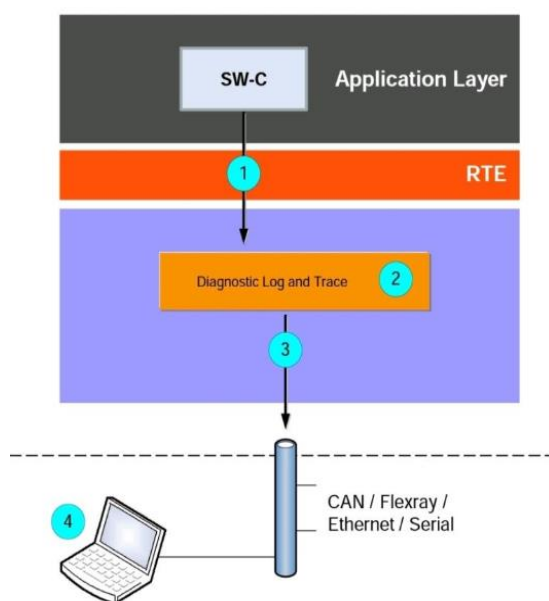
komponenti [6], kao i ponovnu upotrebu softverskih resursa među različitim proizvođačima i modelima automobila [7]. Ono što AUTOSAR standard donosi sa sobom kao novitet u automobilsku industriju jeste slojevita arhitektura, koja omogućava razdvajanje različitih aspekata razvoja automobilskih sistema [8], a glavni koncepti na kojima se bazira pomenuta arhitektura jesu:

- softverska komponenta (eng. *SW-C*, *Software Component*),
- virtualna funkcijska sabirnica (eng. *VBF*),
- sloj apstrakcije softvera (eng. *RTE*, *Run Time Environment*),
- bazični softver (eng. *BSW*, *Basic Software*) i
- izvršna jedinica (eng. *Runnable*).

U sistemima čije se izvršavanje odvija u realnom vremenu i gdje je podrazumijevano determinističko ponašanje, od krucijalne važnosti je postojanje rutina za praćenje performansi i aktivnosti softvera, a ono što, između ostalog, AUTOSAR koncept nudi jesu mehanizmi za logovanje događaja (eng. *logging*), snimanje tragova izvršavanja (eng. *tracing*), te prikupljanje dijagnostičkih informacija koje mogu biti od izuzetne pomoći, pogotovo u slučajevima kad je potrebno otkriti problem u kompleksnim automobilskim sistemima [9].

AUTOSAR *logging/tracing* protokol omogućava inženjerima da efikasno prate i analiziraju ponašanje softvera u realnom vremenu, a takođe ga je moguće jednostavno povezati sa alatima za analizu logova i dijagnostiku grešaka, što doprinosi poboljšanju performansi, pouzdanosti, te održivosti softverskih sistema u automobilu [10]. Sam koncept protokola i njegov položaj unutar slojevite arhitekture AUTOSAR standarda prikazani su na slici 2.

U praksi postoje pojedini alati za prikaz izvršavanja komponenti softvera, odnosno vizualizaciju AUTOSAR *logging/tracing* podataka, poput *Vector CANoe/CANalyzer* ili *Trace.check* alata.



Slika 2. Pozicija *logging/tracing* protokola u okviru AUTOSAR arhitekture

Međutim, dobavljanje licence za pomenuta, komercijalna rješenja je skupo i često neisplativo, a u određenim situacijama izostaje i podrška za dijelove implementacije rješenja, koji su specifični za određeni projekat. Upravo je to poslužilo kao motivacija za razvoj alata za vizualizaciju AUTOSAR *logging/tracing* protokola, čiji će koncepti i programsko rješenje biti objašnjeni u nastavku rada.

2. KONCEPT RJEŠENJA

Ovo poglavlje opisuje koncept rješenja za vizualizaciju AUTOSAR *logging/tracing* protokola. Dat je opis konceptualnih i tehničkih aspekata razvoja, odnosno teorijski prikaz koraka koje je neophodno preduzeti, počev od prikupljanja mjernih rezultata, preko obrade prikupljenih podataka, kako bi se na kraju došlo do rezultata, što je u ovom slučaju prikaz izvršavanja funkcionalnosti sistema u realnom vremenu, na osnovu čega je moguće veoma lako procijeniti stanje i unaprijediti performanse sistema.

• Postavka problema prikupljanja i vizualizacije podataka

Kako je već navedeno u uvodnom dijelu poglavlja, alat za vizualizaciju AUTOSAR *logging/tracing* protokola, predstavlja rješenje višestruke namjene. Primarni cilj alata je vizuelni prikaz redoslijeda i vremena izvršavanja izvršnih jedinica u sklopu automobilskeg softvera, ali osim toga rješenje ima bitnu ulogu u unapređenju procesa otklanjanja grešaka u složenom sistemu, a samim tim i poboljšanje samog softvera. Da bi navedeno bilo realizovano potrebno je da alat ispuni sljedeće zahtjeve:

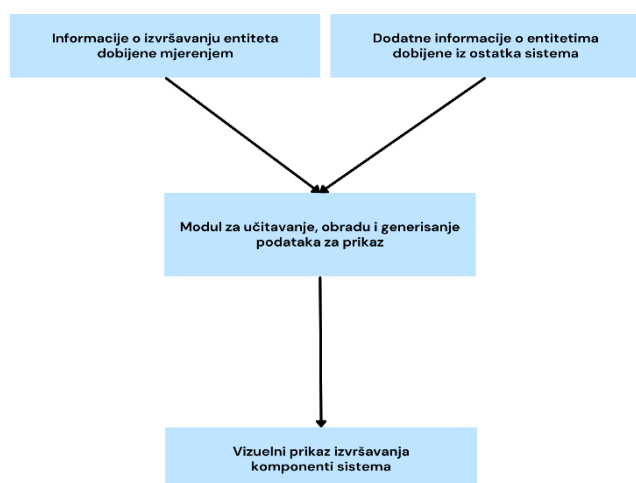
1. učitavanje mjernih podataka i prikaz izvršavanja izvršnih jedinica unutar jednog ciklusa,
2. pronalaženje i obilježavanje ciklusa čije izvršavanje potencijalno narušava stabilnost sistema i
3. pronalaženje i obilježavanje izvršne jedinice unutar nestabilnog ciklusa, čije izvršavanje narušava performanse sistema:
 - a) izvršna jedinica nije pokrenuta – nedostaje u ciklusu,
 - b) izvršna jedinica je pokrenuta i traje duže od vremena najgoreg slučaja njenog izvršavanja i
 - c) izvršavanje jedne izvršne jedinice prekinuto od strane druge, što dovodi do njenog produženog trajanja.

Sam alat se sastoji iz dva dijela. Prvi dio rješenja čini modul za učitavanje i obradu mjernih rezultata, te generisanje strukture podataka potrebne za pravilan prikaz. Pod učitavanjem i obradom ulaznih podataka, odnosno rezultata mjerenja performansi sistema, se ne podrazumijeva samo učitavanje informacija iz baze podataka i njihovo filtriranje na osnovu korisničkih zahtjeva, već i dobavljanje dodatnih informacija o entitetima sistema iz različitih fajlova određenih modelom sistema, kako bi vizualizacija imala sve neophodne podatke.

Drugi dio alata se odnosi na samu vizualizaciju podataka. Ovaj dio rješenja obezbjeđuje vizuelni prikaz generisanih podataka, na način koji omogućava njihovo lako

razumijevanje i analizu. Alat donosi mehanizme za lako manipulisanje vizuelno prikazanim podacima, čime je ostvarena i podrška za interakciju korisnika sa samim grafičkim prikazom. Dodatno, drugi dio koncepta rješenja nudi mogućnost proširenog prikaza entiteta sistema koji su korisniku od interesa, čime je znatno olakšan manuelni rad inženjera izdvajanjem i naglašavanjem problematičnih slučajeva izvršavanja softvera.

Na slici 3 prikazan je tok upotrebe alata za vizualizaciju *AUTOSAR logging/tracing* protokola, implementiranog u ovom radu. Kao prvi korak u procesu generisanja prikaza izvršavanja softvera u *AUTOSAR* sistemu, izdvaja se prikupljanje ulaznih podataka na kojima se bazira sam prikaz. Kako se može zaključiti sa slike 3, dio podataka dobija se direktno iz mjerenja izvršavanja softvera, pomoću eksternih alata, dok se drugi dio neophodnih informacija dobavlja iz različitih konfiguracionih fajlova unutar samog sistema. Prikupljeni podaci predstavljaju ulaz u modul za analizu i obradu informacija, na osnovu čega se generiše struktura podataka pogodna za samu vizualizaciju. Data struktura podataka sadrži sve relevantne podatke koji se proslijeđuju dijelu rješenja za grafički prikaz, što ujedno predstavlja i izlaz alata za vizualizaciju *AUTOSAR logging/tracing* protokola.



Slika 3. Koncept rješenja implementacije alata

Na osnovu datog opisa koncepta rješenja, urađena je implementacija programskog rješenja, o čemu će više detalja biti u nastavku rada.

3. REALIZACIJA

Za prikupljanje i obradu podataka neophodnih za kreiranje vizualizacije *AUTOSAR logging/tracing* protokola, korišćen je programski jezik *Python*, koji svojom jednostavnošću upotrebe i bogatstvom biblioteka različitih namjena, nudi veoma lak i efikasan način za implementaciju različitih softverskih rješenja. Za razliku od toga, sam grafički prikaz izvršavanja izvršnih jedinica u automobilskom softveru implementiran je *html + css* tehnologijom, uz dodatak *JavaScript* koncepta.

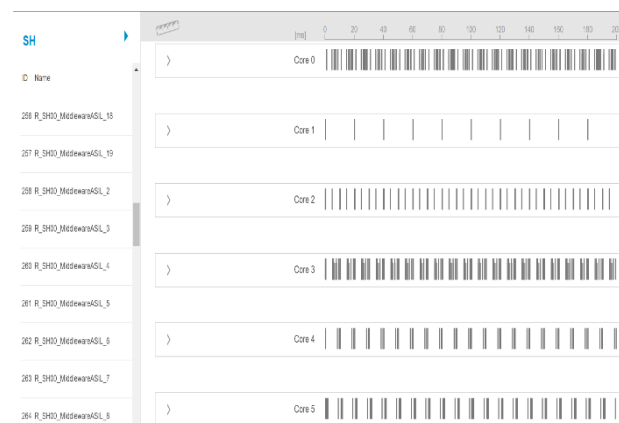
- **Učitavanje podataka dobijenih mjerenjem**

Osnovni preduslov, da bi uopšte izvršavanje softvera u sistemu moglo da se vizualizuje pomoću alata opisanog ovim radom, jeste prikupljanje ulaznih podataka, odnosno mjerenja. Za potrebe rada, korišćen je eksterni alat, čiji

izlaz čini više dokumenata u *.csv* formatu. Najznačajniji od njih, koji predstavljaju ujedno i ulazne podatke za alat za vizualizaciju, jesu *trace_events.csv* i *host_runnable_summary.csv*, u kojima su sadržane sve informacije o realnom izvršavanju izvršnih jedinica sistema, poput vremena u kom se desio neki događaj u sistemu, identifikatora datog događaja ili njegovog tipa. Za obradu datih dokumenata koristi se modul, nazvan *csv_file_handler.py*, u kom je implementirana kompletna logika za učitavanje i obradu informacija o realnim događajima u sistemu. Završni rezultat obrade datih podataka predstavlja kolekcija objekata klase *Slot* koji predstavljaju izvršne jedinice u sistemu. Navedena kolekcija se dalje proslijeđuje kao ulaz glavnom modulu alata, u kom je implementirano glavno rješenje za pripremu podataka da bi vizuelni prikaz bio generisan sa kompletnim skupom detalja o izvršavanju svakog entiteta.

- **Modul za obradu učitanih podataka i generisanje strukture za vizuelni prikaz**

Osnovni algoritam za analizu učitanih ulaznih podataka, te otkrivanje potencijalno nestabilnih ciklusa izvršavanja softvera, implementiran je u okviru *hyperperiod_calculation.py* modula. Osim navedenog, dati modul posjeduje logiku za dobavljanje dodatnih informacija o izvršnim jedinicama, koje nisu dio ulaznih podataka, a služe za generisanje potpunog prikaza. Sam algoritam za obradu rezultata mjerenja performansi, funkcioniše na način da se za svaki ciklus izvršavanja softvera provjeravaju njegova potencijalna odstupanja od idealnog rasporeda izvršavanja, definisanog arhitekturom sistema. Ukoliko se detektuje bilo kakvo odstupanje, dati ciklus i potencijalno problematična izvršna jedinica unutar njega se označavaju, na način da bivaju lako uočljive u vizuelnom prikazu.



Slika 4. Vizualizacija jednog ciklusa izvršavanja

Nakon završetka obrade ulaznog skupa podataka, proširenog dodatnim informacijama o svakoj izvršnoj jedinici, formira se baza podataka u formi *.json* dokumenta, odakle će informacije biti čitane od strane dijela alata za generisanje vizuelnog prikaza. Baza podataka, zapravo, sadrži listu ciklusa izvršavanja, gdje svaki ciklus sadrži realne podatke o svakoj izvršnoj jedinici, o trenutku početka i kraja njenog izvršavanja, te da li je svojim izvršavanjem narušila globalni vremenski raspored. Nakon upisa podataka u bazu podataka, automatski se pokreće procedura njihovog učitavanja i

generisanja prikaza, pri čemu automatski dolazi do otvaranja internet pretraživača i pojavljivanja generisanog prikaza u njemu, što se može vidjeti na slici 4.

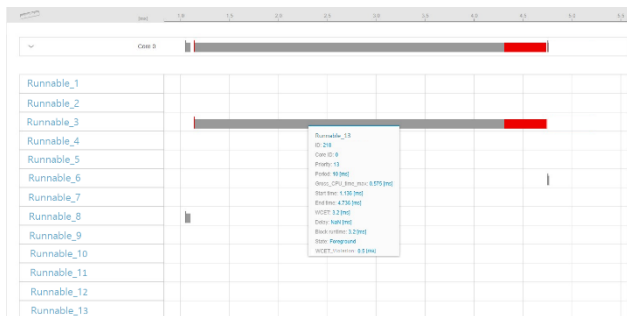
Uspješnim generisanjem grafičkog prikaza događaja, postignut je glavni cilj rada. Sve akcije koje je dalje potrebno preduzeti jesu na inženjeru koji vrši analizu ponašanja sistema, a donošenje odluka je značajno uprošćeno i olakšano uz posjedovanje ovakvog alata.

4. TESTIRANJE I REZULTATI

U sklopu testiranja programskog rješenja, u obzir su uzeti slučajevi, odnosno zahtjevi navedeni u poglavlju Koncept rješenja. Svaki od testova urađen je nad različitim ulaznim podacima, kao i različitim podešavanjima sistema, kako bi što veći broj kritičnih slučajeva primjene alata bio pokriven.

Prvi i osnovni slučaj koji je testiran jeste samo učitavanje mjernih podataka iz stabilnog sistem i generisanje prikaza. Opisana situacija je prikazana na slici 4, gdje se može vidjeti da je alat uspješno izgenerisao prikaz jednog ciklusa izvršavanja izvršnih jedinica, čime je uspješno prošao dati test.

Drugi slučaj koji je testiran jeste slučaj koji se često sreće u praksi, pogotovo u procesu razvoja automobilske softvera, a to je da je izvršavanje određene izvršne jedinice produženo, čime je narušeno ograničenje njenog maksimalnog dozvoljenog vremena izvršavanja u najgorem slučaju. Alat je uspješno prepoznao dati problematični ciklus i izvršnu jedinicu koja narušava stabilnost sistema, što se može vidjeti na slici 5. Dodatno je testiran i prikaz detalja o samom kritičnom događaju, a sa slike 5 se može vidjeti da klikom na datu izvršnu jedinicu korisnik dobija detaljan pregled podataka, čime je alat prošao i ovaj test.



Slika 5. Narušavanje Worst Case Execution Time ograničenja

Treći bitan test kom je podvrgnut alat za generisanje vizualnog prikaza AUTOSAR logging/tracing protokola je sličan prethodnom slučaju, s tim da izvršna jedinica produžava svoje izvršavanje zbog prekida od strane druge jedinice većeg prioriteta, što dovodi do nestabilnosti softvera. Alat je uspješno detektovao i obilježio i ovaj slučaj, čime je potvrđeno da su uspješno implementirani svi zahtjevi navedeni u poglavlju Koncept rješenja.

5. ZAKLJUČAK

Istraživanja sprovedena u toku izrade rada pokazala su da kvalitetno dizajnirani alati, poput rješenja obrađenog u

ovom radu, igraju ključnu ulogu u unapređenju kvaliteta proizvodnje, na način da olakšavaju analize kompleksnih podataka koji proizilaze iz sistema zasnovanih na AUTOSAR standardu. Dokazano je da adekvatna vizualizacija podataka omogućava dublje razumijevanje performansi i ponašanja softverskih sistema u automobilu. Osim toga, važno je naglasiti i značaj alata pri detektovanju i otklanjanju grešaka sistema, što direktno doprinosi kvalitetu i pouzdanosti automobila.

Analizom dobijenih rezultata dokazano je da alat za vizualizaciju AUTOSAR logging/tracing protokola predstavlja pouzdano rješenje, koje se uz minimalne izmjene u pogledu konfiguracije, može primjeniti za analizu i poboljšanje performansi različitih komponenti automobilske softvera.

6. LITERATURA

- [1] K. Senthilkumar and R. Ramadoss, „Designing multicore ECU architecture in vehicle networks using AUTOSAR,“ *Third International Conference on Advanced Computing*, Chennai, India, 2011, pp. 270-275).
- [2] Dragan Samardžija, Milena Milošević, *Duboko učenje za autonomna vozila*, 2021, str. 1-2.
- [3] https://ti.tuwien.ac.at/cps/teaching/courses/deterministic-networking/lecture-slides/detnet04_TTEthernet.pdf [pristupljeno februar 2024.].
- [4] Obermaisser R., *Time-Triggered Communication, Networked Embedded Systems*, CRC Press, 2017.
- [5] Navet N., Simonot-Lion F., *Automotive Embedded Systems Handbook*, CRC Press, 2021.
- [6] <https://www.autosar.org/standards/classic-platform> [pristupljeno februar 2024.].
- [7] Qiang Wang, Baiyu Xin, Chao Li and Hong Chen, „The realization of reusability in vehicular software development under AUTOSAR,“ *IEEE Conference Anthology*, China, 2013, pp. 1-6.
- [8] https://www.autosar.org/fileadmin/standards/R22-11/CP/AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf [pristupljeno februar 2024.].
- [9] S. Raju, G. Jeyakumar, A. Mukherji and J. K. Thanki, „Time synchronized diagnostic event data recording based on AUTOSAR,“ *IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, Bhubaneswar, India, 2017, pp 1-6.
- [10] M. Rotim, „Adaptive AUTOSAR Diagnostic Log and Trace processing platform,“ *45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, Opatija, Croatia, 2022, pp. 101-105.

Kratka biografija:



Mladen Planojević rođen je na Sokocu 1998. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Automobilske softver odbranio je 2024.god.
kontakt: mladenplanojevic@gmail.com

MATEMATIČKO MODELOVANJE I PID REGULACIJA TERMALNE ZONE U HIL OKRUŽENJU**MATHEMATICAL MODELING AND PID REGULATION OF THE THERMAL ZONE IN THE HIL ENVIRONMENT**

Predrag Nikolić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu opisan je matematički model termalne zone, kao i primena Proporcionalno – Integralno – Diferencijalnog zakona PID (eng. Proportional - integral - derivative) regulacije temperature u okviru zone. Praktični deo rada obuhvata implementaciju hardvera u petlji - HIL (eng. Hardware In the Loop) pristupa upotrebom odgovarajućeg hardvera i programskog okruženja. Razvijen je dinamički model termalne zone u kojoj je integrisano grejno telo (radijator) sa mogućnošću upravljanja putem kontrole masenog protoka dotoka tople vode do istog.

Ključne reči: Modelovanje, HIL, PID, termalna zona, regulacija

Abstract – The paper details mathematical modeling and PID (Proportional - integral - derivative) temperature regulation applied to a thermal zone, implemented experimentally using specific hardware and software. A dynamic model of the thermal zone has been developed in which a heating body (radiator) is integrated with the possibility of management by controlling the mass flow of hot water inflow to it.

Keywords: Modeling, PID, HIL, thermal zone, regulation

1. UVOD

Simulacija sa hardverom u petlji predstavlja moćnu tehniku koja se koristi u inženjerskim delatnostima gde je potrebno testirati i oceniti rad složenih dinamičkih sistema. HIL predstavlja vezu između virtualne simulacije i realnih procesa sistema.

Neki od razloga za upotrebu HIL sistema:

- Olakšana manipulacija sa elementima zone u poređenju sa stvarnim procesom.
- Povećana vremenska efikasnost omogućava kontinuirano ispitivanje i postavljanje sistema u željene uslove.
- Smanjeni troškovi putovanja i rada sistema, s obzirom da inženjer ne mora fizički biti prisutan na mestu odvijanja procesa.

NAPOMENA:

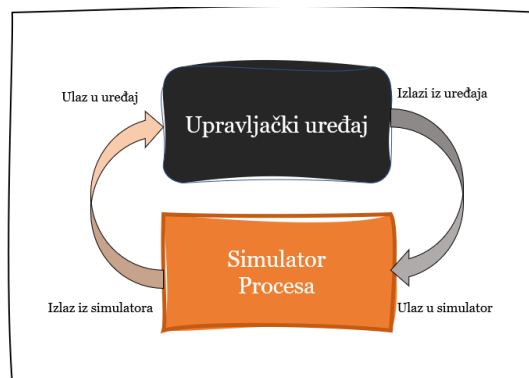
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Velimir Čongradac, red. prof.

Model procesa zahteva detaljan opis skupa neophodnih diferencijalnih jednačina dinamike sistema koji opisuju energetske tokove, masene tokove kao i promene parametara.

U daljem tekstu rada, detaljno će se izvesti matematičko modelovanje termalne zone, biće sprovedena eksperimentalna ispitivanja sa odgovarajućim softverskim i hardverskim uređajima, a zatim biti analizirani dobijeni rezultati. Ova analiza će dodatno doprineti razumevanju rezultata modela i postaviti smernice za usavršavanje kako modela, tako i kompletnog HIL sistema u simulaciji termalnih procesa.

2. OSNOVE HIL SISTEMA

Hardware in the Loop predstavlja konfiguraciju u kojoj je upravljački uređaj povezan u petlji sa delom ili kompletnim hardverom na kojem se odvija simulacija procesa. Ovaj pristup omogućava inženjerima testiranje upravljačkih algoritama u kontrolisanim uslovima, ubrzavajući tako razvoj i povećavajući efikasnost istraživanja i razvoja simulacija sistema upravljanja.



Slika 1. Sistem hardvera u povratnoj petlji.

3. ZONA

Toplotna zona predstavlja prostoriju ili skup prostorija unutar nekog objekta sa sličnim zahtevima u pogledu održavanja sledećih parametara:

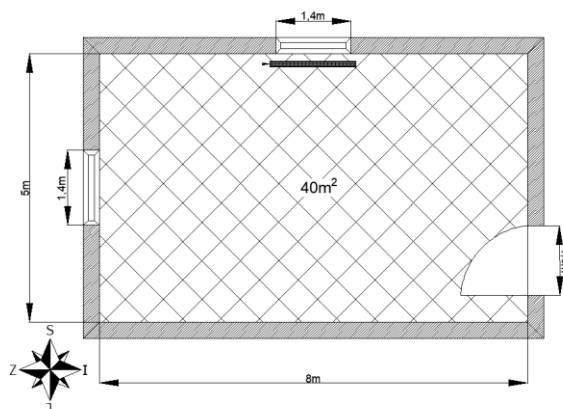
1. Temperatura vazduha,
2. Relativna vlažnost vazduha,
3. Koncentracija CO₂,
4. Osvetljenje.

Prostorije koje čine jednu toplotnu zonu mogu, a i ne moraju biti fizički odvojene što znači da u okviru jednog

objekta može postojati veći broj toplotnih zona svaka sa nezavisnim izvorom grejanja i hlađenja [1]. Zbog različite namene, broja ljudi i sunčevog zračenja, svaka pojedinačna zona zahteva određene parametre kako bi uslovi bili odgovarajući, pa se zoniranje, tj. upravljanje parametrima pojedinačne zone čini adekvatnim rešenjem. Treba napomenuti da se zoniranjem postižu znatne uštede u energiji, spram na primer grejanja kompletne zgrade, jer se stavlja akcenat na zone koje se koriste.

4. REFERENTNA TOPLOTNA ZONA

U radu će se koristiti toplotna zona koja predstavlja kuću sa jednom prostorijom, površine 40m^2 , ne uzimajući u obzir debljinu zidova. Kuća je pozicionirana na teritoriji opštine Novi Sad, geografske širine $45^\circ 14' 9.27''$ i geografske dužine $19^\circ 45' 40.17''$. Podrazumeva se da su svi zidovi kuće spoljašnji i orijentisani kao što je prikazano na slici 2. Visina prostorije, od poda do plafona iznosi 3.14 m, što predstavlja proizvoljnu izabranu vrednost. Za projektovanu temperaturu u zoni je izabrana vrednost od 20°C .



Slika 2. Referentna toplotna zona.

5. MATEMATIČKI MODEL TOPLOTNE ZONE

Model će biti zasnovan početnoj bilansnoj jednačini toplotnih dobitaka i gubitaka u okviru zone. Definiše se kao:

$$\dot{Q}_{omotač} + \dot{Q}_{inf} = \dot{Q}_{ljudi} + \dot{Q}_{rad} \quad (1)$$

gde $\dot{Q}_{omotač}$ predstavlja gubitke toplotnih omotača zone, $\dot{Q}_{inf}$ toplotne gubitke usled infiltracije kroz procepe vrata i prozora, $\dot{Q}_{ljudi}$ toplotni dobitak od ljudi i $\dot{Q}_{rad}$ što predstavlja toplotni dobitak od strane grejnog tela.

5.1 Model toplotnog omotača zone

Proračun toplotnih gubitaka zone je izveden po EN 12831 [2] standardu. Deli se na transmisione i ventilacione gubitke i definiše se kao:

$$\Phi_u = \Phi_T + \Phi_V \quad (2)$$

gde Φ_T predstavlja transmisione gubitke i Φ_V predstavlja ventilacione gubitke. Transmisioni gubici se ogledaju u

gubicima prilikom prenosa toplote iz zone ka spoljašnjoj sredini. Granične površine prenosa su zid, plafon, pod, prozor i vrata. Potrebno je izračunati transmisione gubitke kroz sve te površine, te se definiše kao:

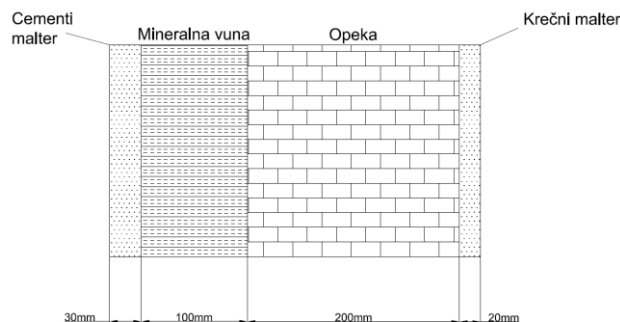
$$\Phi_T = (H_{T,z} + H_{T,p} + H_{T,v} + H_{T,p} + H_{T,p}) \cdot \Delta t \quad (3)$$

gde je $H_{T,i}$ pojedinačan koeficijent transmisisonog gubitka od grejanog prostora prema spoljašnjoj okolini dok Δt predstavlja razliku temperatura između unutrašnje projektovane i spoljašnje sredine.

Gde se pojedinačni koeficijent transimisonog gubitka od grejanog prostora prema spoljašnjoj okolini $[\text{W/K}]$ računaju:

$$H_{T,i} = A_i \cdot U_i \cdot e_i \quad (4)$$

gde je A_i površina elementa građevine (zid, prozor, vrata, plafon, pod), U_i koeficijent prelaza toplote i - te građevine i e_i korekcionni faktor izloženosti, usvaja se 1. Konstrukcija zida je prikazana na slici 3.



Slika 3. Konstrukcija zida

Ventilacioni gubici se definišu kao:

$$\Phi_{V,i} = H_{V,i} \cdot (t_u - t_s) \quad (6)$$

gde je $H_{V,i}$ koeficijent ventilacionih toplotnih gubitaka, t_u unutrašnja projektana temperatura, spoljašnja temperatura t_s .

5.2 Model unutrašnji toplotnih dobitaka

Unutrašnji toplotni dobici potiču od:

1. Ljudi
2. Osvetljenja
3. Uređaja

U ovom radu će se zanemariti uticaj osvetljenja i uređaja dok će se toplotni dobitak ljudi i dalje razmatrati. Da bi se izvršila sinteza modela unutrašnjih dobitaka od strane ljudi, neophodno je poznavati vrednosti toplotne energije koju ljudi oslobode prilikom boravka u prostoriji.

Treba napomenuti da ta vrednost varira od tipa aktivnosti kojom se ljudi bave u prostoriji. Podaci su preuzeti iz ASHRE priručnika [3] i prikazani u tabeli 1.

Aktivnost	Osetna W	Latentna W	Ukupna W
Sedenje (odmor)	70	30	100
Sedenje (laki rad)	75	45	120
Stojanje	75	70	145
Laki fizički rad	90	160	250
Teški fizički rad	185	285	470

Tabela 1. Procenjena emitovana toplotna energija od ljudi.

Na osnovu procenjene emisije toplotne energije od strane ljudi je definisana je sledeća diferencijalna jednačina:

$$\frac{dQ_{ljudi}}{dt} = n_s \cdot 70 + n_{st} \cdot 75 + n_l \cdot 90 + n_t \cdot 185. \quad (12)$$

gde je n_s broj ljudi koji sedi, n_{st} broj ljudi koji stoji, n_l broj koji obavlja laki rad i n_t broj koji obavlja teški rad.

5.3 Model grejnog tela

Bilansna jednačina grejnog dela je definisana kao:

$$\dot{Q}_{dov} = \dot{Q}_{aku} + \dot{Q}_{rad}. \quad (13)$$

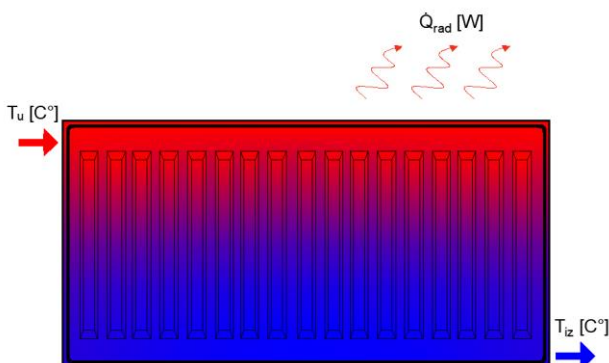
gde $\dot{Q}_{dov}$ prestavlja toplotnu energiju dopremljenu grejnom telu, $\dot{Q}_{aku}$ akumulisanu toplotnu energiju i $\dot{Q}_{rad}$ toplotnu energiju emitovanu u okolinu [4].

Detaljni model je definisan kao:

$$\dot{m}_w \cdot c_w \cdot (T_u - T_{iz}) = K_1 \cdot \frac{dT}{dt} + \dot{Q}_n \cdot \left[\frac{\Delta t}{\Delta t_n} \right]^n. \quad (14)$$

$$K_1 = (m_v \cdot c_w + m_m \cdot c_m). \quad (15)$$

gde je $\dot{m}_w$ maseni protok tople vode ka radijaturu, c_w toplotni kapacitet vode, m_v masa vode u radijatoru, m_m masa metala radijatora, c_m toplotni kapacitet metala radijatora, $\dot{Q}_n$ nominalna snaga radijatora za dati tip radijatora merena po standardu EN 442, Δt logaritamska razlika u temperaturi, Δt_n logaritamska razlika u temperaturi koja se računa za odabrani temperaturski režim (90/70/20 °C), dok je n karakteristični eksponent za dati tip radijatora.



Slika 4. Skica panelnog radijatora.

5.4 Model emisije CO₂

Količina ugljen - dioksida u prostoriji je u direktnoj zavisnosti od ljudi koji se nalaze u prostoriji i vrste aktivnosti kojom se bave. Osvežavanje prostorije se jedino vrši prilikom infiltracije vazduha kroz procepe vrata i prozora što će se uzeti u obzir prilikom modelovanja [5]. Treba napomenuti da je ovaj model izveden iz statičkog modela i da je usvojen sa dosta pretpostavki.

Model emisije je definisan kao:

$$\frac{dC(t)}{dt} = \frac{K_e}{3600} - \frac{V_{inf} \cdot 400}{3600} \quad (16)$$

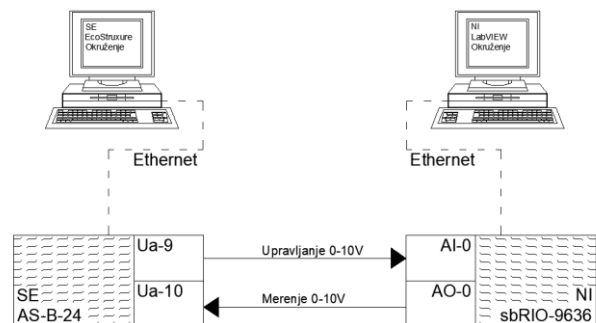
$$K_e = n_s \cdot 5 + n_{st} \cdot 6,06 + n_{lr} \cdot 7 + n_t \cdot 9,5 \quad (17)$$

gde je n_s broj ljudi koji sedi, n_{st} broj ljudi koji stoji, n_{lr} broj ljudi koji obavlja laki fizički rad, n_{tr} broj ljudi koji obavlja teški fizički rad i V_{inf} protok svežeg vazduha usled infiltracije kroz procepe vrata i prozora.

5 EKSPERIMENTALNA KONFIGURACIJA

Prilikom formiranja HIL sistema, potrebno je posedovati određeni hardverski sklop na kojem će se izvršavati upravljanje kao i simulacija u realnom vremenu. Za potrebe upravljanja u ovom radu će se koristiti kontroler Schneider Electric AS-B-24 na kojem će se izvršavati zakon upravljanja PID, a simulacija termalne zone će se odvijati na kontroleru National Instruments sbRIO-9636. National Instruments sbRIO 9636 kontroler je opremljen Real-Time procesorom i FPGA-om na istom čipu što je ključno za funkcionisanje ovog kontrolera u realnom vremenu.

Naime, na analognom izlazu iz NI kontrolera se šalje informacija o trenutnoj temperaturi zone, takav signal se doprema do SE kontrolera koji na osnovu referentne vrednosti (20°C) računa grešku uz pomoć PID zakona upravljanja, a pritom na svom analognom izlazu generiše upravljanje. Signal upravljanja se obrađuje u NI kontroleru i pomoću njega reguliše otvorenost ventila grejnog tela radijatora. Treba napomenuti da su oba kontrolera povezana Ethernet vezom sa PC računarima na kojima se vrši prikaz trenutnih vrednosti parametara simulacije zone i upravljanja.

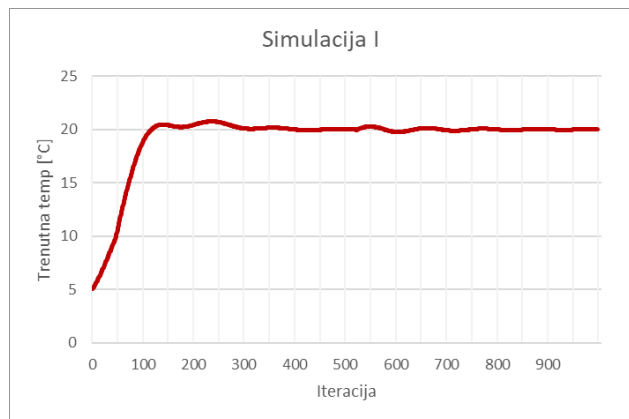


Slika 5. Šematski prikaz HIL konfiguracije.

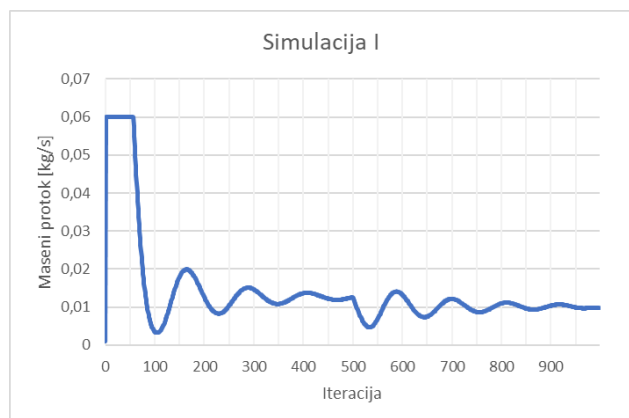
6 REZULTATI EKSPERIMENTALNE POSTAVKE

Prilikom simulacije termalne zone, izvršeno je par simulacija radi prikaza funkcionisanja HIL sistema kao i provere normalnog rada simulacionog modela i upravljanja.

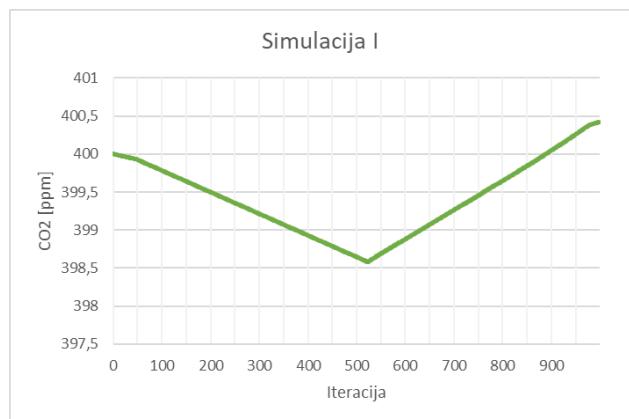
Na slici 6, 7 i 8 su prikazani rezultati simulacije I, na kojima se mogu uočiti promene temperature u prostoriji, promene masenog protoka tople vode na dovodu radijatora, kao i promena koncentracije CO₂.



Slika 6. Rezultati simulacije I, trenutna temperatura.



Slika 7. Rezultati simulacije I, maseni protok.



Slika 8. Rezultati simulacije I, koncentracija CO₂.

Simulacija termalne zone je u skladu sa očekivanjima, prateći zadatke dostizanja ciljanih vrednosti. Ipak, primećen je poremećaj u simulaciji kada dve osobe ulaze u zonu i obavljaju aktivnosti sedenja i lakog rada, što se može uočiti oko petstote iteracije. Grafikoni masenog

protoka tople vode jasno pokazuju nagli pad u tim trenucima, s PID zakonom upravljanja koje uspešno kompenzuje dodatne poremećaje i uspeva da održi temperaturu u željenom opsegu.

7 ZAKLJUČAK

Na osnovu sprovedenog istraživanja, čiji je cilj bio proučavanje i simulacija termalne zone u HIL sistemu, mogu se izvesti sledeći zaključci.

Razvijen je dinamički model termalne zone u kojoj je integrisano grejno telo (radijator) sa mogućnošću upravljanja putem kontrole masenog protoka dotoka tople vode do istog. Kroz simulaciju pokazano je da model termalne zone, koji je povezan sa PID regulatorom, postiže željene vrednosti projektovane temperature zone i oponaša poremećaje u vidu ulaska ljudi u prostoriju termalne zone. Iako se HIL sistemi često koriste u automobilske, avio, vojnoj i energetske industriji, pokazuje se da se dobri rezultati postižu i u procesima grejanja i hlađenja stambeno-poslovnih objekata [6]. Na osnovu zaključaka i rezultata iznetih u ovom radu, potrebno je dalje istraživanje kako bi se uspostavio precizniji i detaljniji model termalne zone.

8 LITERATURA

- [1] <http://ymgigroup.com/> Pristupljeno 10.02.2024.
- [2] Evropski komitet za standardizaciju. (2017). SRPS EN 12831-1:2017: Energetska efikasnost zgrada - Metoda za proračun projektovanog toplotnog opterećenja - Deo 1: Opterećenje i proračun grejanja.
- [3] Kreider, J. F., Curtiss, P. S., & Rabl, A. (2010). Heating and Cooling of Buildings: Design for Efficiency. (pp. 300).
- [4] Chen, L., & Sørensen, B. R. (2020). Modelling Multi-layer Hydronic Radiators. E3S Web of Conferences, 172, 12007.
- [5] Sakamoto, M., Li, M., Kuga, K., Ito, K., Bekö, G., Williams, J., & Wargocki, P. (2022). CO2 emission rates from sedentary subjects under controlled laboratory conditions. Building and Environment, 211, 108735.
- [6] Mihalić, F., Truntiće, M., & Hren, A. (2022). Hardware-in-the-Loop Simulations: A Historical Overview of Engineering Challenges. Electronics, 11, 2462.

Kratka biografija:



Predrag Nikolić rođen je u Boru 1998. god. Osnovne akademske studije završio je 2022. god., smer Čiste energetske tehnologije. Master rad na Fakultetu tehničkih nauka iz oblasti Automatike i računarstva odbranio je 2024.god. kontakt: predragnikolic99@gmail.com

SOFTVERSKI SISTEM ZA SEMANTIČKU PRETRAGU TEKSTA PRIMENOM VEKTORSKIH BAZA PODATAKA**SOFTWARE SYSTEM FOR SEMANTIC TEXT SEARCH USING VECTOR DATABASE TECHNOLOGY**

Srđan Šuković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – PRIMENJENE RAČUNARSKE NAUKE I INFORMATIKA

Kratak sadržaj – *Ovaj rad predlaže alternativni pristup pretrage teksta upotrebom vektorskih baza podataka i pretreniranih modela veštačke inteligencije. Rad obrađuje proces kreiranja vektorske reprezentacije tekstualnih podataka, indeksiranje istih u različite vektorske baze i postupak pretrage sličnosti sa upitima korisnika.*

Ključne reči: Vektorska baza, vektorska reprezentacija, semantička pretraga.

Abstract – *This paper proposes an alternative approach to text search using vector databases and pretrained artificial intelligence models. The paper deals with the process of creating a vector representation of textual data, indexing them into different vector databases and the procedure of searching for similarities with user queries.*

Keywords: Vector database, vector embedding, semantic search.

1. UVOD

Semantička pretraga teksta predstavlja metod pretrage koji se ne oslanja isključivo na tačne vrednosti ključnih reči, nego na razumevanje celokupnog konteksta i namere osobe koja pretražuje. Ova ideja se realizuje uz pomoć pretreniranih modela veštačke inteligencije koji imaju sposobnost razumevanja teksta u prirodnom jeziku, tako što na osnovu teksta kreiraju visoko-dimenzionalne vektorske reprezentacije koje uspešno u sebi nose semantičko značenje ulaznog teksta [1]. Vektorske baze, kao što su *Pinecone*, *Weaviate* i *Qdrant*, efikasno rešavaju problem čitanja, pisanja i upravljanja vektorima.

Ovaj rad objedinjuje mogućnosti koje nude modeli veštačke inteligencije i vektorske baze podataka - uz pomoć modela generišu se vektorske reprezentacije tekstualnog sadržaja, koje se dalje upisuju u vektorske baze. Odatle, na osnovu korisničkog upita, koji se takođe prevodi u vektorsku reprezentaciju, radi se pretraga sličnosti u odnosu na prethodno upisane vektore i korisnici dobijaju najbližije rezultate njihovim upitima.

Ovaj algoritam nije ograničen ni na jedan domen problema, i može poboljšati bilo koju funkcionalnost

pretrage u već postojećim aplikacijama. Konkretno, u ovom radu, algoritam je implementiran na primeru pretrage filmova.

1.1 Skup podataka

Za implementaciju rešenja korišćen je skup podataka [2] koji sadrži preko 34 hiljade različitih prepričanih filmova, sa svojim opisnim podacima (naslov, godina, žanr, poreklo, režiseri, glumci, fabula, link do *Vikipedije*). S obzirom da skup podataka nije skroz prilagođen problemu koji ovaj rad rešava, neophodno je da indeksiranju prethodi pretprocesiranje podataka. Pretprocesiranje uključuje otklanjanje nepotrebnih informacija, nepotpunih sadržaja, duplikata i slično.

1.2 Pinecone

Pinecone [3] je specijalizovana baza podataka za upravljanje vektorima, dizajnirana da olakša rad sa velikim skupovima podataka i složenim zadacima pretraživanja u domenu veštačke inteligencije i mašinskog učenja. Kao usluga koja se temelji na konceptu vektorskog pretraživanja, *Pinecone* omogućava korisnicima da efikasno organizuju, indeksiraju i pretražuju podatke visoke dimenzionalnosti, što je posebno relevantno za aplikacije poput preporuka sadržaja, semantičkog pretraživanja, detekcije anomalija i personalizovanih iskustava korisnika.

Jedna od ključnih prednosti *Pinecone*-a je njegova sposobnost da precizno i brzo vrši pretraživanje sličnosti u velikim skupovima podataka. To se postiže korišćenjem sofisticiranih algoritama za vektorsko pretraživanje, koji omogućavaju brzo pronalaženje najrelevantnijih rezultata na osnovu sličnosti ugrađenih vektora. Ova sposobnost je posebno važna u primenama gde je potrebno obraditi velike količine tekstualnih, audio ili vizuelnih podataka i pronaći visoko relevantne informacije ili obrasce.

Platforma *Pinecone* omogućava skalabilnost i visoku dostupnost, što je čini pogodnom za upotrebu u produkciji. Njeni alati i API-ji su dizajnirani da budu intuitivni i laki za korišćenje, što omogućava brzu integraciju u postojeće sisteme.

Takođe, *Pinecone* podržava integraciju sa popularnim okvirima za mašinsko učenje i obradu podataka, kao što su *TensorFlow* i *PyTorch*, olakšavajući razvoj i implementaciju složenih modela veštačke inteligencije.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dragan Ivanović, red. prof.

1.3 Weaviate

Weaviate [4] je napredna, platforma otvorenog koda za upravljanje bazom podataka koja se fokusira na kombinovanje modela grafova sa vektorskim pretraživanjem, pružajući tako jedinstvenu infrastrukturu za obradu i analizu podataka. Osnovana s idejom da unapredi pristup obradi složenih podataka, *Weaviate* se posebno ističe u domenima kao što su *NLP* (engl. *Natural Language Processing*), semantičko pretraživanje i *AI* (engl. *Artificial Intelligence*) aplikacije.

Jedinstvenost *Weaviate*-a leži u njegovoj hibridnoj strukturi koja spaja tradicionalno grafovsko skladištenje podataka sa naprednim metodama vektorskog pretraživanja. Ova kombinacija omogućava korisnicima da izvode kompleksne upite, analiziraju velike količine podataka i pronalaze relevantne informacije kroz sličnosti u sadržaju. *Weaviate* automatski transformiše podatke u vektorske reprezentacije, što olakšava precizno i kontekstualno pretraživanje.

Weaviate je dizajniran da bude visoko skalabilan i prilagodljiv, omogućavajući efikasnu upotrebu u različitim okruženjima i aplikacijama. Podržava raznolike tipove podataka, uključujući tekst, slike, i složene strukture podataka, što ga čini pogodnim za širok spektar upotreba. Integracija sa popularnim alatima za mašinsko učenje poput *TensorFlow* i *PyTorch* dalje proširuje njegovu primenljivost, čineći ga pogodnim za razvoj *AI* modela.

1.4 Qdrant

Qdrant [5] je moderna, otvorena platforma za upravljanje podacima, specijalizovana za efikasno vektorsko pretraživanje i analizu podataka. Razvijena kako bi zadovoljila rastuće potrebe za obradom složenih i velikih skupova podataka, posebno u domenima veštačke inteligencije i mašinskog učenja, *Qdrant* se ističe svojom sposobnošću da brzo i precizno obradi i indeksira podatke visoke dimenzionalnosti.

Ključna karakteristika *Qdrant*-a je njegova napredna podrška za vektorsko pretraživanje, omogućavajući korisnicima da izvrše kompleksne upite i dođu do relevantnih informacija u velikim bazama podataka. Ova funkcionalnost je posebno korisna u aplikacijama poput pretrage sličnih slika, personalizacije sadržaja, semantičkog pretraživanja i sistema preporuka, gde je važno brzo pronalaženje najrelevantnijih rezultata.

Qdrant je dizajniran s fokusom na visoku performansu i skalabilnost, što ga čini idealnim za upotrebu u zahtevnim poslovnim okruženjima i aplikacijama koje zahtevaju obradu velikih količina podataka. Njegova arhitektura omogućava efikasno upravljanje resursima i optimizaciju za brzo pretraživanje, što značajno smanjuje vreme potrebno za obradu upita.

1.5 Kreiranje vektorske reprezentacije od tekstualnog sadržaja

Za kreiranje vektora korišćeni su *HuggingFace* pretrenirani modeli veštačke inteligencije [6], specijalizovani za kreiranje vektorskog sadržaja. Model koji je pokazao najbolje performanse je *all-mpnet-base-v2*. Navedeni modeli su otvorenog koda i besplatni su za korišćenje.

1.6 Upis podataka u baze

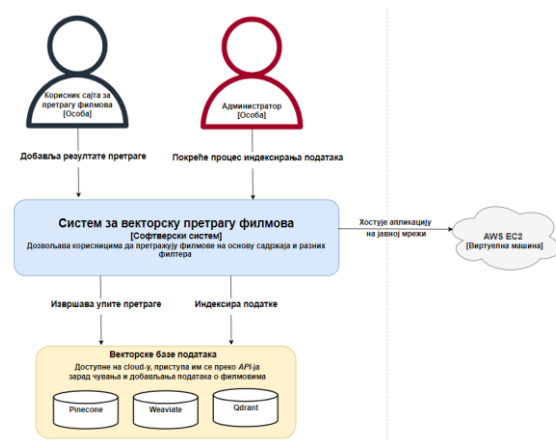
Za upis podataka u svaku od baza, potrebno je prilagoditi se API-ju koji one nude. Ono što je zajedničko za sve, jeste da se u jednom pozivu biblioteke vrši upis prethodno generisanog vektora i metapodataka koji se odnose na njega. U ovom slučaju to je vektor generisan od teksta prepričanog filma i samog naslova, dok su metapodaci godina, žanr, glumci, režiseri i poreklo. Baze koje su korišćene u radu su *Pinecone*, *Weaviate* i *Qdrant*.

1.7 Dobavljanje podataka

Proces dobavljanja podataka se sastoji iz 4 celine: vektorizacija upita, adaptacija korisničkog upita i filtera *SDK*-ju odgovarajuće baze, sam poziv klijenta baze i prilagođavanje odgovora korisničkom interfejsu aplikacije. Vektorizacija upita vrši se na isti način kao i prilikom indeksiranja, uz pomoć specijalizovanih modela veštačke inteligencije i zajednička je za sve baze.

2. SPECIFIKACIJA SISTEMA SEMANTIČKE PRETRAGE TEKSTA

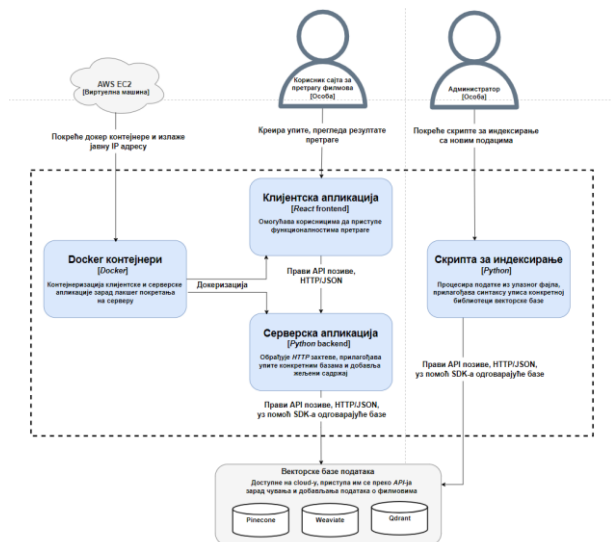
U ovom poglavlju prikazana je arhitektura sistema sa aspekta *C4* modela dijagramiranja, zasnovanom na apstrakcijama koje odražavaju kako arhitekta i programeri softvera razmišljaju i grade softver. Na slici 2.1 prikazan je kontekstni dijagram sistema, koji prikazuje kako sistem interaguje sa eksternim entitetima, kao što su korisnici, administratori, vektorske baze i klad provajderi (engl. *cloud providers*).



Slika 2.1 Kontekstni dijagram sistema za semantičku pretragu filmova

Detaljniji prikaz komponenta sistema prikazan je na dijagramu kontejnera (slika 2.2), koji prikazuje osnovne kontejnere sistema i njihove međusobne odnose. Ovakvo softversko rešenje sastoji se iz dve glavne celine:

1. Deo za indeksiranje podataka – implementiran kroz *Python* skripte koje pokreće administrator sistema. Oslanja se na *Python* biblioteku *sentence-transformers* koja služi kao apstrakcija korišćenja raznih modela za kreiranje vektora od teksta.
2. Deo za opsluživanje podataka – *Flask* aplikacija koja obrađuje zahteve i vrši upite ka vektorskim bazama.



Slika 2.2 Dijagram kontejnera sistema

Pored prethodno pomenutih celina sistema, implementiran je i korisnički interfejs uz pomoć *React* biblioteke, koji komunicira sa delom za opsluživanje podataka korisnicima.

3. EKSPERIMENT

Zarad verifikacije rezultata izdvojen je podskup celog skupa podataka, od kojeg je sačinjena mapa gde je ključ naslov filma, a vrednost tekst prepričanog filma. Zatim, upotrebom *eugenesiow/bart-paraphrase* modela sa *HuggingFace* platforme, tekst je preformulisan tako da bude donekle izmenjen, ali sadrži isto značenje kao i pre primene modela.

Ovaj pristup je odabran kako bi se dočarala fleksibilnost i moć semantičke pretrage teksta, sa upitom čiji je tekst izmenjen u odnosu na prethodno indeksirane podatke, ali je pak značenje zadržano.

Zatim, svaka od vrednosti iz mape prolazi kroz isti tok kao kada aplikacija primi zahtev za pretragom: kreiranje vektorske reprezentacije i izvršavanje upita nad bazom. Najzad, najbliži rezultat koji baza vrati, upoređuje se sa filmom koji je parafraziran iskorišćen za upit, i ukoliko dođe do podudaranja, pretraga se smatra uspešnom. Na osnovu broja pozitivnih i negativnih rezultata računa se metrika preciznosti.

4. REZULTATI I TUMAČENJA

Kvalitet odgovora na upit zavisi gotovo isključivo od modela koji je kreirao vektore i metrike sličnosti po kojoj se oni dobavljaju. Na osnovu analize performansi modela dokumentovane na *HuggingFace* platformi, za testiranje odabrani su sledeći modeli:

1. *all-mpnet-base-v2* – koji sveobuhvatno ima najbolje performanse što se kvaliteta vektora tiče
2. *all-MiniLM-L12-v2* – koji je tri puta manji i brži od prethodnog, međutim daje slabije performanse.

Kao metrika sličnosti, za oba modela odabrana je kosinusna sličnost, kao što je preporučeno u dokumentaciji. U tabeli 4.1 prikazani su broj pogodaka, broj promašaja i

preciznost dobijeni prilikom testiranja sistema sa vektorima dobijenim od navedenih modela, na podskupu podataka od 66 elemenata. Rezultati se odnose na to da li se očekivani dokument dobija kao najrelevantniji ili ne.

Model	Broj pogodaka	Broj promašaja	Preciznost
all-mpnet-base-v2	62	4	0,939
all-MiniLM-L12-v2	55	11	0,833

Tabela 4.1 Performanse sistema za očekivani najrelevantniji rezultat

U slučaju kada se proverava da li se očekivani rezultat nalazi u top 4 rezultata dobijenih pretragom, dobijaju se sledeći rezultati, prikazani u tabeli 4.2.

Model	Broj pogodaka	Broj promašaja	Preciznost
all-mpnet-base-v2	66	0	1.0
all-MiniLM-L12-v2	64	2	0,969

Tabela 4.2 Performanse modela za očekivani top 4 rezultat

5. ZAKLJUČAK

U radu je predstavljen sistem za vektorsku pretragu filmova na osnovu njihovog sadržaja i dostupnih filtera. Ovakvo softversko rešenje omogućuje korisnicima preciznu i brzu pretragu filmova, gde je dovoljno da osoba svojim rečima napiše šta želi da se u filmu dešava, i sistem će dobiti najrelevantnije rezultate naspram korisničkog upita, i time proces traženja filma svesti na minimum.

Rešenje je implementirano paralelnom upotrebom tri vektorske baze podataka - *Pinecone*, *Qdrant* i *Weaviate*, i dva pretrenirana modela za generisanje vektora od teksta - *all-mpnet-base-v2* i *all-MiniLM-L12-v2* sa *Huggingface* platforme. Korisnicima je na sajtu omogućen odabir baze koju žele da koriste za svoj upit, dok su u svaku od njih indeksirani vektori modela *all-mpnet-base-v2*, koji je pokazao preciznost od 100% na podskupu podataka za top 4 rezultata.

Odlične performanse vektorskih baza, čak i u sklopu besplatnog plana na veoma ograničenoj memoriji i računskoj snazi, i pomenutih modela koji su otvorenog pristupa, i mnogo manji u odnosu na komercijalne modele koje nudi OpenAI platforma, predstavljaju izuzetno dobru osnovu za dalju nadogradnju sistema.

5.1 Dalji razvoj sistema

Da bi se dostigao nivo komercijalnog proizvoda, neophodno je unaprediti postojeći sistem baš u ovom smeru. Pretplatom na produkcionu plan vektorskih baza obezbedila bi se veća brzina i računarska moć za veliki broj konkurentnih korisničkih sesija. Adaptacijom rešenja za proizvoljne skupove podataka određenog formata omogućilo bi neograničenu primenu u svakom polju industrije. Dalje, prelaskom na komercijalne modele za generisanje vektora obezbedilo bi višestruko bolju preciznost i zadovoljstvo korisnika. Radi poređenja, trenutno najperformantniji model *OpenAI* platforme, *text-embedding-3-large*, generiše vektore sa 3072 dimenzije, što je tačno 4 puta više od modela *all-MiniLM-L12-v2*, koji se u ovom radu pokazao najbolje. Shodno tome, vektorske reprezentacije indeksiranih podataka bile bi višestruko kvalitetnije i pretraga bi pokazivala još bolje rezultate.

6. LITERATURA

- [1] S. J. Russel i P. Norvig, u *Artificial intelligence: a modern approach*, New Jersey, Pearson Education, 2010, pp. 1-29.
- [2] J. R, „kaggle.com,“ [Na mreži]. Available: <https://www.kaggle.com/datasets/jrobischo/wikipedia-movie-plots>.
- [3] pinecone.io, „<https://www.pinecone.io/>,“ pinecone.io. [Na mreži].
- [4] „weaviate.io,“ Weaviate, [Na mreži]. Available: <https://weaviate.io/developers/weaviate>. [Poslednji pristup 16 February 2024].
- [5] „qdrant.tech,“ Qdrant, [Na mreži]. Available: <https://qdrant.tech/documentation/>. [Poslednji pristup 16 February 2024].
- [6] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue i A. Moi, „HuggingFace's Transformers: State-of-the-art Natural Language Processing,“ Hugging Face, Brooklyn, 2020.

Kratka biografija:



Srđan Šuković rođen je 11.01.2000. godine u Zrenjaninu. Smer računarstvo i automatika na Fakultetu tehničkih nauka u Novom Sadu upisao je 2018 godine. Osnovne studije završio je u septembru 2022. godine. Od oktobra 2022. godine upisuje master studije i počinje da radi kao softverski inženjer.

EKSTRAKTIVNA SUMARIZACIJA RECENZIJA HOTELA UPOTREBOM WORD2VEC, DOC2VEC I GPT-3.5 MODELA**EXTRACTIVE TEXT SUMMARIZATION OF HOTEL REVIEWS USING WORD2VEC, DOC2VEC AND GPT-3.5 MODELS**

Sonja Tomčić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je prikazana ekstraktivna sumarizacija recenzija hotela upotrebom Word2Vec, Doc2Vec i GPT-3.5 modela. Objašnjene su teorijske osnove NLP-a, pomenutih modela i TextRank algoritma. Prikazan je proces sumarizacije teksta korišćen u implementaciji ovog rešenja. Skup podataka za treniranje i evaluaciju modela je preuzet sa sajta booking.com. Podaci su javno dostupni i nalaze se u vlasništvu sajta. U radu su prikazani eksperimenti izvršeni upotrebom različitih varijanti modela, kao i upotrebom različitih skupova podataka. Rezultati eksperimenata su poređeni međusobno, kao i sa rezultatima jednog od srodnih radova.

Ključne reči: Ekstraktivna sumarizacija, NLP, Word2Vec, Doc2Vec, GPT-3.5, TextRank

Abstract – This paper shows the extractive text summarization of hotel reviews using Word2Vec, Doc2Vec and GPT-3.5 models. The theoretical foundations of the NLP, the models and the TextRank algorithm are explained. The process of text summarization used in the implementation of this solution is presented. The dataset used for model training and evaluation was downloaded from the booking.com. The data is publicly available, and it is owned by the website. The paper shows experiments performed using different variations of the models, as well as using different datasets. The results of the experiments are compared with each other, as well as with the results of one of the related works.

Keywords: Extractive text summarization, NLP, Word2Vec, Doc2Vec, GPT-3.5, TextRank

1. UVOD

Razvoj tehnologije i platformi za pronalaženje i rezervaciju smeštaja omogućio je njihovim korisnicima da brzo i lako sami pronalaze smeštaj bez dodatnih agencijskih troškova. Pomenute platforme pružaju mogućnost ostavljanja recenzija, što u značajnoj meri utiče na izbor potencijalnih korisnika. Komentari iz recenzija umeju da budu dugački i teški za čitanje i tumačenje, a njihova sumarizacija može da skрати vreme potrebno za odabir smeštaja i da na taj način korisnicima

olakša donošenje odluke. Iako veoma korisna, ova opcija nije česta na platformama za pronalaženje i rezervaciju smeštaja.

Motivacija za pisanje ovog rada jeste to što bi uvođenje funkcionalnosti za sumarizaciju recenzija hotela omogućilo korisnicima tih platformi da brže i lakše donesu odluku prilikom odabira i rezervacije smeštaja. U radu je prikazan sistem za ekstraktivnu sumarizaciju recenzija hotela. Objašnjene su teorijske osnove NLP-a, Word2Vec, Doc2Vec i GPT-3.5 modela i TextRank algoritma, koji su korišćeni u implementaciji rešenja. Skup podataka je preuzet sa sajta booking.com, a podaci su javno dostupni i nalaze se u vlasništvu sajta.

Prikazan je celokupan proces sumarizacije, kao i eksperimenti izvršeni upotrebom različitih varijanti pomenutih modela i upotrebom različitih skupova podataka. Rezultati eksperimenata su poređeni međusobno, kao i sa rezultatima jednog od srodnih radova. Problem sumarizacije teksta, rešavan u ovom radu, pripada NLP (Natural Language Processing) oblasti, grani veštačke inteligencije koja se fokusira na interakciju između kompjutera i ljudskog jezika.

2. PREGLED STANJA U OBLASTI

U radu [1] iz 2019. godine autori su prikazali upotrebu Word2Vec i Doc2Vec algoritama u kombinaciji sa TextRank algoritmom, za pronalaženje ključnih reči u kratkim tekstovima sa društvenih mreža. Word2Vec algoritam je upotrebljen za pronalaženje semantičkih veza između reči, dok je Doc2Vec algoritam korišćen za pronalaženje vektora paragrafa i pronalaženje tačnosti. Rešenje je evaluirano pomoću accuracy, recall i F-measure metrika. Opisivana metoda se pokazala kao najbolja kada je u pitanju ekstrakcija ključnih reči kod kratkih tekstova, ali radi pouzdanosti i kod dugih tekstova. U radu [1] Word2Vec i Doc2Vec algoritmi su kombinovani kako bi se dobila semantička veza reči u tekstu, dok su u ovom radu te dve metode razdvojene.

Rad [2] iz 2020. godine opisuje rešenje problema sumarizacije biomedicinskih informacija u naučnim člancima, medicinskim zapisima, web dokumentima i kliničkim slikama. Za pronalazak lingvističke, semantičke i kontekstualne veze između rečenica, u radu [2] koriste se Word2Vec SkipGram i CBOW, context-free jezički modeli, kao i BioBERT, context-sensitive model. Za rangiranje koriste se PageRank, HITS i PPF tehnike. Rešenje je evaluirano upotrebom ROUGE-1 i ROUGE-2 metrike, a korpus za evaluaciju sadrži 2000 članaka iz

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kovačević, red. prof.

PubMed Central, gde je apstrakt članaka korišćen kao sumarijacija. Najbolji rezultat generalno dala je kombinacija BioBERT + GloVe + PageRank, a context-sensitive model se pokazao kao bolji u odnosu na context-free modele. U radu [2] prikazana je kombinacija različitih metoda za rangiranje i pronalazak veze između rečenica, dok su u ovom radu upotrebljene samo dve metode za traženje vektora kao dva odvojena rešenja u kombinaciji sa TextRank algoritmom.

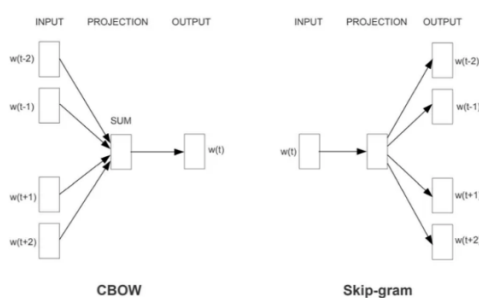
3. TEORIJSKI POJMOVI I DEFINICIJE

3.1. NLP (Natural Language Processing)

Natural Language Processing (NLP) je grana veštačke inteligencije koja omogućava računarima da procesiraju ljudski jezik u formi teksta ili govora na isti način kao što to rade ljudska bića. NLP kombinuje računarsku lingvistiku, koja predstavlja modelovanje ljudskog jezika zasnovano na pravilima, zajedno sa statističkim modelima i modelima mašinskog i dubokog učenja. Sumarijacija teksta, opisana u ovom radu, pripada NLP oblasti. Ekstraktivna sumarijacija podrazumeva identifikovanje i izdvajanje najvažnijih delova teksta, bez ikakvih modifikacija, što je tema ovog rada [3, 4].

3.2. Word2Vec

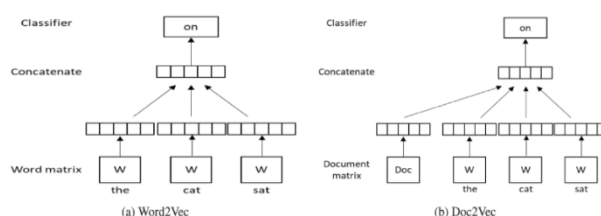
Word2Vec predstavlja metodu generisanja vektora reči pomoću neuronske mreže koja se sastoji od ulaznog, skrivenog i izlaznog sloja. Upotrebom ove metode, reči se predstavljaju kao vektori u kontinualnom vektorskom prostoru, kako bi se zabeležila semantička veza među njima. Reči sličnog značenja imaju sličnu vektorsku reprezentaciju, odnosno pozicionirane su blizu jedna drugoj u vektorskom prostoru, što je moguće predstaviti i izmeriti pomoću kosinusne sličnosti (eng. Cosine Similarity). Postoje dve različite arhitekture Word2Vec modela: Continuous Bag of Words (CBOW) i Skip-Gram. Na slici 1 prikazan je trening CBOW i Skip-Gram modela [5, 6].



Slika 1: Trening CBOW i Skip-Gram modela [7]

3.3. Doc2Vec

Doc2Vec je unsupervised algoritam za generisanje numeričkih vektora fiksne dužine koji reprezentuju dokument. Baziran je na Word2Vec modelu. Vektori se uče na sličan način kao kod Word2Vec modela - slični dokumenti se mapiraju na obližnje tačke u vektorskom prostoru, odnosno nalaze se blizu jedan drugom. Postoje dve varijante Doc2Vec modela: Distributed Memory (DM) i Distributed Bag of Words (DBOW). Na slici 2 prikazana je razlika između Word2Vec i Doc2Vec modela [8].



Slika 2: Razlika između Word2Vec i Doc2Vec modela [9]

3.4. TextRank algoritam

TextRank algoritam pripada algoritmima zasnovanim na grafovima, kod kojih se važnost temena unutar grafa određuje uzimajući u obzir globalne informacije rekursivno izračunate iz celog grafa, umesto da se oslanjaju na lokalne informacije vezane za temena. TextRank algoritam je baziran na PageRank algoritmu koji se koristi za rangiranje web stranica prilikom online pretrage. Umesto web stranica koriste se rečenice iz teksta, a matrica sličnosti je popunjena brojevima koji označavaju sličnost između rečenica [10].

3.5. GPT-3.5 Turbo model

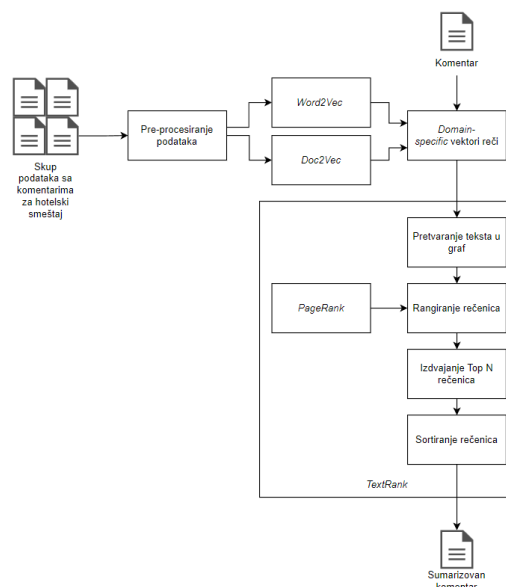
GPT-3.5 Turbo model pripada familiji GPT-3 modela i predstavlja jezički model baziran na neuronskim mrežama, treniran nad velikim skupom podataka. GPT-3.5 model je treniran nad preko 570GB podataka i sposoban je da uči kompleksne šablone u ljudskom jeziku, a njegova najbitnija osobina je duboko razumevanje ljudske komunikacije. Arhitektura GPT-3.5 modela je bazirana na transformer modelima.

Model je treniran po principu predikcije sledeće reči u rečenici, koristeći unsupervised učenje. Izgrađen je pomoću transformer dekođer blokova [11, 12].

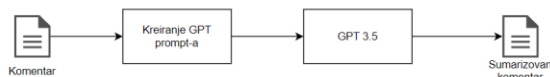
4. METODOLOGIJA

4.1. Arhitektura sistema

Za implementaciju rešenja kreirane su dve različite arhitekture. U prvoj arhitekturi upotrebljeni su Word2Vec i Doc2Vec modeli (slika 3), dok je u drugoj arhitekturi upotrebljen GPT-3.5 model (slika 4).



Slika 3: Arhitektura sistema sa Word2Vec i Doc2Vec modelima



Slika 4: Arhitektura sistema sa GPT-3.5 modelom

4.2. Implementacija rešenja

Kod arhitekture sa slike 3 pristupa su dva ulaza u sistem. Prvi ulaz je skup podataka sa komentarima iz recenzija za hotelski smeštaj. Komentar je potrebno preprocesirati kako bi bili pogodni za treniranje Word2Vec i Doc2Vec modela. Nakon preprocesiranja, modeli se treniraju nad domain-specific skupom reči.

Drugi ulaz u sistem je komentar koji je potrebno sumariizovati. Na osnovu ulaznog komentara se prave domain-specific vektori reči pomoću istreniranih Word2Vec i Doc2Vec modela. U narednoj fazi rešenja, koja podrazumeva upotrebu TextRank algoritma, vektori reči iz prethodnog koraka se koriste za kreiranje grafa, a graf se dalje koristi za rangiranje rečenica pomoću PageRank algoritma, nakon čega se izdvaja top N rangiranih rečenica, koje se na kraju sortiraju. Izlaz iz sistema je sumariizovan ulazni komentar.

Na ulazu u sistem prikazan na slici 4 nalazi se originalni komentar. On se zatim koristi za kreiranje prompt-a, koji se prosleđuje GPT-3.5 modelu. Izlaz iz sistema predstavlja sumariizovan komentar.

5. EKSPERIMENTI

5.1. Skup podataka

Skup podataka za obučavanje modela korišćenih u arhitekturi sa slike 3 napravljen je scrape-ovanjem podataka sa sajta booking.com. Svi podaci su već javno dostupni i nalaze se u vlasništvu sajta.

Ovaj skup podataka kreiran je samo za potrebe rada i ne koristi se u druge svrhe, a sadrži komentare iz recenzija za smeštaj u hotelima u mestu Puket. Sa web sajta je preuzeto 27055 komentara. Obučavajući skup podataka se sastoji od 2188 komentara na engleskom jeziku, dužine od 30 ili više reči. Ukupan broj rečenica je 9906, dok je ukupan broj reči 72871, nakon što su izbačene engleske stop words. Za deo rešenja koje koristi arhitekturu sa slike 4 nije potreban obučavajući skup podataka.

5.2. Organizacija eksperimenata

Rešenje je evaluirano na osnovu četiri eksperimenta. U sva četiri eksperimenta koriste se Word2Vec (CBOW i Skip-gram), Doc2Vec (DBOW i DM) modeli i njihove vrste, kao i GPT-3.5 model. Za svaku od navedenih vrsta modela, podaci su preprocesirani na tri načina: bez upotrebe Stemming i Lemmatization procesa, upotrebom Stemming procesa i upotrebom Lemmatization procesa. Sistem je testiran tako da vraća sumariizovane komentare dužine od tri i pet rečenica. U sva četiri eksperimenta postavka je ista, a razlikuje se jedino skup podataka za evaluaciju.

5.3. Eksperiment 1

U Eksperimentu 1 isti skup podataka se koristi za treniranje modela i za evaluaciju. Evaluacioni skup sadrži 20 kratkih komentara. Dužina originalnih komentara je 8 rečenica, a svaki komentar ima manje od 500 karaktera.

5.4. Eksperiment 2

U Eksperimentu 2 isti skup podataka se koristi za treniranje modela i za evaluaciju. Evaluacioni skup sadrži 20 dugačkih komentara. Dužina originalnih komentara je 10-12 rečenica, a svaki komentar ima više od 800, a manje od 1200 karaktera.

5.5. Eksperiment 3

U Eksperimentu 3 se skup podataka za treniranje modela i za evaluaciju razlikuju. Skup podataka za evaluaciju napravljen je ručno kopiranjem komentara za smeštaj na drugoj lokaciji. Za evaluaciju je izdvojeno 20 kratkih komentara, dužine kao u Eksperimentu 1.

5.6. Eksperiment 4

U Eksperimentu 4 se skup podataka za treniranje modela i za evaluaciju razlikuju. Skup podataka za evaluaciju napravljen je ručno kopiranjem komentara za smeštaj na drugoj lokaciji. Za evaluaciju je izdvojeno 20 dugačkih komentara, dužine kao u Eksperimentu 2.

5.7. Evaluacija

Za evaluaciju rezultata upotrebljen je ROUGE set metrika za evaluaciju automatske sumariizacije teksta na osnovu manuelno kreiranih referentnih sumariizacija, a dodatno rezultati su evaluirani i manuelno, na osnovu subjektivnog mišljenja autora. Korišćene su ROUGE-1, ROUGE-2 i ROUGE-L metrike. Rezultati ovih metrika prikazuju se pomoću recall, precision i F1 Score metrika. U navedenim eksperimentima prikazane su vrednosti F1 Score metrike, koja predstavlja harmonijsku sredinu precision i recall metrika.

6. REZULTATI

U tabeli 1 prikazani su najbolji rezultati iz sva četiri eksperimenta. Najbolji rezultat imao je Eksperiment 3 u kombinaciji Word2Vec CBOW + Stemming za sumariizaciju od 5 rečenica, što je delom iznenađujuće jer CBOW model u poređenju sa SkipGram modelom nije efikasan u beleženju nijansiranih veza između reči. Ovaj eksperiment vršen je nad novim komentarima koji su nepoznati sistemu, dok su rezultati Eksperimenta 1 i Eksperimenta 2, koja su vršena nad komentarima koji su već poznati sistemu, veoma slični mogu se smatrati jednako uspešnim. S obzirom na to da ovaj eksperiment nije mogao da se izvrši za pojedine Doc2Vec varijante, njegov rezultat se može smatrati nedovoljno relevantnim.

Tabela 1: Ukupni rezultati eksperimenata

Eksperiment	Model	Pre-processing	N	ROUG E-1 F1	ROUG E-2 F1	ROUG E-L F1
Eksp. 1	Doc2Vec DBOW	None	5	0.7119	0.6417	0.7099
Eksp. 2	Doc2Vec DM	None	3	0.7091	0.6389	0.7073
Eksp. 3	Word2Vec CBOW	Stemming	5	0.7507	0.6848	0.7470
Eksp. 4	Word2Vec SG	Lemmatization	5	0.6502	0.5509	0.6437

7. DISKUSIJA

U radu [2], gde su sumariizovani biomedicinski članci, korišćene su različite metode za rangiranje rečenica i kreiranje vektora reči. Rezultati su se pokazali boljim, ako se u obzir uzmu samo rezultati za kombinaciju Word2Vec + PageRank, s obzirom na to da su njeni rezultati relevantni za ovaj rad. Uzimajući u obzir veličinu skupova podataka za razvoj i evaluaciju, kao i to da su za referentne sumariizacije uzeti apstrakti članaka koji predstavljaju objektivnu sumariizaciju, za razliku od subjektivnih referentnih sumariizacija korišćenih u ovom radu, očekivano je da su vrednosti rezultata u radu [2] bolji u odnosu na rezultate u ovom radu.

Subjektivno mišljenje autora ovog rada upotrebljeno je kao dodatna metoda evaluacije pored ROUGE metode. Manuelno su evaluirani komentari za koje su vrednosti rezultata najbolje (tabela 1), kao i oni koji imaju najlošije rezultate. Može se zaključiti da za primere gde je rezultat dobar, sumariizacija daje smislene komentare. Najbolji rezultati su prisutni kod sumariizacija od pet rečenica, što ima smisla jer algoritam u tim slučajevima mora da izabere veći broj rečenica i to daje manje šanse za grešku. S obzirom na nemogućnost modela da konvergiraju u eksperimentima 3 i 4, ovaj sistem nije spreman za upotrebu u produkciji.

Probleme sa nemogućnošću sistema da konvergira je moguće rešiti promenom parametara modela i ponovnim testiranjem. Pored promene parametara modela, moguće unapređenje bi bilo i treniranje modela nad većim i raznovrsnijim skupom podataka. Evaluaciju ovog sistema je moguće poboljšati kreiranjem većeg evaluacionog skupa, tako da je on kreiran od strane više ljudi, kako bi se isključila subjektivnost jedne osobe, ili pronalaskom skupa podataka na internetu. Promenom načina kreiranja evaluacionog skupa bi i rezultati eksperimenata bili drugačiji, a najverovatnije i tačniji.

8. ZAKLJUČAK

U ovom radu predstavljen je sistem za ekstraktivnu sumariizaciju komentara iz recenzija za hotelski smeštaj. Dva od tri rešenja su implementirana u vidu dva odvojena dela: modul za treniranje modela za kreiranje vektora reči i modul za rangiranje rečenica. Treće rešenje ne koristi vektore reči niti algoritam za rangiranje rečenica i veoma je jednostavno za implementaciju.

Word2Vec CBOW model u kombinaciji sa Stemming pretprocesiranjem za sumariizacije od pet rečenica se pokazao kao najbolji uzimajući u obzir varijante modela i pretprocesiranja u kom je sistem uspešno izvršio sumariizaciju. Ovaj eksperiment je vršen nad kratkim nepoznatim komentarima. Rezultati dva eksperimenta koja su vršena nad komentarima koji su već poznati sistemu imaju veoma slične vrednosti i mogu se smatrati jednako uspešnim. Doc2Vec model je generalno imao bolje rezultate, ali za pojedine modele sistem nije uspeo da izvrši sumariizaciju do kraja, pa zbog toga određeni rezultati fale, a možda bi doprineli drugačijem ishodu i zaključku o uspešnosti sistema. GPT-3.5 model je bio srednje uspešan i sva četiri eksperimenta se slično ponašao.

Uspešnost ovog sistema mogla bi se povećati upotrebom većeg i raznovrsnijeg skupa podataka. Metode evaluacije

je takođe moguće poboljšati kreiranjem većeg i manje subjektivnog evaluacionog skupa koji bi bio kreiran od strane više ljudi, što bi doprinelo drugačijim i tačnijim rezultatima. Kao dodatno poboljšanje sistema za ekstraktivnu sumariizaciju komentara, drugi algoritmi za rangiranje i kreiranje vektora reči bi mogli da se testiraju. Sistem bi dodatno mogao da se proširi mogućnošću da radi i za ostale jezike, a ne samo za engleski jezik.

9. LITERATURA

- [1] LI, JUN; HUANG, GUIMIN; FAN, CHUNLI; SUN, ZHENGLIN; and ZHU, HONGTAO (2019) "Key word extraction for short text via word2vec, doc2vec, and textrank," *Turkish Journal of Electrical Engineering and Computer Sciences*: Vol. 27: No. 3, Article 17.
- [2] Milad Moradi, Maedeh Dashti, Matthias Samwald, "Summarization of biomedical articles using domain-specific word embeddings and graph ranking", *Journal of Biomedical Informatics*: Vol. 107, July 2020, 103452.
- [3] <https://www.geeksforgeeks.org/natural-language-processing-overview/> (pristupljeno u januaru 2024.)
- [4] <https://medium.com/sciforce/towards-automatic-text-summarization-extractive-methods-e8439cd54715> (pristupljeno u februaru 2024.)
- [5] <https://www.geeksforgeeks.org/python-word-embedding-using-word2vec/> (pristupljeno u februaru 2024.)
- [6] <https://wiki.app.uib.no/info216/images/c/c4/IntroToWordEmbeddings.pdf> (pristupljeno u februaru 2024.)
- [7] <https://medium.com/@manansuri/a-dummys-guide-to-word2vec-456444f3c673> (pristupljeno u februaru 2024.)
- [8] <https://www.geeksforgeeks.org/doc2vec-in-nlp/> (pristupljeno u februaru 2024.)
- [9] A. Hernández-Castñeda, R. A. García-Hernández, Y. Ledeneva and C. E. Millán-Hernández, "Extractive automatic text summarization based on lexical-semantic keywords", *IEEE Access*, vol. 8, pp. 49896-49907, 2020.
- [10] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text", *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 404-411, July 2004.
- [11] <https://www.linkedin.com/pulse/chatgpts-guide-understanding-gpt-35-architecture-heena-koshti#:~:text=The%20GPT%2D3.5%20architecture%20is%20a%20neural%20network%2Dbased%20language,txt%20completion%2C%20and%20question%20answering> (pristupljeno u februaru 2024.)
- [12] <https://medium.com/nerd-for-tech/gpt3-and-chat-gpt-detailed-architecture-study-deep-nlp-horse-db3af9de8a5d> (pristupljeno u februaru 2024.)

Kratka biografija:



Sonja Tomčić rođena je 04.02.1999. godine u Vrbasu. Osnovne akademske studije završila je 2022. godine na Fakultetu tehničkih nauka, nakon kojih upisuje master akademske studije na studijskom programu Primenjene računarske nauke i informatika, usmerenje Inteligentni sistemi. kontakt: sonjatomcic@gmail.com

БЕЖИЧНИ ПРЕНОС ЕЛЕКТРИЧНЕ ЕНЕРГИЈЕ ЗА ПОТРЕБЕ ПУЊЕЊА ЛАКИХ ЕЛЕКТРИЧНИХ ВОЗИЛА**WIRELESS TRANSMISSION OF ELECTRICAL ENERGY FOR THE CHARGING NEEDS OF LIGHT ELECTRIC VEHICLES**Мирослав Богдановић, Владимир Рајс, *Факултет техничких наука, Нови Сад***Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО**

Кратак садржај – У овом раду приказана је једна реализација система за бежични пренос енергије за потребе пуњења лаких електричних возила. Под лаким електричним возилима се сматрају сва возила која се користе за превоз мањег броја људи или мања транспортна возила. Најчешћи представници лаких електричних возила су путнички аутомобили и за њих је првенствено и дизајниран дати систем. Наведени систем омогућава пуњење батерија унутар возила без галванског споја између пуњача и возила. Ово резултује великом погодношћу приликом коришћења оваквог система јер је довољно само возило превести на одговарајуће место где се налази систем за пуњење. Систем користи резонантно индуктивно спрежање што је једна од метода за бежични пренос електричне енергије. Овај метод је препоручен за бежично пуњење лаких електричних аутомобила у стандарду који описује овај вид пуњења и по ком стандарду је и дизајниран цео систем. Управљање целим системом је базирано на развојном систему Микромедиа 7.

Кључне речи: Микроконтролер, бежични пренос електричне енергије, енергетска електроника, резонантно индуктивно спрежање, батерије за електричне аутомобиле.

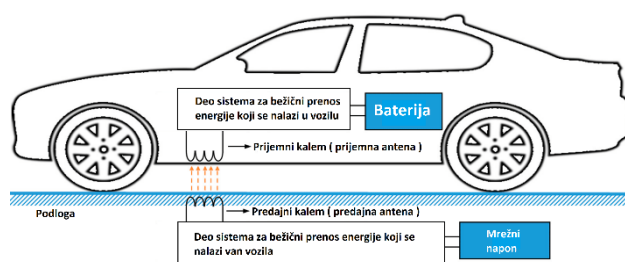
Abstract – This paper presents one realization of a system for wireless energy transmission for charging light electric vehicles. Light electric vehicles are considered to be all vehicles used for the transport of a small number of people or smaller transport vehicles. The most common representatives of light electric vehicles are passenger cars, for which this system was primarily designed. The mentioned system enables the charging of batteries inside the vehicle without a galvanic connection between the charger and the vehicle. This results in great convenience when using this type of system because it is enough to bring the vehicle to the appropriate place where the charging system is located. The system uses resonant inductive coupling, which is one of the methods for wireless transmission of electricity. This method is recommended for wireless charging of light electric cars in the standard that describes this type of charging and according to which standard the entire system is

designed. Management of the entire system is based on the development system Mikromedia 7.

Keywords: Microcontroller, wireless power transmission, power electronics, resonant inductive coupling, batteries for electric cars.

1. УВОД

Електрични аутомобили су тренутно предмет истраживања многих водећих компанија у аутомобилској индустрији, стога како напредује развој електричних возила, тако се ради и на развијању пуњача, било обичних или бежичних. Бежични пуњачи су бољи по питању робусности и лакоће коришћења јер не захтевају каблове, прикључке и механичке контакте између возила и пуњача [1].



Слика 1. Графички приказ једног начина позиционирања система у односу на возило

На слици 1. приказан је један начин позиционирања система у односу на возило. Груба блок шема система је такође приказана на слици 1.

2. АНАЛИЗА ПРОБЛЕМА

Пре почетка рада на самом дизајну система за бежични пренос енергије дефинисани су захтеви које би такав систем требао да испуњава [3]. Финални списак захтева састојао се од 9 појединачних захтева:

- 1) Избор напона напајања
- 2) Одређивање растојања на коме се преноси енергија
- 3) Варијабилни износ пренесене снаге
- 4) Максимална снага
- 5) Графички интерфејс
- 6) Потрошач

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Владимир Рајс, ванр. проф.

- 7) Актуелни стандард у аутоиндустрији
- 8) Модуларност
- 9) Робусност система и температурни дизајн

2.1. Избор напајања система

Захтев број 1 говори о типу извора електричне енергије из кога се извлачи енергија. Због распрострањености мрежног напајања овај систем се може напајати из монофазног напона (220VAC) или трофазног напона (380VAC). Ово напајање је приказано на слици 1 као блок мрежни напон.

2.2. Одређивање растојања на коме се преноси енергија

Захтев број 2 говори о растојању на коме се преноси енергија. У нашем случају то је растојање између предајног калема (антене) и пријемног калема (антене). На слици 1 то је размак између шасије аута и подлоге. За ово растојање узето је растојање од 10cm.

2.3. Варијабилни износ пренесене снаге

Захтев број 3 говори о томе да се мора вршити регулација снаге тј. да се количина пренете снаге може одабирати у опсегу од 0 до неке максималне вредности пренете снаге (дефинисане у поглављу 2.4.).

2.4. Максимална снага

Захтев 4 говори о максималној могућој снази која се може пренети овим системом. Овај систем је пројектован за максимални пренос снаге од 12.5kW. Ова вредност је изабрана спрам класификацији бежичних пуњача која је дата у стандарду (стандард описан у поглављу поглављу 2.7.).

2.5. Графички интерфејс

Захтев 5 говори да на систему мора постојати интуитиван кориснички интерфејс који обезбеђује промену жељене пренесене снаге (у опсегу дефинисаном у поглављу 2.3. и 2.4.) као и информацију о тренутном стању пуњача (батерија се пуни или батерија се не пуни).

2.6. Потрошач

Захтев 6 говори о типу потрошача коме се шаље енергија. Овде је захтев да је потрошач увек батерија за лако електрично возило чији напон може ићи до више стотина волти.

2.7. Актуелни стандард у аутоиндустрији

Захтев број 7 говори да систем мора бити реализован у складу са актуелним стандардом за бежичне пуњаче у аутоиндустрији.

У априлу 2019. године изашао је стандард SAE J2954 кога је издала организација SAE (*Society of Automotive Engineers*) [3]. Овај стандард је дефинисао више ствари а испод су дате најважније:

- 1) Дефинисана подела система по снази и циљаној ефикасности
- 2) Дефинисана подела система по растојању на коме се преноси енергија

2.7.1. Подела система по снази и циљаној ефикасности

Стандард дефинише 4 класе система за бежични пренос пуњења. Подела је приказана на табели 1.

Табела 1. Подела по снази и циљаној ефикасности према стандарду SAE J2954

	WPT класе по снази			
	WPT1	WPT2	WPT3	WPT4
Maks. snaga ka bateriji vozila [kVA]	3.7	7.7	11.1	22
Min. efikasnost pri x,y poravnanju [%]	>85	>85	>85	biće definisano u narednim dopunama standarda

Према наведеној подели систем који је реализован припада класи WPT3.

2.7.2. Подела система по растојању на коме се енергија преноси

Стандард дефинише 3 класе система за бежични пренос пуњења. Подела је приказана на табели 2.

Табела 2. Подела по растојању на коме се енергија преноси према стандарду SAE J2954

Z класа	Размак између предајника (GA) и пријемника (VA)
Z1	100 до 150
Z2	140 до 210
Z3	170 до 250

Према наведеним поделама систем који је реализован припада класи WPT3 Z1.

2.8. Модуларност

Захтев број 8 говори да систем мора бити модуларан тј. да му делови буду лако замењиви и да се исти делови система могу користити у различитим врстама система за бежични пренос електричне енергије.

2.9. Робусност система и температурни дизајн

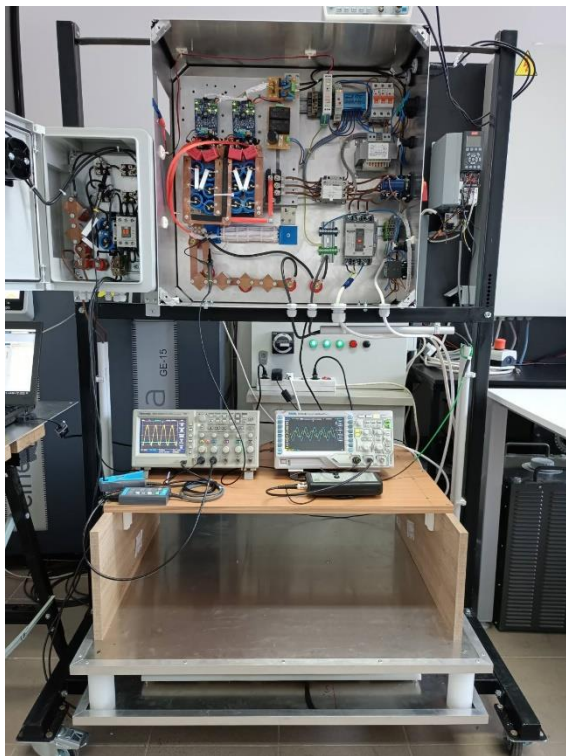
Захтев број 9 говори да систем мора да задовољава основне захтеве за робусност, отпорност на практична оптерећења као што су вибрације, благи удари и сл.

Такође, систем мора задовољити температурне захтеве тј. да при континуалном раду при максималној снази температура не прелази неку претходно изабрану температуру при којој систем ради очекивано.

3. РЕАЛИЗОВАНО РЕШЕЊЕ ЗА БЕЖИЧНИ ПРЕНОС ЕЛЕКТРИЧНЕ ЕНЕРГИЈЕ

На основу разматрања и захтева обрађених у претходном поглављу долазимо до реализације система за бежични пренос електричне енергије који испуњава све наведене захтеве.

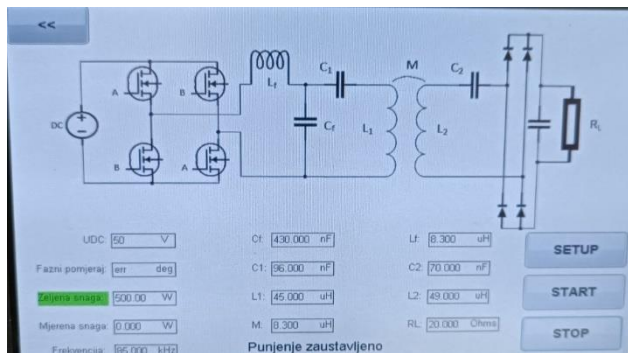
Слика 2. приказује реализовани систем у лабораторијском окружењу.



Слика 2. Реализовани систем за бежични пренос електричне енергије

3.1. Управљање системом за бежични пренос електричне енергије

Управљање системом се врши преко *Mikromedia 7* [2] развојног система који на себи има екран осетљив на додир од 7". Приказ графичког интерфејса преко кога се управља системом приказан је на слици 3.



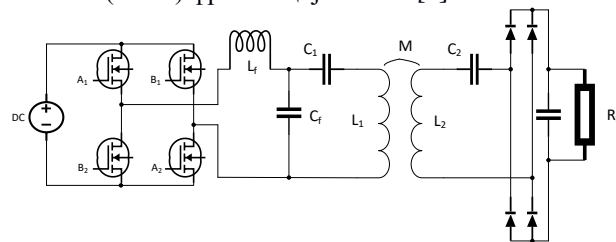
Слика 3. Графички интерфејс за управљање пуњењем

Контрола пуњења врши се тако што се на дисплеју изабере жељена снага (на слици 3. означено зеленом бојом) и притисне дугме *START*. Уколико желимо да искључимо пуњење притиснемо дугме *STOP* а уколико систем оставимо да ради на дуже пуњење ће се само угасити када батерија која се пуни достигне одређени претходно дефинисан напон. Када достигне тај напон сматрамо да је батерија пуна и пуњење се завршава.

3.2. Фазна регулација

Принцип контроле снаге пуњења биће објашњен на поједностављеној електричној шеми система приказаној на слици 4. Контрола снаге пуњења се врши фазним кашњењем између контролних сигнала на

инвертору реализованом као потпуни мост (*full bridge inverter*) [4]. Инвертор је имплементиран са 4 мосфета и гејтови наведених мосфетова су означени са A_1, B_1, A_2, B_2 . Контролни сигнали на гејтовима инвертора су импулсно-ширинско модулисани сигнали (*PWM*) фреквенције 85kHz [5].



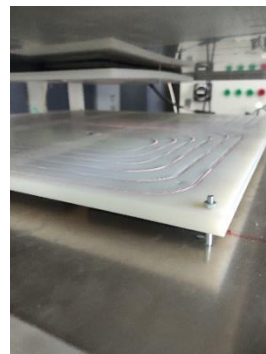
Слика 4. Поједностављена електрична шема бежичног пуњача

Фазним кашњењем сигнала $A_2 \overline{B_1}$ у односу на $A_1 \overline{B_2}$ добија се да излаз инвертора већи део времена на свом излазу даје ненулту вредност напона што резултује већом пренешеном снагом. Као параметар, фазно кашњење се означава са α и може имати вредност од 0° до 180° степени при чему за 0° степени сигнали су у фази и тада је пренешена снага 0W док ако је фазно кашњење 180° онда је снага максимална. Након одабирања жељене снаге на дисплеју *Mikromedia 7* прорачунава фазно кашњење и испишује га на дисплеј (слика 3.). Након притиска на дугме старт почиње и генерисање сигнала на улаз инвертора са датим фазним кашњењем и тиме се започиње пренос енергије.

3.4. Спрегнути калемови

Бежични пренос електричне енергије одвија се у простору између предајног и пријемног калема (приказано на слици 1). Калемови се налазе на удаљености од 10 cm у складу са захтевом број 2 објашњеним у поглављу 2.2. Наведени калемови су направљени према упутствима из стандарда *SAE J2954*. Калемови су приказани на доњем делу слике 2.

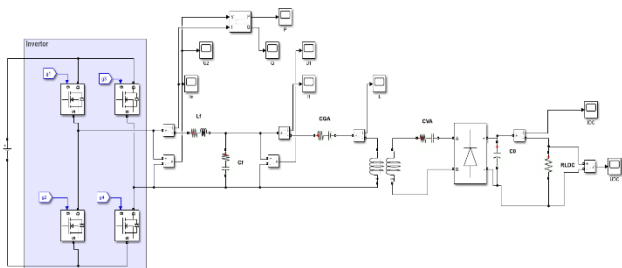
Простор између калемова приказан је на слици 5. За намотавање калемова коришћен је лицнасти проводник код којег су изоловане све лицне засебно. Ово је потребно јер је сигнал који пролази кроз намотаје 85 kHz па би ту доминирао скин ефекат уколико би се користио класичан лицнасти или пун проводник.



Слика 5. Спрегнути калемови (доњи калем је предајни док је горњи пријемни)

4. СИМУЛАЦИЈА РЕАЛИЗОВАНОГ РЕШЕЊА ЗА БЕЖИЧНИ ПРЕНОС ЕЛЕКТРИЧНЕ ЕНЕРГИЈЕ

За систем приказан на слици 2. направљена је симулација у програмском пакету *Simulink* у којој је детаљно моделован систем са циљем да се истраже његове карактеристике и понашање. Модел система, приказан у симулацији, дат је на слици 6.

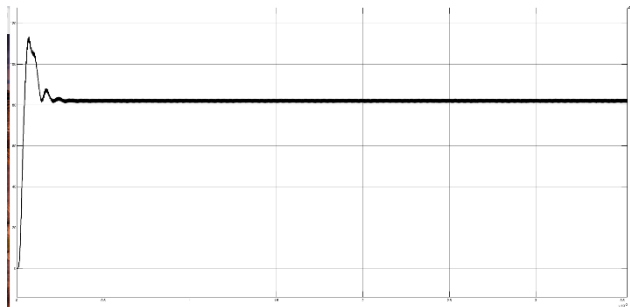


Слика 6. Модел система за бежични пренос електричне енергије (у програмском пакету *Simulink*)

Наведена симулација се поклапа са мерењима на физичком систему у толеранцији од 5% чиме се сматра да је симулација задовољавајуће тачности.

4.1. Резултати симулације

Најбитнији сигнали са симулације су напон на потрошачу и излазна струја кроз потрошач. На сликама 7 и 8 приказани су ови сигнали, где се може видети да су вредности у устаљено стању 82 VDC односно 8,2A а параметри система су подешени за пуњење батерије снагом од 0,6kW.



Слика 7. Напон пуњења батерије



Слика 8. Струја пуњења батерије

5. ЗАКЉУЧАК

Задатак овог рада био је дизајн и реализација једног система за бежични пренос енергије који задовољава захтеве који су наведени пре почетка пројекта

(захтеви описани у поглављу број 2). Намена овог уређаја је вишеструка али је акценат био на аутоиндустрији и улози једног оваквог система за бежично пуњење возила. Кроз итерације, наведени уређај је пролазио кроз разне преправке све док није дошао до финалног изгледа какав је описан у поглављу број 3.

За наведени уређај направљена је његова виртуелна копија тј. модел који у симулацији описује рад реалног система. Планови за будућност овог система укључују реализацију мода пуњења константном струјом да би се остварило оптимално пуњење литијум-јонских батерија.

6. ЛИТЕРАТУРА

[1] Опис и подела технологија за бежични пренос електричне енергије:

https://en.wikipedia.org/wiki/Wireless_power_transfer [фебруар 2024.]

[2] *MIKROE, Mikromedia 7 for STM32F7 – User Manual:*

<https://www.mikroe.com/blog/mikromedia-7-stm32f7-user-manual> [фебруар 2024.]

[3] Стандард *SAE J2954:*

https://www.sae.org/standards/content/j2954_202010/ [фебруар 2024.]

[4] Објашњење инвертора:

<https://circuitdiagrams.in/full-bridge-inverter/> [март 2024.]

[5] Објашњење ширинско-импулсно модулисаног сигнала:

<https://resources.pcb.cadence.com/blog/2020-pulse-width-modulation-characteristics-and-the-effects-of-frequency-and-duty-cycle> [март 2024.]

Кратка биографија:



Мирослав Богдановић рођен је у Чачку 1997. год. Дипломирао је на Факултету техничких наука из области електротехнике и рачунарства (примењена електроника) 2022. године. Ради као сарадник у настави на катедри за електронику Факултета техничких наука у Новом Саду.



Владимир Рајс рођен је 1982. год. у Апатину. Дипломирао је 2007, а докторирао 2015. године на Факултету техничких наука у Новом Саду. Од 2016. године запослен је као доцент, а од 2021. као ванредни професор на Департману за енергетику, електронику и телекомуникације ФТН-а. Области интересовања су му електроника и примењена електроника.

USLUGA UPRAVLJANJA KLJUČEVIMA ZA AMAZON VEB SERVICE AMAZON WEB SERVICES KEY MANAGEMENT SERVICE

Plema Lažetić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Ovaj rad istražuje AWS Key Management Service, ključnu komponentu u AWS-ovom okruženju, fokusirajući se na procese upravljanja kriptografskim ključevima i osiguranje sigurnosti podataka u oblaku. Problemi koje se razmatraju obuhvataju izbor, kreiranje, upravljanje i integraciju ključeva u različitim AWS uslugama. Rezultati istraživanja doprinose razumijevanju primjene sigurnosnih mjera u cloud okruženju i prepoznavanju mogućnosti za poboljšanje prakse upravljanja ključevima u AWS-ovom okruženju.

Ključne reči: Usluga upravljanja ključevima na AWS-u, kriptografski ključevi, upravljanje pristupom.

Abstract – This paper explores the AWS Key Management Service, a crucial component in the AWS environment, focusing on cryptographic key management processes and data security assurance in the cloud. Issues under consideration encompass the selection, creation, management, and integration of keys across various AWS services. The research findings contribute to understanding the implementation of security measures in the cloud environment and identifying opportunities for enhancing key management practices within the AWS environment.

Keywords: AWS Key Management Service, cryptographic keys, access management.

1. UVOD

U današnjem digitalnom dobu, programiranje u oblaku postaje sve prisutnija paradigma u razvoju softvera. Ova tehnologija omogućava efikasnije korišćenje resursa, skalabilnost aplikacija i olakšano upravljanje infrastrukturom [1].

Amazon Web Services (AWS) izdvaja se kao lider u pružanju infrastrukture u oblacima i različitih usluga za razvoj aplikacija. U okviru AWS-ovog okruženja, AWS Key Management Service (KMS) zauzima značajno mjesto kao ključni alat za upravljanje kriptografskim ključevima i osiguranje bezbjednosti podataka u oblacima.

Ovaj rad istražuje AWS KMS, analizirajući njegovu arhitekturu, funkcionalnosti i praktičnu primjenu.

Fokus je na karakteristikama ključeva, procesu upravljanja ključevima, politikama pristupa i integraciji sa drugim

AWS uslugama, kako bi se istražilo kako AWS KMS doprinosi bezbjednosti i poverljivosti softverskih sistema u oblacima.

Kroz detaljnu analizu ovih elemenata, ovaj rad će prikazati ulogu i značaj AWS KMS-a u kontekstu programiranja u oblacima, pružajući uvid u najbolje prakse i uputstva za efikasno korišćenje ove ključne komponente AWS-ovog okruženja.

2. AMAZON WEB SERVICES

Amazon Web Services su vodeći provajderi usluga računarstva u oblaku. Pružaju raznolike usluge za izgradnju, upravljanje i skaliranje IT infrastrukture i aplikacija. Njihova globalna prisutnost omogućuje korisnicima pristup računarskim resursima širom svijeta. Model "plaćaj koliko koristiš" eliminiše potrebu za kupovinom skupe hardverske infrastrukture unaprijed. AWS nudi širok spektar alata za zaštitu podataka, uključujući enkripciju i kontrolu pristupa. Svojim korisnicima omogućuje brzo pokretanje aplikacija, skaliranje resursa prema potrebama i plaćanje samo za korištene resurse. Sa preko 25 regija i 80 zona dostupnosti, AWS olakšava implementaciju aplikacija širom svijeta, poboljšavajući latenciju i dostupnost. Ove karakteristike čine AWS nezamjenjivim za organizacije u procesu digitalne transformacije [2].

2.1. AWS servisi

Amazon Web Services predstavlja širok spektar usluga računarstva u oblaku koje se mogu grupisati u sedam osnovnih kategorija, omogućavajući korisnicima da izaberu tačno ono što im je potrebno. Neki od najznačajnijih servisa koji pripadaju osnovnim kategorijama uključuju Amazon Elastic Compute Cloud (EC2) za računarstvo u oblaku, Amazon S3 za skladištenje podataka, Amazon DynamoDB i Amazon RDS za baze podataka, Amazon VPC i Amazon CloudFront za mrežne usluge i isporuku sadržaja, Amazon Redshift i Amazon Athena za usluge analize podataka, Amazon SageMaker i Amazon Rekognition za mašinsko učenje, i AWS IAM i AWS Key Management Service za bezbjednost, identitet i usklađenost [3].

3. AWS KEY MANAGEMENT SERVICE

AWS Key Management Service je usluga koja omogućava korisnicima generisanje, upravljanje i kontrolu kriptografskih ključeva koji se koriste za enkripciju podataka na AWS platformi. Koristeći AWS KMS, korisnici mogu lako stvarati i upravljati ključevima za enkripciju podataka bez potrebe za upravljanjem fizičkim hardverom ili implementacijom složenih kriptografskih algoritama.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Srđan Vukmirović, red. prof.

Ova usluga pruža visoku sigurnost i pouzdanost, obezbeđujući enkripciju podataka tokom njihovog skladištenja i prenosa, što je ključno za zaštitu osjetljivih informacija u oblaku [4].

3.1. AWS KMS ključevi

KMS ključevi su ključni elementi za AWS Key Management Service, koji omogućavaju enkripciju i dekripciju podataka uz visok nivo bezbjednosti i fleksibilnost u generisanju dodatnih ključeva. Svaki KMS ključ sadrži metapodatke poput identifikatora, specifikacije, upotrebe i stanja ključa, pružajući korisnicima kontrolu i informacije o kriptografskim operacijama.

3.1.1. Osnovne vrste ključeva

Tri osnovne kategorije ključeva u AWS Key Management Service platformi su: AWS Managed key (ključevi koje upravlja AWS), Customer Managed key (ključevi koje upravlja korisnik), i AWS Owned key (ključevi u vlasništvu AWS-a).

3.1.2. Specijalne vrste ključeva

Pored osnovnih tipova ključeva, postoje i posebne vrste ključeva, koji uključuju i asimetrične ključeve, HMAC ključeve i multi-region ključeve.

4. IZBOR KARAKTERISTIKA KMS KLJUČEVA

U ovom poglavlju biće istražene različite karakteristike i opcije prilikom kreiranja ključeva za upravljanje kriptografskim operacijama pomoću AWS Key Management Service.

4.1. Odabir tipa KMS ključa

Izbor odgovarajuće vrste KMS ključa ključan je za bezbjednost podataka i ispravno funkcionisanje sistema. Simetrični ključevi su pogodni za većinu slučajeva, dok asimetrični omogućavaju šifrovanje izvan AWS-a. HMAC ključevi su korisni za integritet i autentičnost podataka. Važno je prilagoditi izbor ključa potrebama sistema za najviši nivo bezbjednosti i funkcionalnosti.

4.2. Odabir namjene KMS ključa

Pri odabiru namjene KMS ključa važno je razmotriti kako će se taj ključ koristiti, bilo za šifrovanje i dešifrovanje, potpisivanje i provjeru potpisa ili generisanje i provjeru HMAC oznaka. Svaki KMS ključ ima svoju specifičnu namjenu, što znači da će biti prilagođen samo jednoj vrsti operacija, što je važno kako bi se izbjeglo korišćenje istog ključa za više vrsta operacija, što može učiniti proizvod podložnijim napadima.

4.3. Odabir ključne specifikacije

Specifikacija ključa je kriptografska konfiguracija koja određuje karakteristike KMS ključa, uključujući tip ključa, kriptografski materijal i podržane algoritme, postavljajući osnove ključa koje kasnije nije moguće promijeniti. Pri odabiru specifikacije ključa važno je uzeti u obzir konkretan slučaj upotrebe i regulatorne zahtjeve, kao i mogućnost različitih cijena koje mogu biti povezane s različitim specifikacijama ključa.

5. UPRAVLJANJE KLJUČEVIMA

Upravljanje ključevima je temeljni aspekt osiguravanja povjerljivosti i sigurnosti podataka u današnjim cloud okruženjima, ključno za osiguravanje integriteta i

privatnosti podataka, bez obzira na vrstu informacija. AWS Key Management Service omogućuje organizacijama efikasno generisanje, upravljanje i korištenje kriptografskih ključeva za zaštitu [4].

5.1. Proces kreiranja ključeva

U okviru ovog odjeljka biće objašnjeni različiti načini za kreiranje kako simetričnih tako i asimetričnih tipova ključeva.

5.1.1. Proces kreiranja ključeva putem konzole

Korisnici mogu koristiti AWS Management Console za kreiranje simetričnih i asimetričnih ključeva. Proces obuhvata odabir tipa ključa (simetrični ili asimetrični), namjenu ključa (šifrovanje i/ili dešifrovanje, potpisivanje i/ili verifikacija), unos pseudonima ključa, dodavanje opcionalnih opisa i oznaka, te definisanje korisnika i uloga koje mogu upravljati ključem ili ga koristiti u kriptografskim operacijama.

5.1.2. Proces kreiranja ključeva kroz AWS Command Line Interface (CLI)

Korištenjem AWS CLI-a, korisnici mogu efikasno upravljati sigurnosnim ključevima putem programskog interfejsa. Ova metoda omogućava korisnicima da koriste operaciju CreateKey, koja generiše simetrične ključeve za šifriranje u odabranom AWS regionu ili asimetrične ključeve za potpisivanje i verifikaciju poruka.

5.1.3. Proces kreiranja ključeva putem AWS CloudFormation-a

Kreiranje AWS KMS resursa putem AWS CloudFormation-a omogućava korisnicima da efikasno modeliraju i postave KMS ključeve i pseudonime, oslobađajući ih potrebe za ručnim upravljanjem resursima i infrastrukturom. Ovaj proces obuhvata definisanje ključeva i njihovih atributa, uključujući tip ključa (simetrični ili asimetrični) i druge parametre, kroz tekstualne šablone u JSON ili YAML formatu.

5.1.4. Proces kreiranja ključeva korištenjem AWS KMS API-ja

Korisnici mogu koristiti AWS SDK za podržane programske jezike (Java, .NET, Python, Ruby, PHP ili JavaScript u Node.js.) za interakciju sa AWS KMS putem API-ja. Kroz API, korisnici mogu koristiti operaciju CreateKey za kreiranje novih ključeva u AWS KMS-u. Ovo omogućava širok spektar funkcionalnosti za interakciju sa AWS uslugama putem podržanih programskih jezika.

5.2. Alijasi

Alijasi u AWS Key Management Service su prijateljski nazivi koji se koriste za identifikaciju KMS ključeva umjesto korištenja drugih identifikatora. Njihova nezavisnost od samog KMS ključa znači da akcije koje se izvode na alijasu ne utiču direktno na povezani ključ. Osim olakšavanja prepoznavanja i referenciranja KMS ključeva u različitim situacijama, alijasi omogućavaju i kontrolu pristupa KMS ključevima, čime se olakšava upravljanje pravima pristupa.

Svaki alijas ima jedinstveni Amazon Resource Name (ARN) koji ga identifikuje unutar određenog računa i regije. Mogu se kreirati u različitim regijama, omogućavajući upotrebu istog koda u više regija, pri

čemu se automatski upućuju na povezani KMS ključ u svakoj regiji. Alijasima je moguće upravljati putem AWS KMS konzole, CreateAlias API-ja ili pomoću AWS CloudFormation šablona. AWS KMS API pruža potpunu kontrolu nad alijasima, omogućavajući operacije kao što su kreiranje, pregled, ažuriranje i brisanje alijasa u svakom nalogu i regiji, koristeći ugrađene metode koje su dio AWS SDK. Alijasi su osjetljivi na velika i mala slova i ne mogu se mijenjati nakon što su kreirani.

5.3. Pregled dostupnih KMS ključeva

Pregled dostupnih KMS ključeva je ključan za osiguranje sigurnosti i zaštite podataka u oblaku. Ovaj proces omogućava administratorima efikasno upravljanje ključevima i pristupom, te održavanje statusa sigurnosti i praćenje sigurnosnih parametara.

Pregled je omogućen putem AWS konzole i API-ja, pružajući detaljne informacije o ključevima i njihovim karakteristikama, kao i mogućnost filtriranja i sortiranja radi lakšeg upravljanja resursima. Korištenjem operacija poput ListKeys, DescribeKey, GetKeyPolicy, ListAliases i ListResourceTags putem API-ja, korisnici mogu detaljno upravljati i pratiti svoje KMS ključeve i povezane alijase.

5.4. Omogućavanje i onemogućavanje KMS ključeva

Omogućavanje i onemogućavanje AWS KMS ključeva je ključni dio upravljanja sigurnošću podataka u AWS KMS-u. Ključevi se podrazumijevano omogućavaju prilikom kreiranja, ali ih je moguće onemogućiti kako bi se privremeno isključili iz upotrebe, pružajući korisnicima mogućnost zadržavanja ključeva za buduću upotrebu. Međutim, ključevi koje upravlja AWS ne mogu se onemogućiti ili omogućiti jer su trajno omogućeni za korištenje od strane AWS usluga. Postoje dva načina za omogućavanje i onemogućavanje KMS ključeva: putem konzole i putem AWS KMS API-ja.

5.5. Automatska rotacija KMS ključeva

Automatska rotacija ključeva omogućava sigurnu zamjenu kriptografskog materijala ključeva bez potrebe za promjenom aplikacija ili AWS usluga. AWS KMS periodično generiše nove kriptografske materijale za ključeve koji su omogućeni za automatsku rotaciju, čuvajući pritom sve prethodne verzije.

Rotirani materijal ključa se ne briše dok se ne obriše sam KMS ključ. Rotacija ključeva se može pratiti putem Amazon CloudWatch-a i AWS CloudTrail-a, a podržana je samo za simetrične KMS ključeve koje generiše AWS KMS. Omogućavanje ove funkcije je opcionalno za KMS ključeve kojima upravlja korisnik, dok AWS KMS automatski rotira ključni materijal za KMS ključeve kojima sam upravlja. Ova funkcija donosi nekoliko prednosti, uključujući očuvanje svojstava ključa kao što su ID i ARN, kao i kontinuiranu upotrebu KMS ključeva u AWS uslugama.

5.6. Nadgledanje KMS ključeva

Nadgledanje ključeva u AWS KMS-u je ključno za osiguravanje njihove dostupnosti, stanja i optimalne upotrebe. To je važan korak u održavanju pouzdanosti, dostupnosti i performansi AWS rešenja.

Za tu svrhu koriste se alati poput AWS CloudTrail-a, Amazon CloudWatch-a i Amazon EventBridge-a.

5.7. Brisanje KMS ključeva

Proces brisanja AWS KMS ključeva ima trajne posledice, budući da se ne može povratiti i čini podatke nepristupačnim. Važno je pažljivo razmotriti brisanje ključa, uz procjenu rizika i eventualno isključivanje ključa umesto brisanja. AWS KMS ne briše ključeve osim ako korisnik izričito ne zakaže brisanje, što dodatno ističe važnost pažljivog upravljanja ključevima.

Periode čekanja prije brisanja ključeva pruža dodatnu bezbjednost i mogućnost povratka, a proces brisanja je isti za simetrične i asimetrične ključeve, ali kod asimetričnih ključeva zahtijeva pažljivo planiranje i nadzor radi očuvanja bezbjednosti podataka.

6. POLITIKE UPRAVLJANJA PRISTUPOM AWS KMS RESURSIMA

Upravljanje pristupom AWS Key Management Service resursima predstavlja ključni aspekt sigurnosne strategije unutar AWS okruženja. Različiti mehanizmi omogućavaju organizacijama da precizno kontrolišu ko može pristupiti ključnim resursima i na koji način [4].

6.1. Politika ključa

Politika ključa u AWS KMS predstavlja dokument koji precizira pravila pristupa i dozvole za korišćenje određenog KMS ključa. Svakom KMS ključu mora biti dodeljena tačno jedna politika ključa, koja detaljno definiše ko ima pristup ključu i na koji način. Politika ključa precizira dozvole za korisnike, grupe ili uloge koje imaju pristup ključu. Izjave u politici ključa definišu koje operacije su dozvoljene ili zabranjene za određene subjekte. Politika ključa se primenjuje samo na KMS ključeve unutar iste AWS regije i može biti ažurirana u bilo kom trenutku kako bi se promijenile dozvole ili pravila pristupa. Pri definisanju politike za AWS KMS, ključni dokument politike treba da sadrži određene elemente, uključujući Version, Statement, Sid, Effect, Principal, Action, Resource i Condition.

6.2. IAM politika

IAM politike su ključni alat u upravljanju pristupom AWS resursima, omogućavajući preciznu kontrolu nad dozvolama i ograničenjima za različite korisnike, grupe i uloge. Ove politike definišu koje akcije mogu biti izvršene nad određenim AWS resursima, pružajući sigurnosni okvir za upravljanje identitetima u AWS oblaku. Kroz kombinaciju IAM politika sa drugim AWS alatima za upravljanje pristupom, administratori mogu prilagoditi pristup različitim AWS resursima u skladu sa potrebama korisnika, osiguravajući time bezbjednost i usklađenost sa regulativama.

Putem IAM konzole ili AWS API-ja, administratori mogu dodeljivati, uređivati i uklanjati IAM politike kako bi prilagodili pristup resursima u skladu sa zahtjevima poslovanja i bezbjednosti.

6.2. Grantovi

Grantovi u AWS KMS-u omogućavaju privremeni pristup određenim KMS ključevima za kriptografske operacije poput šifrovanja i dešifrovanja podataka. Oni se koriste za kontrolu pristupa podacima zaštićenim KMS ključevima, često u AWS uslugama koje zahtijevaju šifrovanje podataka. Prilikom korišćenja grantova, važno

je razumjeti koncepte kao što su ograničenja granata, jedinstveni identifikatori granata i dozvoljene operacije. Kreiranje granata obavlja se putem operacije CreateGrant, gdje se navode ključ KMS-a, principi primaoaca, dozvoljene operacije i eventualna ograničenja.

Važno je izbeći dupliranje granata i koristiti neutralne nazive kako bi se spriječilo otkrivanje osjetljivih informacija.

7. INTEGRACIJA I UPOTREBA AWS KMS U RAZLIČITIM AWS USLUGAMA

Integracija i korišćenje AWS Key Management Service u različitim AWS uslugama ključni su za sigurnost u oblaku. AWS KMS pruža napredne funkcionalnosti enkripcije i omogućava zaštitu podataka u oblaku. Mnoge AWS usluge, poput AWS CloudTrail-a, Amazon DynamoDB-a, Amazon S3 i drugih, koriste AWS KMS za šifrovanje podataka i upravljanje ključevima. Ova integracija osigurava bezbjednost i poverljivost informacija korisnika u različitim kontekstima.

7.1. AWS CloudTrail

AWS CloudTrail je servis koji omogućava snimanje događaja i aktivnosti na AWS nalogu radi praćenja, analize i bezbjednosnih razloga, uz korišćenje AWS KMS-a za sigurno čuvanje podataka [5].

7.2. Amazon Simple Storage Service

Amazon Simple Storage Service (S3) je usluga za čuvanje podataka u oblaku koja omogućava korisnicima da skladište i koriste podatke iz bilo kog mjesta na internetu, pružajući pristup podacima putem standardnih veb protokola. AWS KMS se koristi za šifrovanje podataka prilikom čuvanja u Amazon S3, omogućavajući visok nivo bezbjednosti [6].

7.3. Amazon WorkMail

Amazon WorkMail je sigurna i kontrolisana usluga za poslovnu e-poštu, koja pruža podršku za različite klijente e-pošte na desktopu i mobilnim uređajima. Za zaštitu ključeva za šifrovanje koristi se AWS KMS, koji omogućava kontrolu nad ključevima i dodatni sloj bezbjednosti [7].

7.4. Amazon Elastic Block Store

Amazon Elastic Block Store (Amazon EBS) omogućuje korisnicima skladištenje i upravljanje podacima na nivou blokova, pružajući dodatni sloj zaštite uz korišćenje AWS KMS-a za enkripciju podataka [8].

7.5. Amazon DynamoDB

Amazon DynamoDB je potpuno upravljana NoSQL baza podataka koja se skalira prema potrebama korisnika, integrišući AWS KMS radi enkripcije podataka na strani servera [9].

7.6. AWS Secrets Manager

AWS Secrets Manager je usluga za čuvanje tajnih informacija koja koristi AWS KMS za šifrovanje i dešifrovanje tih informacija, namijenjena čuvanju osjetljivih podataka, kao što su lozinke [10].

8. ZAKLJUČAK

Kroz analizu različitih aspekata AWS Key Management Service, od vrsta ključeva do politika upravljanja pristupom, istaknuta je važnost ove usluge u osiguravanju pouzdane zaštite podataka u kladu okruženju.

Integracija KMS-a sa različitim AWS uslugama demonstrira širok spektar primjena ove usluge.

Dalje unapređenje uključuje detaljnu analizu performansi KMS-a, primjere primjene, komparativnu analizu sa konkurentskim rješenjima i istraživanje relevantnih sigurnosnih propisa i standarda. Takva dodatna istraživanja doprinose dubljem razumijevanju prednosti, nedostataka i implikacija korišćenja AWS KMS-a u praksi.

9. LITERATURA

- [1] https://en.wikipedia.org/wiki/Cloud_computing (pistupljeno u martu 2024.)
- [2] https://en.wikipedia.org/wiki/Amazon_Web_Services (pistupljeno u martu 2024.)
- [3] <https://aws.amazon.com/products/> (pistupljeno u martu 2024.)
- [4] <https://docs.aws.amazon.com/kms/latest/developerguide/overview.html> (pistupljeno u martu 2024.)
- [5] <https://aws.amazon.com/cloudtrail/> (pistupljeno u martu 2024.)
- [6] <https://aws.amazon.com/s3/> (pistupljeno u martu 2024.)
- [7] <https://aws.amazon.com/workmail/> (pistupljeno u martu 2024.)
- [8] <https://aws.amazon.com/ebs/> (pistupljeno u martu 2024.)
- [9] <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html> (pistupljeno u martu 2024.)
- [10] <https://aws.amazon.com/secrets-manager/> (pistupljeno u martu 2024.)

Kratka biografija:



Plema Lažetić rođena je 17. aprila 1999. godine u Trebinju. Osnovnu školu "Sveti Sava" završila je u Avtovcu, dok je gimnaziju, opšti smjer, završila u srednjoškolskom centru "Pero Slijepčević" u Gacku, kao đak generacije. Školske 2018/2019 godine upisuje Fakultet tehničkih nauka u Novom Sadu, na studijski program Primenjeno softversko inženjerstvo. Osnovne akademske studije završila je u predviđenom roku, te školske 2022/2023 upisuje master studije, takođe, na studijskom programu Primenjeno softversko inženjerstvo. kontakt: lazeticplema@gmail.com

КОМПАРАТИВНА АНАЛИЗА ПЕРФОРМАНСИ РЕЛАЦИОНИХ И НЕРЕЛАЦИОНИХ СИСТЕМА БАЗА ПОДАТАКА COMPARATIVE PERFORMANCE ANALYSIS OF RELATIONAL AND NON- RELATIONAL DATABASE SYSTEMS

Миљана Симић, Факултет техничких наука, Нови Сад

**Област – ЕЛЕКТРОТЕХНИЧКО И
РАЧУНАРСКО ИНЖЕЊЕРСТВО**

Кратак садржај – У овом раду описани су основни концепти система релационих и нерелационих база података и перформанси база података. Поређење перформанси база података је извршено у погледу просечног времена извршења упита ка бази података. Описан је коришћени скуп података, изабране релациона (MySQL) и нерелациона (Neo4j) база података су креиране и испуњене подацима. Посебна пажња приликом креирања упита посвећена је класификацији упита у различите категорије ради лакшег поређења перформанси. Упити су анализирани и оптимизовани коришћењем одговарајућих техника. За развој и управљање изабраним базама података коришћени су MySQL и Neo4j алати, а за креирање везе ка базама података и извршавање упита коришћен је програмски језик Python.

Кључне речи: системи база података, релациона база података, нерелациона база података, перформансе, упит

Abstract – This thesis describes the fundamental concepts of relational and non-relational database systems and their performance. The comparison of database performance was conducted in terms of the average query execution time. The selected dataset was described, and relational (MySQL) and non-relational (Neo4j) databases were created and populated with data. Special attention is given to the classification of queries into different categories for easier performance comparison. Queries are analyzed and optimized using appropriate techniques. MySQL Workbench and Neo4j Desktop tools are used for the development and management of the selected databases. Python is used to establish connections with databases and execute queries.

Keywords: database systems, relational database, non-relational database, performance, query

1. УВОД

У савременом свету, информационе технологије се константно развијају и постају неизоставне у свакодневном животу и пословању, те је правилно управљање подацима од великог значаја. Базе података представљају темељну инфраструктуру за чување,

организацију и управљање подацима, а различите врсте база података нуде различите приступе решавању изазова.

У почецима развоја база података, SQL (енгл. *Structured query language*) базе података су доминирале информационим системима и биле су најчешћа опција за чување и управљање подацима. Међутим, у задње време, свет информационих технологија доживео је драстичну трансформацију, што је довело до експлозије великих количина података и појаве полу-структурираних података. NoSQL (енгл. *Not Only SQL*) базе података су стекле огромну популарност и све чешће се примењују.

Циљ истраживања јесте дубље разумевање карактеристика и перформанси SQL и NoSQL база података. У фокусу истраживања јесте анализа и упоређивање перформанси ових двеју база података у смислу времена извршавања упита над истим скупом података. Рад се користи велику количину стварних података прикупљених са Shopify платформе. Циљ је да се пројектним приступом развију релациона (MySQL) и нерелациона (Neo4j) база података, испуне подацима са Shopify платформе, изврше различити упити и упореде перформансе ових база података у одређеним сценаријима.

2. ТЕОРИЈСКЕ ОСНОВЕ И КОНЦЕПТИ БАЗА ПОДАТАКА

База података представља темељ сваког информационог система. Без обзира на врсту апликације или област примене, разумевање основних концепата и принципа база података је кључно за ефикасно чување и управљање подацима.

2.1 Релационе и нерелационе базе података

Релациона база података представља врсту базе података која се заснива на релационом моделу података. За разлику од неких комплексних модела, релациони модел је једноставан и интуитиван начин за представљање података. Релационе базе података обезбеђују ACID својства трансакције (атомичност, конзистентност, изолација и трајност), те су трансакције поуздане и обезбеђују конзистентност података [1].

Нерелационе базе података пружају различите опције за организацију података. Нудећи разнолике структуре података, могу примењивати у различите сврхе, као што су анализа података, управљање великим ко-

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији је ментор био др Милан Челиковић, доцент.

личинама података, друштvene мреже, развој мобилних апликација и друге. У контексту дистрибуиране обраде значајна је CAP теорема у оквиру које се тврди да дистрибуирани системи у датом тренутку времена могу гарантовати само два од следећа три својства: конзистентност, доступност и толеранција на партиционисање. NoSQL базе података често фаворизују доступност и толеранцију на партиционисање у односу на конзистентност у сваком тренутку [2].

Нерелационе базе података најчешће се деле у четири групе: документ, кључ-вредност, графовске и колонске.

Граф база података користи структуре графа за представљање односа између података. Елементи података су дефинисани као чворови, везе и својства [3]. Чворови представљају ентитете. Веза (потег у графу) дефинише однос између чворова, док својства представљају информације које се односе на чворове или везе.

2.2 Перформансе базе података

Перформансе базе података се односе на способност базе података да ефикасно обрађује упите и трансакције, као и да пружа висок ниво одзива корисницима.

База података је неизоставан део било ког система који складишти податке и веома је битно да ради коректно, буде доступна, омогућава брзе операције, конзистентност података и буде отпорна на отказе. Чак и релативно мањи проблеми са перформансама базе података могу негативно утицати на функционисање система. и на корисничко искуство.

Управљање и надгледање перформанси осигурава да база података није уско грло (енгл. *bottleneck*) или није та која узрокује проблеме, и да ради што је могуће ефикасније и оптималније [4]. Многи проблеми са перформансама базе података повезани су са перформансама упита. Дobar дизајн упита може смањити оптерећење базе података. Поред упита, индекси су још један кључни елемент базе података. Правилно индексирање може побољшати перформансе базе података јер повећава ефикасност процеса прибављања података. Коришћење кеша, оптимизација шеме базе података и хардвера су још неки од фактора који могу допринети бољим перформансама.

3. ПРЕГЛЕД ПОСТОЈЕЋЕ РЕЛЕВАНТНЕ ЛИТЕРАТУРЕ

Истраживањем и анализом постојећих научних радова који се бави сличним проблемима и концептима, издвојена су следећа три као најрелевантнија.

Рад „Поређење перформанси SQL и NoSQL система за управљање базама података (скраћено СУБП) за конфигурацију 5G базе радио станице“ [5] спроводи истраживање са циљем побољшања постојећег СУБП-а. Циљ је био имплементација и упоређивање две базе података, како би замениле све тренутно сачуване податке у датотекама. Упоређивање укључује време за креирање, читање и ажурирање података у бази података. Као представник база података коришћене

су PostgreSQL и MongoDB базе података. Креиране базе су упоређиване у погледу времена одговора на уобичајене упите, као и у тестирању оптерећења са очекиваним оптерећењем на базу података. Резултати су показали да обе базе имају своје предности и оба решења испуњавају своју сврху. С обзиром на познавање система и начина на који ће систем бити коришћен, MongoDB је изабран као бољи СУБП.

У оквиру истраживачког рада на тему „Перформансе граф и релационе базе података у сложеним упитима“ [6] коришћене су MariaDB и две верзије MySQL базе података, док је Neo4j изабран да буде представник граф СУБП-а. Поред анализе сложених упита, посвећена је пажња факторима који се односе на ефикасност. Резултати граф СУБП-а превазишли су релациони СУБП са структурално једноставним скуповима података и једноставним упитима. Међутим, са сложенијим упитима, нови системи за управљање релационим базама података надмашили су Neo4j. Резултати овог истраживања показују како су релационе базе података и даље добра алтернатива када је реч о перформансама. Закључак рада је да, иако је Neo4j ефикаснији за једноставне упите, MariaDB је доста ефикаснија када се ради о сложеним упитима.

Рад на тему „Поређење перформанси граф и релационе базе података“ [7] је као мотивацију имао поређење релационе и граф базе података и испитивање примене ових база података у различитим случајевима коришћења. Поређење је извршено извршавањем упита над истим подацима у оба шеме базе података. Резултати су показали да граф база података надмашује релациону базу података до 146 пута када су у питању сложени упити и велики скупови података.

4. ИМПЛЕМЕНТАЦИЈА ИСТРАЖИВАЊА

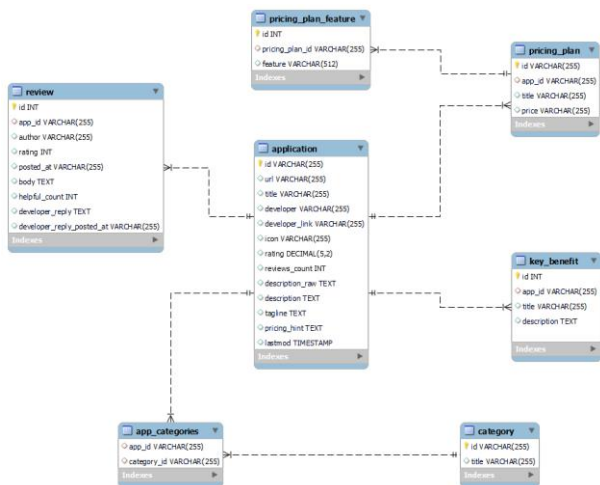
Приступ истраживању је „одоздо према горе“ (енгл. *bottom-up*), што значи да је прво пронађен одговарајући скуп података који ће служити као основа истраживања, а да је потом извршен избор одговарајућих релационих и нерелационих система за управљање базом података.

4.1 Скуп података и шеме база података

Извор скупа података је Shopify App Store [8], популарна платформа за интернет трговину, а сам скуп података доступан је на Kaggle веб страници. Овај скуп података садржи информације о различитим апликацијама које су креирали програмери и које су доступне на платформи, као и о рецензијама које су објавили купци.

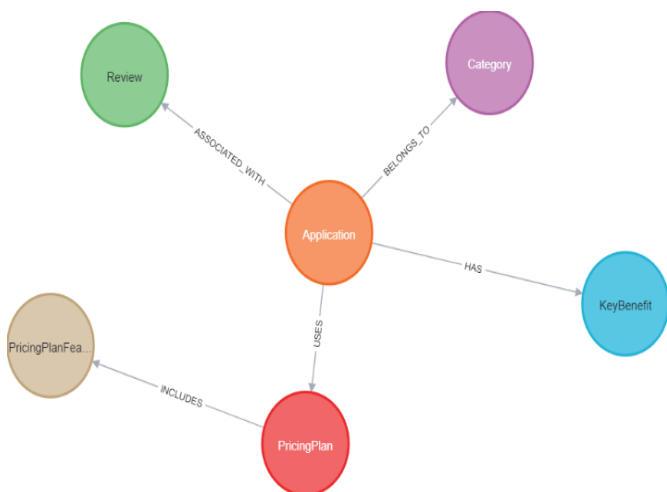
Детаљним истраживањем великог броја скупова података, одабран је овај скуп података у циљу спровођења анализе и поређења перформанси. Одабир скупа података потиче из жеље да се обухвати велики обим података, везе међу подацима и разноврсне информације. У процесу припреме података извршено је пречишћавање података, уклањање неправилности и разлика у кодирању вредности одређених колона како би подаци били конзистентни и употребљиви.

Као релациони СУБП, изабран је *MySQL* сервер базе података је веома брз, поуздан, скалабилан и лак за коришћење. Слика 4.1 приказује визуалну репрезентацију шеме базе података.



Слика 4.1 Шема релационе базе података

Што се тиче нерелационе базе података, изабрана је граф база података, *Neo4j*. *Neo4j* граф база података омогућава детаљно истраживање веза између апликација, категорија, бенефита, рецензија и других ентитета из скупа података. Слика 4.2 приказује визуелну репрезентацију генерисане шеме граф базе података



Слика 4.2 Шема граф базе података

4.2 Извршавање упита

Метрика евалуације перформанси била је просечно време одзива за сваки од креираних упита. Перформансе обе базе података оцењиване су упоређивањем времена одзива. Просечно време је мерено покретањем сваког упита у одређеном броју итерација (50).

За комуникацију са базама података коришћен је *Python* програмски језик. Креиране су *Python* скрипте које се повезују на базу података и шаљу упите ка истој. Прва три захтева ка бази су игнорисана како би се систем за управљање базом података правилно поставио и стабилизовао везу, што значи да је средња

вредност сваког упита израчуната на основу 47 захтева.

Упити су подељени у три категорије:

1. Једноставна селекција: Упити који припадају овој категорији су једноставни упити. Ови упити фокусирани су на прибављање података из једне табеле или типа чвора на основу одређених атрибута.
2. Операције спајања: Упити који припадају овој категорији подразумевају спајање две или више табела, односно чворова када се ради о граф бази података.
3. Агрегација података: Упити који припадају овој категорији обухватају агрегацију података, као што су бројање, сумирање, рачунање просека, максималних и минималних вредности и слично, из више извора података.

Резултати покренутих упита сачувани су у CSV датотеке које су касније коришћене за креирање дијаграма и табела са резултатима, ради евалуације и анализе резултата.

4.3. Анализа стратегије упита

Анализа плана извршавања упита доприноси бољем разумевању перформанси упита. *MySQL* пружа веома добру документацију и подршку за оптимизацију и анализу упита. *MySQL* оптимизатор разматра многе технике за ефикасно извршење претрага у *SQL* упиту. Скуп операција које оптимизатор бира како би остварио најефикаснији упит назива се „план извршења упита“, познат као *EXPLAIN* план. Потребно је препознати аспекте *EXPLAIN* плана који указују на то да је упит добро оптимизован, али и искористити технике индексирања ради побољшања плана када се неке од операција не извршавају ефикасно. *EXPLAIN* наредба је извршена над сваким од креираних упита ради побољшања и бољег разумевања перформанси.

Neo4j упити се извршавају према одређеном плану извршења. Анализа перформанси упита *Neo4j* базе података изводи се уз помоћ *EXPLAIN* и *PROFILE* команди. Команда *EXPLAIN* се користи за приказивање плана извршења упита, без извршавања упита. Ова команда пружа информације о томе како ће упит бити извршен, укључујући индексе који ће бити коришћени, које операције ће бити извршене и проценен број редова који ће бити обрађен. Команда *PROFILE* извршава упит и пружа детаљније информације од *EXPLAIN* команде.

5. РЕЗУЛТАТИ И ДИСКУСИЈА

Након креираних упита, извршена је анализа и праћење резултата и упити су оптимизовани колико је то могуће.

Резултати упита сачувани су у CSV датотекама. Поред графичког приказа резултата извршаваних упита, креирана је и табела са резултатима. Табела је ту ради лакшег и прегледнијег сумирања резултата сваког од упита, као и лакшег поређења просечних времена по категоријама упита.

У наставку се налази табела 5.1 која садржи просечно време извршења упита, по категоријама.

Табела 5.1 Резултати извршења упита

Категорија	Упит	MySQL	Neo4j
Једноставна селекција	Упит 1	26.81ms	38.24ms
	Упит 2	0.58ms	4.14ms
	Упит 3	90.21ms	23.7ms
Операције спајања	Упит 4	0.93ms	4.35ms
	Упит 5	18.32ms	38.5ms
	Упит 6	11.07ms	52.95ms
Агрегација података	Упит 7	20.45ms	68.83ms
	Упит 8	171.49ms	58.63ms
	Упит 9	120.75ms	24.31ms

Кроз евалуацију перформанси, фокус је на три врсте упита – једноставна селекција, операције спајања и агрегација података.

Анализом графика и резултата, простији упити, који селектују податке само једног типа, дали су боље резултате код релационе базе података (и до 7 пута мање просечно време извршења). Са друге стране, упит који садржи под упите у себи се показао боље код граф базе података (3.8 пута брже). Узимајући у обзир резултате, упити спајања дали су убедљиво боље резултате код релационе базе података. На основу резултата агрегационих упита, граф база података се показала доста боље, јер овакви упити захтевају много више времена за извршење над релационим базом података.

6. ЗАКЉУЧАК

Циљ рада је био дубље разумевање карактеристика и перформанси релационих и нерелационих база података и издавања предности и ограничења сваке од ових база података. Истраживање се бавило анализом и упоређивањем перформанси ових двеју база података у смислу времена извршавања упита над истим скупом података. Евалуација је била фокусирана на време одговора, које је приказано просечним временом одговора упита који су покретани у одређеном броју итерација.

Упити који су извршавани над пројектованим базама података класификовани су у три категорије. Резултати извршених упита су показали да релациона података у неким случајевима надмашује граф базу података и до седам пута у одређеним упитима. Ови резултати указују на значај индексирања у релационим базама података. Међутим, у комплекснијим упитима који агрегирају податке, перформансе граф базе података надмашиле су релациону базу података, просечно време извршења упита над граф базом података било је до четири пута мање.

Закључак рада је да се на основу тестирања изабраног скупа података и упита не може опште закључити који од система база података има најбољу ефикасност упита. Приликом избора система база података за одређени домен апликације, потребно је пажљиво и унапред анализирати потребе за будућим упитима, те је често веома корисно комбиновати и интегрисати обе врсте база података у складу са потребама.

Постоји неколико идеја за могућа унапређења и проширења истраживања овог мастер рада. Једна од идеја јесте увести још неке од представника популарних система база података и поредити перформансе између више система. Било би корисно проширити скуп података, користећи још неки од извора података, ради додавања нових ентитета и веза, како би се креирали комплекснији упити. Такође, поређење перформанси је могуће урадити симулацијом стварног оптерећења базе података користећи неки од јавно доступних алата и на тај начин тестирати како ће се понашати сваки од система.

7. ЛИТЕРАТУРА

- [1] Oracle, Database topics, <https://www.oracle.com/database/what-is-database/>
- [2] Couchbase, Why NoSQL, <https://www.couchbase.com/resources/why-nosql/>
- [3] Wikipedia, Graph database, https://en.wikipedia.org/wiki/Graph_database
- [4] Oracle, Database Performance Tuning Guide
- [5] Alexandra Goltsis, A Performance Comparison of SQL and NoSQL Database Management Systems for 5G Radio Base Station Configuration, Linköping University, 2022, 57 p, <https://www.researchgate.net/>
- [6] Petri Kotiranta, Marko Junkkari, Jyrki Nummenmaa: Performance of Graph and Relational Databases in Complex Queries, Tampere University, 2022, 16 p, <https://www.researchgate.net/>
- [7] Mit Jain, Ashish Khanchandani, Cajetan Rodrigues: Performance Comparison of Graph Database and Relational Database, Computer Science Department San Jose State University, 2023, 19 p, <https://www.researchgate.net/>
- [8] Скуп података Shopify app store, <https://www.kaggle.com/datasets/username3/shopify-app-store>

Кратка биографија:



Миљана Симић рођена је 22. септембра 1999. године у Нишу. Електронски факултет у Нишу, студијски програм електротехника и рачунарство уписала је 2018. године. Дипломирала је 2022. године и исте године уписала мастер академске студије на Факултету техничких наука у Новом Саду.

**TEHNIČKI ASPEKTI PRIKLJUČENJA FOTONAPONSKIH ELEKTRANA NA
DISTRIBUTIVNI SISTEM REPUBLIKE SRBIJE****TECHNICAL ASPECTS OF PHOTOVOLTAIC POWER PLANTS CONNECTED TO THE
DISTRIBUTION SYSTEM OF REPUBLIC OF SERBIA**

Aleksandar Pavić, Boris Dumnić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu su definisani tehnički uslovi za priključenje fotonaponskih elektrana na distributivni sistem Republike Srbije. Upotrebom softverskog alata ETAP, modelovana je fotonaponska elektrana snage 990 kW i priključena na 20 kV distributivnu mrežu. Na osnovu dobijenih rezultata, sprovedena je analiza tehničkih aspekata priključenja fotonaponske elektrane na distributivni sistem.

Ključne reči: Obnovljivi izvori, Fotonaponska elektrana, Tehnički uslovi, Distributivni sistem

Abstract – This paper describes the conditions for connecting photovoltaic power plants to the distribution system of Republic of Serbia. A 990 kW photovoltaic power plant has been modeled using the ETAP software tool and integrated into a 20 kV distribution grid. Based on the obtained results, an analysis of the technical aspects of integrating the photovoltaic power plant into the distribution system was conducted.

Keywords: Renewable energy, Photovoltaic power plant, Technical aspects, Distribution system

1. UVOD

Emisije ugljen-dioksida (CO₂) koje nastaju sagorevanjem fosilnih goriva značajno su podigle koncentraciju CO₂ u zemljinoj atmosferi. Zahvaljujući tome, dolazi do pojačavanja efekta staklene bašte koji izaziva značajne klimatske promene sa negativnim uticajima na proizvodnju hrane, vodosnabdevanje i društvo. Upotreba nuklearne energije je ograničena finansijama kao i pitanjima bezbednosti. Sva ova ograničenja izazvala su veliku potražnju za obnovljivim izvorima energije [1].

Prema IEA (eng. International energy agency) kapaciteti obnovljivih izvora energije u 2023. godine su se povećali za oko 510 GW, što predstavlja najbržu stopu rasta u prethodne dve decenije. Povećanje ovih kapaciteta dostiglo je rekordne vrednosti u Evropi, SAD i Brazilu, međutim Kina je imala najznačajniji rast, puštajući u rad onoliko solarnih fotonaponskih sistema koliko celi svet godinu dana ranije. Na globalnom nivou, solarni fotonaponski sistemi su činili tri četvrtine od ukupnog povećanja kapaciteta obnovljivih izvora energije [2].

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Boris Dumnić, red. prof.

U periodu od 2023. do 2028. godine prema predviđanjima IEA, očekuje se da će se kapaciteti obnovljivih izvora energije povećati za oko 3700 GW, što će biti više nego što je instalirano otkako je prva elektrana na obnovljivu energiju puštena u rad, pri čemu će solarni fotonaponski sistemi i vetar činiti oko 95% globalne ekspanzije obnovljivih izvora energije [2].

Takođe, u ovom periodu očekuje se sledeće [2]:

- U 2024. godini solarni fotonaponski sistemi i vetar generišu više električne energije od hidroenergije.
- U 2025. godini obnovljivi izvori energije premašuju proizvodnju električne energije dobijenu iz uglja.
- U 2025. i 2026. godini, solarni fotonaponski sistemi i vetar premašuju proizvodnju električne energije dobijenu iz nuklearnih elektrana.
- U 2028. godini obnovljivi izvori energije čine preko 42% globalne proizvodnje električne energije.

**2. USLOVI ZA PRIKLJUČENJE ELEKTRANA NA
DISTRIBUTIVNI SISTEM**

Tehnički uslovi za priključenje elektrane na distributivni sistem Republike Srbije, definisani su i preuzeti iz dokumenta pod nazivom “Pravila o radu distributivnog sistema” [3].

Tehnički uslovi za priključenje elektrana treba da omoguće normalan pogon DS uz nenarušavanje pouzdanosti isporuke i kvaliteta električne energije drugim korisnicima DS [3].

Za priključenje i bezbedan paralelan rad elektrane sa DS, elektrana mora da zadovolji sledeće kriterijume [3]:

- Kriterijum maksimalno dozvoljene snage generatora u elektrani,
- Kriterijum dozvoljenih vrednosti napona u stacionarnom režimu,
- Kriterijum dozvoljenog strujnog opterećenja elemenata distributivne mreže,
- Kriterijum snage kratkog spoja,
- Kriterijum flikera,
- Kriterijum dozvoljenih struja viših harmonika i interharmonika.

**3. SOFTVERSKI ALATI ZA SIMULACIJU
AKTIVNIH ENERGETSKIH MREŽA**

Neki od najčešće korišćenih softvera za simulaciju aktivnih energetskih mreža kao i za proveru ispunjenosti

uslova priključenja distribuiranih energetske resursa na DS jesu:

- DigSILENT,
- PSCAD,
- PSS@SINCAL,
- ETAP,
- NEPLAN,
- CYME i dr.

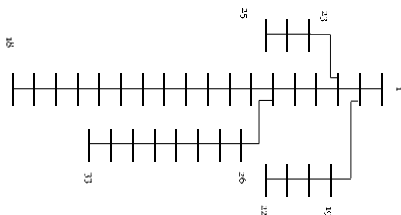
U ovom radu za sve simulacije korišten je softverski paket ETAP. Ovaj softverski paket nudi mogućnost analize, planiranja i optimizacije distributivne mreže, na progresivnoj geoprostornoj platformi za simulaciju, analizu, i optimizaciju performansi distributivnih pametnih mreža [4].

4. ANALIZA TEHNIČKIH ZAHTEVA ZA PRIKLJUČENJE FOTONAPONSKE ELEKTRANE NA DISTRIBUTIVNI SISTEM

U ovom radu predstavljen je konkretan model fotonaponske elektrane koja se priključuje na distributivnu mrežu naponskog nivoa 20 kV.

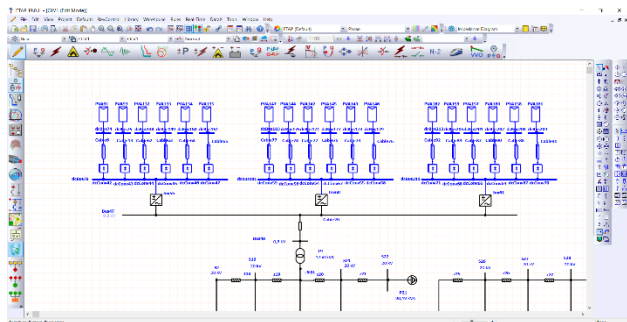
4.1. Opis modela distributivne mreže i fotonaponske elektrane

Distributivna mreža je kreirana po uzoru na IEEE 33 distributivnu mrežu. Naponski nivo modifikovane IEEE 33 distributivne mreže iznosi 20 kV.



Slika 4.1.1. IEEE 33 distributivna mreža [5].

Na slici 4.1.2 prikazan je model fotonaponske elektrane priključene na distributivnu mrežu naponskog nivoa 20 kV u ETAP-u.



Slika 4.1.2. Model fotonaponske elektrane u ETAP-u.

Fotonaponski panel kreiran je po uzoru na fotonaponski panel iz serije Eco Line HJT GG M132 kompanije "LUXOR" [6]. Fotonaponska elektrana je realizovana tako da se fotonaponski nizovi priključuju u šest nezavisnih MPPT (eng. *Maximum power point tracking*) ulaza na svakom invertoru. U okviru elektrane nalaze se tri invertora istih karakteristika. Fotonaponski nizovi su projektovani tako da sadrže 25 redno vezanih panela radi

dobijanja željenog napona od 1075 V. Na prvih pet MPPT ulaza svakog invertora paralelno se vezuju po 4 stringa (koji sadrže 25 redno vezanih panela) što ukupno čini 100 panela po jednom MPPT ulazu. Maksimalna snaga ovih MPPT ulaza iznosi 70 kWp, dok struja na DC strani u ovih pet MPPT ulaza iznosi 65 A. Na šesti MPPT ulaz paralelno se vezuju 2 stringa, koji sadrže 25 redno vezanih panela, što ukupno čini 50 panela. Snaga i struja na ovom MPPT ulazu iznosi 35 kWp i 32,56 A, respektivno. Takođe, potrebno je napomenuti da maksimalna snaga svih panela (snaga fotonaponskog polja cele elektrane) iznosi 1,155 MW. Izlazna snaga elektrane odnosno snaga prema distributivnoj elektroenergetskoj mreži u ovom primeru je 990 kW.

Invertor je kreiran po uzoru na invertor SUN2000-330KTL-H1 kompanije „Huawei“ [7]. U okviru elektrane nalaze se tri invertora istih karakteristika. U ovom modelu, izlazna snaga invertora definisana je na 330 kW, izlazni napon na 0,8 kV, a izlazna struja ima vrednost 238,2 A.

4.2. Provera kriterijuma maksimalno dozvoljene snage

Prema ovom kriterijumu, promena napona na mestu priključenja elektrane na distributivni sistem, ne sme preći vrednost od 2% na SN [3].

Na osnovu proračuna tokova snaga u distributivnoj mreži (elektrana još uvek nije priključena), primećuje se da na mestu gde je planirano priključenje fotonaponske elektrane (sabirnica S20) napon ima vrednost 98,12% nominalnog napona.

Priključenje fotonaponske elektrane na distributivni sistem je realizovano u sabirnici S20. Na osnovu proračuna tokova snaga, nakon priključenja fotonaponske elektrane primećuje se da se napon povećao za 0,41% i iznosi 98,53% nominalnog napona.

Na osnovu prethodnih proračuna, može se zaključiti da promena napona od 2% na mestu priključenja, nije prekoračena.

Maksimalno dozvoljena prividna snaga generatora u elektrani S_{ngm} u [MVA], za priključenje elektrane na SN, izračunava se prema sledećem izrazu [3]:

$$S_{ngm} = \frac{S_{ks}}{50 * k} \quad (4.2.1)$$

Snaga trofaznog kratkog spoja u tački priključenja na distributivni sistem, za naponski nivo 20 kV je tipizirana i iznosi 500 MVA. Za fotonaponske elektrane količnik struje uključenja i naznačene struje generatora k ima vrednost 1 ($k=1$) [4]. Nakon uvrštavanja ovih vrednosti u izraz (4.2.1) dobija se:

$$S_{ngm} = \frac{S_{ks}}{50 * k} = \frac{500}{50 * 1} = 10 \text{ MVA}$$

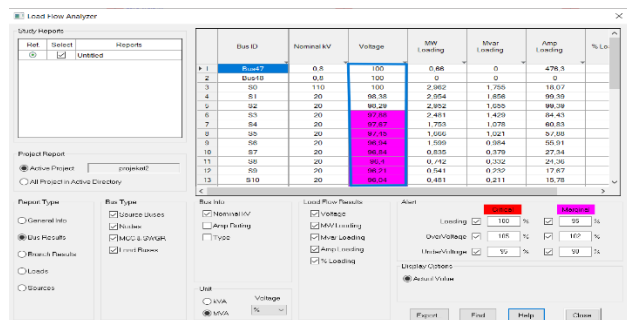
Prividna snaga generatora razmatrane elektrane je manja od 10 MVA, pa se može zaključiti da je kriterijum maksimalno dozvoljene snage zadovoljen.

4.3. Provera kriterijuma dozvoljenih vrednosti napona u stacionarnom režimu

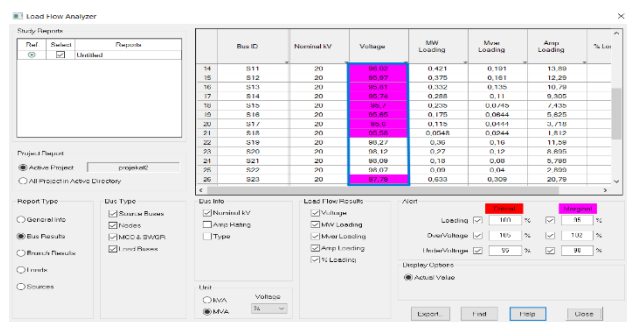
Kriterijum dozvoljenih vrednosti napona u stacionarnom režimu je zadovoljen ukoliko je, u okviru normalnog

pogona distributivnog sistema, promena napona Δu_m manja od 5% u bilo kojoj tački distributivnog sistema. Za proveru ovog kriterijuma je potrebno izvršiti proračune tokova snaga za minimalno i maksimalno opterećenje distributivnog sistema. Kod elektrana koje za proizvodnju električne energije koriste energiju sunca (solarne elektrane) za minimalno i maksimalno opterećenje distributivnog sistema usvaja se minimalno i maksimalno opterećenje u periodu u kojem je moguća proizvodnja ove vrste elektrana (pri prisustvu sunčeve svetlosti) [3].

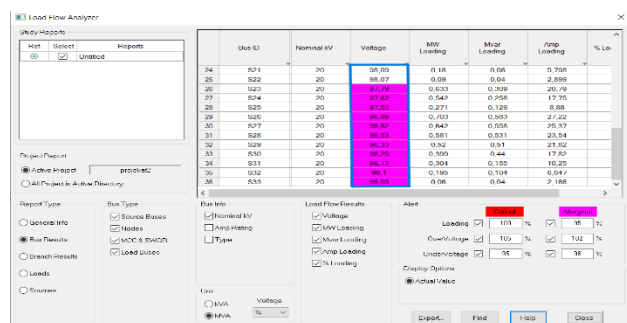
Na slikama 4.3.1, 4.3.2 i 4.3.3 prikazane su naponske prilike distributivnog sistema, pre priključenja FN elektrane.



Slika 4.3.1. Naponske prilike u DS pre priključenja.



Slika 4.3.2. Naponske prilike u DS pre priključenja.



Slika 4.3.3. Naponske prilike u DS pre priključenja.

Analizom rezultata dobijenih na osnovu proračuna tokova snaga, nakon priključenja fotonaponske elektrane, za uslove rada koji podrazumevaju rad elektrane sa nominalnom snagom, može se zaključiti da se napon povećao u svakoj tački distributivnog sistema, pri čemu u bilo kojoj tački distributivnog sistema promena napona nije prekoračila limit od 5%.

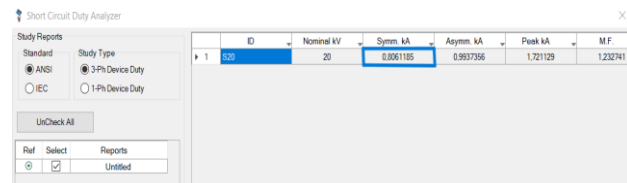
Uzimajući u obzir vrednost minimalnog napona, a usled nedostatka tačnih podataka, na osnovu krive efikasnosti pretpostavljeno je da minimalna snaga koju inverter generiše iznosi približno 6% nominalne snage. Analizom rezultata dobijenih na osnovu proračuna tokova snaga, nakon priključenja fotonaponske elektrane, za uslove rada

koji podrazumevaju rad elektrane sa minimalnom snagom, može se zaključiti da se napon povećao u svakoj tački distributivnog sistema, pri čemu u bilo kojoj tački distributivnog sistema promena napona nije prekoračila limit od 5%.

4.4. Provera kriterijuma snage kratkog spoja

Ovim kriterijumom definisano je da vrednost struje (snage) ne sme preći maksimalno dozvoljene vrednosti struja (snaga) kratkog spoja na koje je dimenzionisana oprema u DS [3]. Za razmatrani slučaj, maksimalna dozvoljena komponenta struje kratkog spoja, na mestu priključenja elektrane na distributivni sistem ne sme biti veća od 1 kA.

Analizom rezultata dobijenih na osnovu proračuna trofaznog kratkog spoja (slika 4.4.1) primećuje se da vrednost struje kratkog spoja, na mestu priključenja elektrane na distributivni sistem (sabirnica S20) iznosi 0,806 kA, pri čemu definisana vrednost od 1 kA nije prekoračena.



Slika 4.4.1. Vrednost struje 3FKS na mestu priključenja.

4.5. Provera kriterijuma dozvoljenih struja viših harmonika i interharmonika

Ovaj kriterijum se proverava pomoću izraza [3]:

$$I_{vhg} \leq I_{vhdoz} = I_{vhs,v,\mu} * S_{ks} \quad (4.5.1)$$

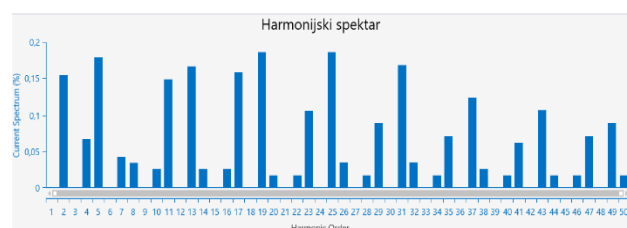
Stvarna vrednost struje viših harmonika koju generator injektira u DS I_{vhg} [A], svedena na mesto priključenja, se dobija na osnovu analize harmonika u softverskom alatu ETAP. Vrednosti ovih struja date su u tabeli 4.5.1.

Dozvoljena vrednost struje viših harmonika I_{vhdoz} [A], na mestu priključenja, proračunata je u tabeli 4.5.1.

Dozvoljena vrednost struje viših harmonika $I_{vhs,v,\mu}$ [A/MVA] svedena na jediničnu snagu kratkog spoja na mestu priključenja na DS definisana je u tabeli 4.2 u dokumentu "Pravila o radu distributivnog sistema" [3].

Snaga trofaznog kratkog spoja u tački priključenja na distributivni sistem S_{ks} [MVA], za naponski nivo 20 kV je tipizirana i iznosi 500 MVA.

Spektar viših harmonika struje za korišćeni model invertora prikazan je na slici 4.5.1.



Slika 4.5.1. Harmonijski spektar struje korišćenog invertora.

Tabela 4.5.1. Stvarna i dozvoljena vrednost struje viših harmonika/interharmonika na mestu priključenja.

Br. višeg harmonika	I_{vhg} [A]	$I_{vhdoz}=I_{vhs,v,u}*S_{ks}$ [A]	$I_{vhg} \leq I_{vhdoz}$
2	0,044	$0,029*500=14,5$	Ispunjeno
3	0	/	/
4	0,019	$0,009*500=4,5$	Ispunjeno
5	0,05	$0,029*500=14,5$	Ispunjeno
6	0	$0,012*500=6$	Ispunjeno
7	0,012	$0,041*500=20,5$	Ispunjeno
8	0,01	$0,004*500=2$	Ispunjeno
9	0	/	/
10	0,007	$0,007*500=3,5$	Ispunjeno
11	0,042	$0,026*500=13$	Ispunjeno
12	0	$0,005*500=2,5$	Ispunjeno
13	0,047	$0,019*500=9,5$	Ispunjeno
14	0,007	$0,003*500=1,5$	Ispunjeno
16	0,007	$0,003*500=1,5$	Ispunjeno
17	0,045	$0,011*500=5,5$	Ispunjeno
18	0	$0,002*500=1$	Ispunjeno
19	0,052	$0,009*500=4,5$	Ispunjeno
20	0,005	$0,0015*500=0,75$	Ispunjeno
22	0,005	$0,0013*500=0,65$	Ispunjeno
23	0,03	$0,006*500=3$	Ispunjeno
25	0,052	$0,005*500=2,5$	Ispunjeno
26	0,01	$0,0011*500=0,55$	Ispunjeno
27	0	$(0,05-0,92)*500=(25-460)$	Ispunjeno
28	0,005	$0,00107*500=0,53$	Ispunjeno
29	0,025	$(0,05-0,86)*500=(25-430)$	Ispunjeno
30	0	$0,001*500=0,5$	Ispunjeno
32	0,01	$0,0009*500=0,45$	Ispunjeno
33	0	$(0,05-0,75)*500=(25-375)$	
35	0,02	$(0,05-0,71)*500=(25-355)$	Ispunjeno
37	0,035	$(0,05-0,67)*500=(25-335)$	Ispunjeno
38	0,008	$0,00078*500=0,39$	Ispunjeno
40	0,005	$0,00075*500=0,37$	Ispunjeno
42	0	$0,00071*500=0,35$	Ispunjeno
44	0,005	$0,00068*500=0,34$	Ispunjeno
48	0	$0,00062*500=0,31$	Ispunjeno
50	0,005	$0,0006*500=0,3$	Ispunjeno

Stvarna vrednost struje viših harmonika, ni u jednom slučaju, ne prelazi dozvoljenu vrednost, pa se može zaključiti da je kriterijum dozvoljenih struja viših harmonika i interharmonika u potpunosti ispunjen.

5. ZAKLJUČAK

Na osnovu dobijenih rezultata, možemo zaključiti da razmatrana fotonaponska elektrana ispunjava sve zadate kriterijume za priključenje i bezbedan paralelan rad sa distributivnim sistemom, što podrazumeva normalan pogon distributivnog sistema, uz nenarušavanje pouzdanosti isporuke i kvaliteta električne energije drugim korisnicima.

Ukoliko rezultati pokažu da su uslovi rada distributivnog sistema nakon priključenja fotonaponske elektrane neprihvatljivi sa stanovišta ispunjenosti tehničkih uslova, neophodno je definisati odgovarajuće korake koje je potrebno preduzeti u cilju ostvarivanja normalnog pogona distributivnog sistema.

6. LITERATURA

- [1] John Twidell, Renewable energy resources, 4 edition, Routledge, 2022.
- [2] IEA, "Renewables 2023", IEA, Paris, 2024. <https://www.iea.org/reports/renewables-2023/executive-summary>
- [3] Pravila o radu distributivnog sistema, 00.000.-08.01.-180303/1-17, Beograd, 13.07.2017.
- [4] <https://etap.com/sectors/distribution>
- [5] Mohammad Kakueinejad, Azim Heydari, Mostafa Askari, Farshid Keynia, "Optimal Planning for the Development of Power System in Respect to Distributed Generations Based on the Binary Dragonfly Algorithm", Applied Sciences, Vol 10. 4795, 2020.
- [6] <https://www.luxor.solar/en/solar-modules/eco-line-gg/hjt-glass-glass-m132-680-700wp.html>
- [7] <https://solar.huawei.com/download?p=%2F-%2Fmedia%2FSolarV4%2Fsolar-version%2Fcommon%2Fprofessionals%2Fall-products%2Futility-smart-pv%2Fsmart-pv-controller%2FSUN2000-330KTL-H1%2Fsun2000-330ktl-h1-datasheet-20230515.pdf>

Kratka biografija:



Aleksandar Pavić rođen je u Trebinju 1998. god. Osnovne studije na Fakultetu tehničkih nauka na odseku elektroenergetski sistemi završio je 2022. god.
Kontakt: aleksandaracopavic@gmail.com



Dr Boris Dumnić rođen je 1976. god. Zaposlen je na Fakultetu tehničkih nauka, Departmanu za energetiku, elektroniku i telekomunikacije, Katedri za energetska elektroniku i pretvarače. Oblasni interesovanja su mu električne mašine, pogoni, energetska elektronika i obnovljivi izvori električne energije.

ИМПЛЕМЕНТАЦИЈА И ТЕСТИРАЊЕ ХИПЕРВИЗОРА ЗА ПАРАЛЕЛНО ИЗВРШАВАЊЕ ПАРТИЦИЈА НА ZEDBOARD ПЛАТФОРМИ

HYPERVISOR IMPLEMENTATION AND TESTING OF PARALLEL PARTITION EXECUTION ON ZEDBOARD PLATFORM

Ксенија Радоњић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – у овом раду описана је имплементација и рад хипервизора на развојној платформи ZedBoard. Ради се о слоју између хардвера и софтвера који оптимизује искоришћеност обје стране, омогућавајући извршавање вишеструких апликација на једном хардверу. За распоређивање паралелних партиција (задатака) на више процесорских језгара коришћен је EDF планер. Детаљно је објашњена структура и кораци за имплементацију система на ZedBoard плочи. Описана су спроведена тестирања комуникације, начина рада и распоређивања партиција на процесорским језгрима.

Кључне речи: хипервизор, распоређивање, партиција, паралелно извршавање, ZedBoard

Abstract – This article describes implementation and functionality of a hypervisor on the ZedBoard development board. Serving as a layer between software and hardware, the hypervisor optimizes resource utilization on both sides, facilitating the concurrent execution of multiple applications on the physical hardware. The Earliest Deadline First (EDF) algorithm is employed for scheduling partitions (tasks) across multiple processor cores. The article details the system structure and provides steps for implementing the hypervisor on the target board. Partition communication, interactions and scheduling modes are tested and thoroughly explained.

Keywords: hypervisor, scheduling, partition, parallel execution, ZedBoard

1. УВОД

Ембедед системи су рачунарски системи намијењени за извршавање специфичних задатака и функција, обично уграђени у уређаје који су веома ресурсно ограничени. Тема овог рада јесте развијање хипервизора [1] за овакво окружење, како би омогућио оптимизацију перформанси и смањење потрошње енергије уређаја. Хипервизор представља слој између софтверског и хардверског дијела система, чија је улога обезбјеђивање максималне искоришћености обје стране.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Предраг Теодоровић, ванр. проф.

Bare-metal хипервизор имплементира се директно на хардвер циљане платформе, без оперативног система. Предност ове методе је одвајање оперативног система од основног хардвера, тако да софтвер престаје бити директно зависан или ограничен на хардверске уређаје. Овакав приступ омогућава оперативним системима и повезаним апликацијама да функционишу на различитим типовима хардвера. Осим тога, хипервизор омогућава вишеструким оперативним система и виртуелним машинама истовремено дијелење хардверске платформе, на овај начин повећавајући искоришћеност ресурса и оптимизујући перформансе цјелокупног система.

Први корак при укључивању на пројекат представљао је упознавање и разумијевање развијеног концепта окружења. Хардвер коришћен за имплементацију хипервизора јесте развојна платформа ZedBoard Zynq-7000 SoC [2]. У ранијој фази пројекта развијен је алгоритам “EDF” (eng. Earliest Deadline First) планера [3], чија је улога ефикасно распоређивање партиција (односно задатака) на више процесорских језгара. Након детаљне анализе рада алгоритма, развијен је фајл генератор за аутоматизацију процеса формирања структуре фајлова неопходне за конфигурацију хипервизора. Наредни кораци подразумијевали су имплементацију система на плочу, а потом и тестирање распореда, комуникације и начина рада партиција.

2. ХИПЕРВИЗОР

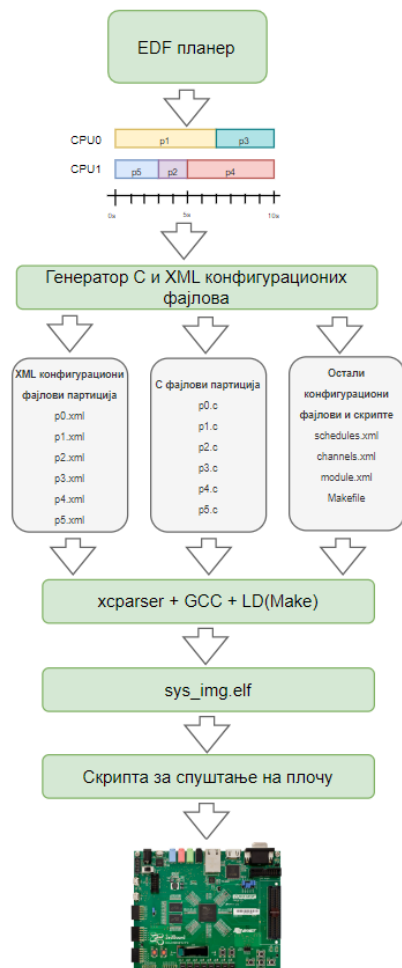
Поглавље описује теоријске основе хипервизора који омогућава покретање и управљање више оперативних система на једном хардверу. Слика 1 илуструје архитектуру таквог модела, са три истовремено покренута оперативна система.



Слика 1. Архитектура система са хипервизором

У овом раду је представљен рад са “*XRE*” (eng. *XtratuM Runtime Environment*), односно системом који обезбеђује задатке са минималним “*C*” окружењем за извршавање.

Приказ корака за имплементацију система приказан је на слици 2. У наредним поглављима детаљно ће бити објашњен сваки дио дијаграма.



Слика 2. Кораци за имплементацију система на хардвер

2.1 “EDF” планер

Циљ имплементације “*EDF*” алгоритма јесте да на најефикаснији начин направи план, односно распоред задатака који обављају различите функционалности. Начин рада овог алгоритма јесте такав да се процеси постављају у приоритетни ред, тако да када језгро процесора буде спремно да изврши следећи задатак бира задатак са најранијим роком за извршење.

Осим тога, алгоритам је “*non-preemptive*”, што значи да задатак који започне да се извршава на неком од процесора мора и да се настави на истом процесору, све до краја његовог извршења. Информације које је потребно обезбједити планеру као улаз су:

- Број језгара процесора
- Крајњи рок за чије вријеме требају бити извршени сви задаци на процесорима
- Идентификатор сваког задатка и вријеме потребно да буде извршен
- Зависности међу задацима, као и крајњи појединачни рокови сваког од њих

2.2 Партиција

Партиција представља основну јединицу сегрегације како у временском, тако и у просторном смислу. Могу се представити као пасивни контејнери који обезбеђују виртуелно извршавање одређеног окружења, у овом случају “*XRE*” система. За почетак партицијама нису додијељени комплексни задаци, већ извршавање основних радњи које би указале да је исправно конфигурисан систем, односно распоред партиција на више језгара, комуникација посредством специјализованих портова, итд. Заправо се ријеч задатак из претходног поглавља може изједначити са ријечју партиција, и може се рећи да “*EDF*” планер распоређује партиције на процесорска језгра. У другом кораку на слици 2 приказан је распоред пет партиција на два процесорска језгра, са дужинама трајања партиција представљеним на временској оси.

2.3 Развијање фајл генератора

Израз програма “*EDF*” планера је испис распореда партиција на процесорима у терминалу. Наредни корак у пројекту представљао је развијање фајл генератора за аутоматизацију формирања структуре датотека која конфигурише хипервизор, с обзиром на то да је мануелно уписивање добијених података у фајлове неефикасно решење.

Након темељног упознавања са радом “*EDF*” планера, код је модификован тако да екстрахује све неопходне податке и уписује их у текстуални фајл “*data_partitions.txt*”. Фајл генератор развијен у “*C++*” програмском језику, ишчитава и парсира податке из “*data_partitions.txt*” фајла. Користи библиотеку “*tinyxml2*” за упис форматираних података у *XML* конфигурационе фајлове. Осим тога, формира cjелокупну структуру података у директоријуму пројекта. У њему се налазе:

- **фолдер *xml*:**
 - *module.xml*
 - *channel.xml*
 - *hypervisor.xml*
 - *multipartition_health_monitor.xml*
 - *schedules.xml*
 - *<partition_name>.xml*
- ***<partition_name>.c***
- ***Makefile***

Наведени фајлови детаљније су објашњени у наредним поглављима.

2.4 *XML* конфигурациони фајлови

Систем се статички конфигурише преко *XML* конфигурационих фајлова, који се називају “*XCF*” (eng. *XML Configuration Files*). Улога “*XCF*” јесте дефинисање системских ресурса и начина на који се додјељују свакој партицији и хипервизору током извршавања система. Неке од информација које се налазе у овим фајловима су подаци о плочи, броју

процесора, фреквенцији рада и начину кеширања. У њима се такође конфигурише конзола за интеракцију са корисником, комуникациони канали који повезују дефинисане партиције, стриктно дефинише распоред партиција на процесорима, итд.

2.5 Изворни фајлови партиција

У “<partition_name>.c” фајловима налазе се изворни кодови партиција, гдје је имплементирана њихова функционалност. На почетку фајла укључују се библиотеке и заглавља за руковање партицијама, развијени у претходним фазама пројекта. Позивају се предефинисане функције за провјеру исправног учитавања партиције и комуникационих канала, а потом имплементирају жељене операције. У фази пројекта коју описује овај рад, циљ имплементације партиције јесте слање једноставних порука свим партицијама које зависе од ње.

2.6 Генерисање извршне датотеке

“Makefile” је текстуални фајл који се користи за аутоматизацију процеса компајлирања и изградње софтвера. Након формирања структуре фајлова, развијен је “Makefile” чија је улога парсирање “XCF”, као и компајлирање и линковање изворних фајлова. Коришћен је рачунар са “Ubuntu” оперативним системом, а за компајлирање “GCC” (eng. *GNU Compiler Collection*) компајлер. Позивом команде *make* покрене се “Makefile” и компајлирају сви фајлови, тако да се као излаз добија извршна датотека “sys_img.elf”. Ова бинарна датотека садржи машински код и потребне информације о партицијама и хипервизору за учитавање података у меморију циљане хардверске платформе.

2.7 Имплементација

У овом раду коришћена је ZedBoard развојна плоча, која припада серији Zynq7000 SoC (eng. *System on Chip*) платформи и приказана је на слици 3.



Слика 3. ZedBoard развојна плоча

ZedBoard је развојна плоча која комбинује ARM процесор са FPGA (eng. *Field-Programmable Gate Array*) технологијом. У овом пројекту користи се само “PS” (eng. *Processing System*) дио интегрисаног кола, што подразумева два 32-bit ARM Cortex A-9 процесорска језгра, без коришћења FPGA.

Рачунар, на ком се налази извршна датотека са свим подацима, повезује се са плочом коришћењем два USB (eng. *Universal Serial Bus*) кабла. Један служи за

серијско повезивање са конзолом и приказ порука партиција. Други USB кабл је “JTAG” програматор који спушта хипервизор на циљану плочу. “JTAG” (eng. *Joint Test Action Group*) је интерфејс који се користи за комуникацију између рачунара и развојних плоча како би се омогућило програмирање и дебаговање интегрисаних кола на истим.

Крајња скрипта “jtag_deployment.sh” увезује све неопходне датотеке и имплементира систем на ZedBoard плочу. Уколико се на исправан начин покрене, сви дјелови пројекта ће заузети меморијске ресурсе на начин који је дефинисан. Партиције ће се распоредити на процесорима и започети своје извршавање. У терминалу је кориснику омогућено праћење повратних информација, односно порука које партиције исписују. На овај начин вршено је дебаговање и истраживање различитих функционалности система. Овај дио експерименталне фазе описан је у наредном поглављу.

3. ЕКСПЕРИМЕНТАЛНА ФАЗА

3.1 Испис порука на конзолу

Прва фаза тестирања обухватила је испитивање правилног распоређивања партиција на хардверској плочи. За дебаговање је у изворним фајловима партиција коришћена функција за испис порука на конзолу. Ово не представља ефикасно решење јер захтјева додатно процесорско вријеме за обраду исписа, али у том тренутку ефикаснији алати нису били развијени. Проблем је постао изразитији због паралелне обраде на два процесорска језгра, што је изазвало преклапање и преплитање порука различитих партиција.

У циљу решавања проблема преплитања порука, примијењени су мутекси. Мутекс (eng. “*mutual exclusion*”, односно „међусобно искључивање”) је механизам синхронизације који се користи у паралелном програмирању како би се осигурало да само једна нит (партиција) може да приступи одређеном дијелу ресурса у датом тренутку. Међутим, сама операција исписа порука на конзолу је ресурсно захтјевна из следећих разлога:

- поруке које се шаљу на конзолу чувају се у баферу прије прослијеђивања на излазни ток
- процесор мора прекинути свој рад да би исписао поруку на конзолу
- током процеса слања података на конзолу, процесор мора обрадити те поруке и извршити операције за ову радњу

Наведене ставке захтјевају значајно вријеме и ресурсе процесора. У контексту времена извршавања, ако се задатак понављано зауставља ради исписа порука на конзолу, то може довести до прекорачења његовог рока извршавања, посебно у окружењима гдје је вријеме извршавања критично.

Из наведених разлога је додатна пажња посвећена минимизацији броја, као и дужине порука које се исписују. Након експериментисања са позицијом исписа у коду, дужине поруке, броја података, мутекси-ма, итд., партиције су успјешно имплементиране и

правилно распоређене на, за њих предвиђеном, процесору.

3.2 Коришћење комуникационих канала

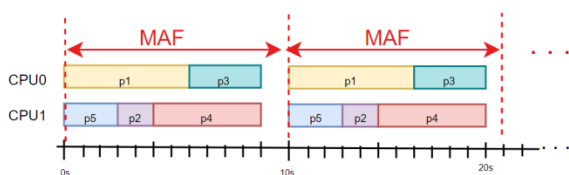
За комуникацију међу партицијама користе се комуникациони портови. Режим порта је “*queuing*”, који функционише на начин да се поруке из партиција чувају у “*FIFO*” (eng. “*First In First Out*”) баферу. Све нове поруке чекају у реду, а затим се обрађују редоследом којим су и уписане. Сви коришћени канали морају бити дефинисани у изворним фајловима партиција и *XML* конфигурационим фајловима.

На почетку је имплементирана најједноставнија функционалност гдје свака партиција дефинише портове и шаље поруку оним партицијама које зависе од ње. У овом кораку је комуникација, односно коришћење механизма канала и портова међу партицијама, успешно имплементирана. Омогућен је пренос информација међу задацима, што је од велике важности за касније фазе пројекта чији је циљ обезбјеђивање паралелног рада много комплекснијих апликација.

Због тога је следећи корак био проширивање величине, као и броја порука које се шаљу. Успјешно је тестиран примјер гдје партиција шаље више порука партицији зависној од ње. Након што их прими, зависна партиција исписује на конзолу укупан број примљених карактера чија вриједност потврђује да су све прочитане, а потом и садржај последње примљене порука чиме се осигурава исправност пријема. Оваква имплементација од великог је значаја због информација које партиције у многим случајевима морају међусобно размијенити.

3.3 Тестирања режима рада система

У складу са циљем пројекта који подразумева имплементацију реалних оперативних система, извршене су адаптације кода ради обезбјеђивања непрекидног рада система. Важан атрибут планера јесте његов главни оквир, “*MAF*” (eng. *Major Frame*), дефинисан као временски интервал у ком се изврше све партиције. У овом кораку примјењен је концепт периодичног понављања распореда, приказан на слици 4.



Слика 4. Примјер рада са понављањем “*MAF*”-а

Након ове фазе, спровођени су различити експерименти за провјеру могућих режима рада партиција. Успјешно је имплементирана подјела једне партиције у два одвојена временска слота извршена у различитим временским интервалима, на истом процесору. Потом је успјешно спроведен и тест извршавања партиције на два различита процесора у истом временском интервалу. Ово је тестирано као провјера дате могућности и није даље испитивано, већ је остављено за касније фазе развоја пројекта.

У наредном тесту је имплементирано успјешно повезивање система са дигиталним пиновима *ZedBoard* плоче. Изворни код партиције је прилагођен тако да иницијализује пин ровезан са “*PB1*” дугметом уграђеним на хардверску платформу. Након успостављања комуникације ишчитава његово стање, а потом и исписује поруку са информацијом да ли је дугме притиснуто или не.

4. ЗАКЉУЧАК

У овом раду представљена је иницијална фаза пројекта у ком се развија хипервизор за оптимизацију перформанси ембедед система који су ресурсно ограничени. Објашњене су основе начина рада и структура система, кораци за његову имплементацију, а потом и различита тестирања којима је потврђен исправан рад система.

Будући да је описан рад почетна фаза цјелокупног пројекта многа унапређења су могућа, а наводе се само нека од њих:

- Коришћење *Zynq UltraScale MPSoC* [4] плоче са шест процесорских језгара.
- Развој специјализованих алата за дебаговање функционалности партиција.
- Имплементација оперативних система веће комплексности.

5. ЛИТЕРАТУРА

- [1] <https://www.vmware.com/topics/glossary/content/hypervisor.html>, април 2024.
- [2] <https://digilent.com/reference/programmable-logic/zedboard/reference-manual>, април 2024.
- [3] <https://www.baeldung.com/cs/scheduling-earliest-deadline-first>, април 2024.
- [4] <https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html>, април 2024.

Кратка биографија:



Ксенија Радоњић рођена је у Никшићу 1999. године. Дипломски рад на Факултету техничких наука из области Електротехника и рачунарство – Ембедед системи и алгоритми одбранила је 2022. године.
контакт:
radonjicksenija20@gmail.com

EKSPERIMENTI SA VEKTORSKIM REPREZENTACIJAMA TEKSTA ZA EKSTRAKTIVNU SUMARIZACIJU**EXPERIMENTS WITH TEXT EMBEDDING METHODS FOR EXTRACTIVE SUMMARIZATION**

Aleksa Šaršanski, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – U ovom radu je implementirano rešenje problema ekstraktivne sumarizacije komentara za smeštaj u hotelima koje su napisali korisnici. Rešenje je implementirano koristeći FastText, BERT i GPT 3.5 modele dubokog učenja. U radu je detaljno opisan način funkcionisanja sva tri pomenuta modela, kao i načini njihove upotrebe prilikom rešavanja ovog problema. Skup podataka koji je korišćen u ovom rešenju sastoji se od javno dostupnih komentara korisnika, kreiran procesom scrape-ovanja. U radu je prilikom evaluacije rezultata upoređena efikasnost sva tri navedena pristupa rešavanja problema.

Ključne reči: Veštačka inteligencija, duboko učenje, ekstraktivna sumarizacija, NLP, FastText, BERT, GPT

Abstract – In this paper is implemented a solution to the problem of extractive summarization of user-written hotel accommodation reviews. The solution uses FastText, BERT, and GPT-3.5 deep learning models. The paper provides a detailed description of how each of these models works, as well as their application methods in addressing this problem. The dataset used in this solution consists of publicly available user comments obtained through the web scraping process. The effectiveness of all three mentioned approaches in solving the problem is compared during the evaluation of the results.

Keywords: Artificial Intelligence, Deep learning, Extractive summarization, NLP, FastText, BERT, GPT

1. UVOD

Sumarizacija teksta predstavlja proces sažimanja teksta, čiji je rezultat kraći tekst koji sadrži sve bitne informacije i suštinu iz originalnog teksta. U oblasti mašinskog učenja, sumarizacija teksta spada u NLP (*Natural Language Processing*) probleme. Obzirom da rezultat sumarizacije teksta nije jednoznačan, svrstava se među teže probleme u oblasti NLP-a. Ekstraktivna sumarizacija predstavlja izdvajanje najvažnijih rečenica iz teksta i tako formira rezultujući tekst. U ovom pristupu se ne menja originalni tekst, već se samo izdvajaju njegovi najrelevantniji delovi. Njena primena se može naći u novinarstvu, sažimajući duge novinske članke, u apstrakciji dugih pravnih dokumenata, u brzom pregledu

dugih medicinskih izveštaja, u analizama raznih poslovnih dokumenata... Ovaj rad se bavi problemom ekstraktivne sumarizacije komentara na online platformama za pretragu i rezervaciju smeštaja. Komentari putnika znaju da budu veoma dugački, da sadrže mnogo informacija koje korisnicima nisu presudne pri donošenju odluke o rezervaciji i obično zahtevaju mnogo izdvojenog vremena za analizu i izvlačenje bitnih informacija.

Motivacija za rešavanje ovog problema jeste da se putnicima olakša odabir smeštaja, tako će se iz svakog komentara izvući najrelevantniji delovi i samim tim značajno uštedeti vreme koje je potrebno korisniku da donese odluku koji će smeštaj rezervisati. Iako je veoma korisna, ova funkcionalnost je retko zastupljena na spomenutim platformama. Implementacija ekstraktivne sumarizacije teksta u ovom radu obuhvata 3 različita pristupa, odnosno upotrebu 3 različita modela dubokog učenja – FastText, BERT i GPT 3.5 modela. Prva dva pristupa (FastText i BERT) koriste TextRank algoritam za rangiranje rečenica. Implementacija pomoću GPT 3.5 modela obuhvata upotrebu svih njegovih slojeva neuronske mreže, bez dodatnih modifikacija i klasifikacija izlaza.

2. PREGLED STANJA U OBLASTI**2.1. Radovi koji se bave TextRank algoritmom**

U svom radu [1] 2004. godine, TextRank model je predstavljen od strane Rada Mihalcea i Paul Tarau. On se zasniva na grafovima i ima dve primene u rešavanju problema iz oblasti obrade prirodnog jezika. Prva njegova primena je ekstrahovanje ključnih reči iz teksta. Za ovu primenu je upotrebljen skup podataka koji se sastoji od kolekcije od 500 sažetaka iz Inspec baze podataka i odgovarajućih manuelno dodeljenih ključnih reči. Evaluacija rezultata je odrađena korišćenjem *precision*, *recall* i *F-measure* metrika, pri čemu su za evaluaciju rezultata korišćeni sistemi za *state-of-the-art* ekstrakciju ključnih reči iz Hulth 2023 eksperimenta.

Fokus druge primene TextRank algoritma jeste na ekstrakciji rečenica iz teksta. Efikasnost ovog eksperimenta proverena je pomoću *single-document* sumarizaciju nad skupom od 567 novinskih članaka iz Document Understanding Evaluations 2002.

2.2. Radovi koji koriste LLM-ove (Large Language Modele)

U maju 2023. godine, Abhishek Kumar je u svom radu [2] uradio istraživanje na temu poređenja efikasnosti nekoliko modela dubokog učenja za rešavanje problema ekstrak-

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kovačević, red. prof.

tivne sumarizacije teksta. Modeli koji su obuhvaćeni radom su: BERT, GPT-2, KL-summarizer, Luhn, LEX i Word Rank. Evaluacija rezultata izlaza modela je rađena pomoću tri tehnike, Rouge score, BERT score i Mover score, gde su rezultati upoređeni sa ljudskim, ručno sumarizovanim komentarima iz dva skupa podataka, BBC New Datasets i DailyMail/CNN datasets. U ovom istraživanju su se kao najefikasniji modeli za ekstraktivnu sumarizaciju teksta pokazali GPT-2 i Luhn, uzimajući u obzir rezultate sve tri navedene tehnike evaluacije.

3. TEORIJSKI POJMOVI I DEFINICIJE

3.1 Vestačka inteligencija

Vestačka inteligencija (eng. *Artificial Intelligence*) – na osnovu pronalaska šablona unutar velike količine podataka omogućava računarima da sami donose odluke ili izvršavaju određene aktivnosti kako bi ostvarili konkretne ciljeve [3].

3.2 Mašinsko učenje

Mašinsko učenje (eng. *Machine Learning*) je podoblast veštačke inteligencije koja se bavi razvojem algoritama i računarskih sistema koji imaju sposobnost adaptacije na nove, do sada neviđene situacije. To im omogućava velika količina podataka u kojoj nalaze šablone koji im omogućavaju da donose odluke uz minimalne ljudske intervencije.

3.3 Duboko učenje

Duboko učenje (eng. *Deep Learning*) – predstavlja podoblast mašinskog učenja koja je zasnovana na veštačkim neuronskim mrežama, algoritmima koji su inspirisani ljudskim mozgom. Veštačka neuronska mreža je matematički model sastavljen od veštačkih neurona – perceptrona gde svaki veštački neuron predstavlja pojednostavljen model biološkog neurona [3].

3.4 NLP – Natural Language Processing

NLP (*Natural Language Processing*) je grana veštačke inteligencije koja se bavi obradom prirodnog, ljudskog jezika, bilo to obrada i analiza teksta ili govora. Ovaj metod kombinuje računarsko-lingvističko modelovanje ljudskog jezika sa statističkim modelima mašinskog učenja kako bi omogućila računarima da prepoznaju, razumeju i generišu tekst i govor [4]. Neke od najčešćih primena NLP-a su:

- Prepoznavanje govora (eng. *Speech recognition*),
- Pretvaranje teksta u govor (eng. *Text-to-Speech*),
- Segmentacija reči (tokenizacija),
- Analiza sentimenta (eng. *Sentiment Analysis*),
- Sumarizacije teksta (eng. *Text summarization*),
- Generisanje prirodnog jezika (eng. *Natural language generation*)...

3.5 Vektori reči (Word embeddings)

S obzirom da modeli mašinskog učenja rade samo sa brojevima (oni su matematički modeli), njima reči koje su sastavljanje od slova ne znače ništa. Iz tog razloga, reči je potrebno nekako prikazati pomoću brojeva, pa se zbog toga uvodi pojam vektora reči. Vektor reči predstavlja numeričku reprezentaciju reči u višedimenzionalnom matematičkom prostoru. Svaka reč je predstavljena

pomoću jedinstvenog vektora koja joj određuje njenu poziciju u vektorskom prostoru. Vektorska reprezentacija reči omogućuje modelima mašinskog učenja da razumeju reči, na osnovu njihovog semantičkog i sintaksnog značenja. Slične reči se nalaze blizu u vektorskom prostoru, dok se različite reči nalaze daleko jedna od druge. Tako se na primer reč "jabuka" nalazi blizu reči "voće" u vektorskom prostoru, ali se nalazi daleko od reči "automobil". Najreprezentativniji primer kako vektori reči funkcionišu su reči "king", "queen", "man" i "woman", gde za njihove vektorske reprezentacije važi "king" + "woman" – "man" $\approx$ "queen", odnosno "king" – "man" $\approx$ "queen" – "woman"

3.6 FastText algoritam

FastText je razvijen od strane Facebook-a 2016. godine. Koristi za dobijanje vektorskih reprezentacija reči koje potom koristi za rešavanje raznih NLP problema kao sto su klasifikacija teksta i dokumenata, modelovanje naslova, analiza sentimenta teksta... Ovaj algoritam čini neuronska mreža jednostavne arhitekture (bez mnogo slojeva) koja se odlično pokazala za kreiranje vektora reči, pri čemu se čuvaju semantičke i sintaksne osobine reči i zbog toga se često koristi. Ovaj algoritam umesto da kreira direktno vektor cele reči, deli reči na delove (eng. *n-gram*) i kreira vektor za svaki od *n-gram*-a, gde se rezultujući vektor reči dobija kao suma vektora svih *n-gram*-a te reči. Na ovaj način FastText čuva morfološku strukturu reči sto ga čini veoma moćnim i za kreiranje vektora reči složenijih jezika. Ovakav pristup mu omogućava da se dobro pokaže pri radu sa rečima koje se retko pojavljuju ili do sada nisu viđene sa njegove strane. Ovaj model je dostupan za 294 različita jezika.

3.7 Veliki jezički modeli (LLMs – Large Language Models)

LLM-ovi su veliki i složeni modeli dubokog učenja koji mogu da prepoznaju, sumarizuju, prevode, predviđaju i generišu tekst koristeći veoma velike skupove podataka. Koriste se za rešavanje raznih vrsta NLP zadataka, od najjednostavnijih do najsloženijih. Neki od najčešćih zadataka LLM-ova su: klasifikacija teksta (analiza sentimenta i emocija), sumarizacija teksta, prevođenje teksta sa jednog jezika na drugi, generisanje teksta, generisanje programskog koda...

Sastoje se od velikog broja slojeva neuronske mreže, obučavani su nad ogromnim skupom podataka (do nekoliko terabajta tekstualnih podataka) i sadrže veliki broj obučavajućih parametara (do nekoliko stotina milijardi parametara). Oni su obučavani na principu nenadgledanog učenja, što zahteva da se obučavaju na velikom broju nelabeliranih podataka.

3.8 BERT model

BERT (*Bidirectional Encoder Representations from Transformers*) je veliki jezički model koji radi na principu transformera. BERT predstavlja specifičan slučaj transformer modela jer se sastoji samo iz enkoder komponenti (ne sadrži dekodek komponente). Enkoder komponente su poredane jedna na drugu i vektori reči dobijeni iz jedne enkoder komponente se prosleđuju dalje na kodiranje narednoj enkoder komponenti. Ulaz u prvi enkoder je sekvenca reči.

BERT je pre-trenirani model, obučavan nad velikim skupom podataka nakon čega ga je u svrhe boljeg rešavanja određenog problema moguće do-trenirati novim skupom podataka koji pripadaju domenu kome pripada i sam rešavani problem.

3.9 GPT model

GPT 3 (*Generative Pre-Trained Transformer*) je treća generacija velikog jezičkog modela dubokog učenja predstavljena 2020. godine od strane kompanije OpenAI u čiji rad i razvoj ulaze kompanije Microsoft. Ovaj pre-trenirani model generiše tekst na osnovu predikcije naredne reči u rečenici primenom sistema “učenja bez učitelja” (eng. *unsupervised learning*).

Za njegovo obučavanje korišćeno je 175 milijardi obučavajućih parametara i oko 570 gigabajta podataka koji su sakupljeni sa raznih internet stranica, foruma, knjiga, članaka...

GPT 3 je transformer model koji je za razliku od BERT modela kojeg čine enkoder blokovi, izgrađen od dekode- blokova. Izgrađen je od 96 *attention* blokova gde se svaki od njih sastoji od 96 *attention head* – ova.

3.10 TextRank algoritam

TextRank algoritam pripada grafovskim algoritmima i oslanja se na Google-ov PageRank algoritam. Koristi se rangiranje tekstova, rečenica i reči tako sto meri vezi između dve ili više reci. Ima veliku primenu pri rešavanju raznih NLP problema kao sto su sumarizacija teksta, izdvajanje ključnih reci iz teksta, analiza citata...

4. METODOLOGIJA

4.1 Skup podataka

Skup podataka koji je korišćen u ovom radu za dotreniravanje i evaluaciju rezultata modela dubokog učenja za rešavanje problema ekstraktivne sumarizacije teksta je kreiran tako što je preuzet sadržaj komentara za hotele sa poznate internet stranice koja se bavi rezervacijom smeštaja. Svi komentari se odnose na utiske putnika koji su boravili u raznim hotelima na tropskoj destinaciji – Tajlandu.

Ovaj skup podatak sadrži 3983 komentara putnika koji se odnose na razne hotele na Tajlandu. Prilikom preuzimanja komentara, filtrirali su se samo oni komentari koji su na engleskom jeziku, jer su rešenja korišćena za rešavanja problema ekstraktivne sumarizacije teksta u ovom radu fokusirana samo na engleski jezik.

4.2 Pristup rešavanja problema pomoću FastText modela

Rešavanje problema ekstraktivne sumarizacije teksta pomoću FastText modela dubokog učenja se sastoji od dotreniravanja FastText modela pomoću domenskog skupa podataka opisanom u poglavlju 4.1 kako bi model dodelio vektorsku reprezentaciju rečima iz skupa podataka.

Na osnovu dobijenih vektorskih reprezentacija reči, traži se njihov prosek po rečenici pri čemu se dobija vektorska reprezentacija cele rečenice.

Tako dobijene vektorske reprezentacije rečenica se prosleđuju TextRank algoritmu koji ih rangira po važnosti i vraća željeni broj najbitnijih rečenica za komentar.

4.3 Pristup rešavanja problema pomoću BERT modela

Rešavanje problema ekstraktivne sumarizacije teksta pomoću BERT velikog jezičkog modela je u velikoj meri slično kao i rešavanje pomoću FastText modela koje je objašnjeno u prethodnom poglavlju. U tu svrhu je korišćena posebna vrsta BERT modela – *Sentence BERT* (SBERT). SBERT se od standardnog BERT modela razlikuje po tome što radi na nivou rečenica umesto na nivou reči kao standardan BERT model. Konkretno, umesto da kreira vektorske reprezentacije reči, on kreira vektorske reprezentacije rečenica. Nije potrebno računati prosečnu vrednost reči u rečenici da bi se dobila vektorska reprezentacija rečenice.

Kako bi odredio vektorsku reprezentaciju čitave rečenice, SBERT koristi Sijamsku arhitekturu (*Siamese architecture*). Sijamska arhitektura se sastoji od dve ili više identičnih podmreža koje paralelno generišu vektore (u ovom slučaju vektore rečenica) i porede ih kako bi im odredile poziciju u vektorskom prostoru.

Ova arhitektura neuronskih mreža se obično koristi prilikom rešavanja problema pronalaska duplikata u tekstovima, pronalaska anomalija, semantičku pretragu, klastervanje dokumenata...

U ovom slučaju Sijamske arhitekture, SBERT koristi paralelno dva standardna BERT modela, gde svakom prosleđuje po jedna rečenica i na izlaznim slojevima se određuje slično između njih i na osnovu toga im se dodeljuje vektorska reprezentacija. Dobijene vektorske reprezentacije rečenica se prosleđuju takođe TextRank algoritmu koji ih rangira po važnosti i vraća željeni broj najbitnijih rečenica za komentar.

4.4 Pristup rešavanja problema pomoću GPT 3.5 modela

Rešavanje problema ekstraktivne sumarizacije teksta. Pristup rešavanja problema ekstraktivne sumarizacije teksta pomoću GPT 3.5 velikog jezičkog modela se razlikuje od prethodnih rešenja u tome sto se ne radi dotreniravanje GPT 3.5 modela kao sto je to bio slučaj sa FastText i BERT modelima. U ovom pristupu se koriste već postojeće vektorske reprezentacije reči koje su kreirane prilikom obučavanja GPT modela. GPT zna kakav izlaz treba da generiše na osnovu *prompta* koji mu je prosleđen. *Prompt* predstavlja tekstualni fajl koji sadrži naredbe, odnosno instrukcije koje simuliraju komunikaciju čoveka i mašine i govore modelu kako bi trebao da se ponaša i kakav izlaz da generiše.

5. EKSPERIMENTI

5.1 Eksperiment 1

Eksperiment 1 obuhvata evaluaciju rešenja nad komentarima koji su već viđeni od strane modela prilikom dotreniravanja i nalaze se u obučavajućem skupu podataka. Kao sto je prethodno opisano, taj skup je dobijen na tri načina: bez primene lematizacije i steminga nad komentarima, sa primenom lematizacije nad komentarima i sa primenom steminga nad komentarima.

Komentari koji se u ovom slučaju koriste kao evaluacioni skup podataka, a zapravo su podskup obučavajućeg skupa podataka su dužine između osam i dvanaest rečenica.

Rezultati svakog eksperimenta sadrže sumarizaciju komentara na tri i na pet rečenica kako bi se i proverila zavisnost očuvanja semantičkih informacija iz originalnog komentara od dužine sumariзованog komentara. Evaluacioni skup broji 40 različitih originalnih komentara.

5.2 Eksperiment 2

Eksperiment 2 obuhvata evaluaciju rešenja nad komentarima koji do sada nisu viđeni od strane modela prilikom dotreniravanja i ne nalaze se u obučavajućem skupu podataka. Razlika između ova dva eksperimenta nije od velike važnosti za rešenje problema pomoću GPT 3.5 modela, s obzirom da nije rađeno njegovo dotreniravanje. Obučavajući skup korišćen u ovom eksperimentu dobijen je na tri načina: bez primene lematizacije i steminga nad komentarima, sa primenom lematizacije nad komentarima i sa primenom steminga nad komentarima. Komentari koji se u ovom slučaju koriste kao evaluacioni skup podataka su dužine između osam i dvanaest rečenica. Rezultati svakog eksperimenta sadrže sumarizaciju komentara na tri i na pet rečenica kako bi se i proverila zavisnost očuvanja semantičkih informacija iz originalnog komentara od dužine sumariзованog komentara. Evaluacioni skup broji 40 različitih originalnih komentara.

5.3 Evaluacija

Za procenu efikasnosti FastText, BERT i GPT 3.5 modela dubokog učenja za problema ekstraktivne sumarizacije komentara upotrebljene su ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) metrike. Upotrebljene ROUGE metrike su ROUGE-1, ROUGE-2 (koje pripadaju ROUGE-N metrikama) i ROUGE-L metrike. Pored navedenih metrika, procena efikasnosti modela je rađena i po subjektivnom osećaju autora sumariзованиh komentara uključenih u evaluaciju. ROUGE-N metrike (ROUGE-1 i ROUGE-2) porede broj odgovarajućih “n-grama” u tekstu, odnosno poređenje n broja reči 2 sumariзована komentara. ROUGE-1 se koristi za poređenje svake pojedinačne reči, dok se ROUGE-2 metrika koristi sa pronalaženje poklapanja parova reči, to mogu biti na primer neke fraze koje označavaju da li je sačuvana semantika komentara prilikom sumarizacije. ROUGE-L metrika meri najdužu sekvencu reči koja se poklapa između generisanog i referentnog komentara. Prilikom evaluacije eksperimenata, prikazane su vrednosti *F1 Score* metrike koja predstavlja prosečnu vrednost između ostalih *precision* i *recall* metrika. S obzirom da je za svaki eksperiment korišćeno 40 različitih komentara, prikazana vrednost *F1 Score* metrike za svaki testni slučaj predstavlja prosečnu vrednost *F1 Score* metrike za svih 40 komentara za taj testni slučaj.

6. REZULTATI I DISKUSIJA

U tabelama 3 i 4 možemo da vidimo modele koji su dali najbolje rezultate na oba eksperimenta za ekstraktivnu sumarizaciju komentara na 3 i na 5 rečenica. Prilikom ekstraktivne sumarizacije na 3 rečenice, po svim ROUGE metrikama najbolje se pokazao BERT model i to u Eksperimentu 2 gde je evaluacija rađena nad podacima koji do sada nisu viđeni od strane modela. BERT je za sva 3 načina pre-procesiranja obučavajućeg skupa podataka pokazao identične, najbolje rezultate. Situacija je ista što

se tiče i ekstraktivne sumarizacije na 5 rečenica. BERT se takođe po svim metrikama pokazao kao najbolje rešenje nezavisno od načina pre-procesiranja podataka i to ponovo u Eksperimentu 2.

Na osnovu prethodnih zaključaka možemo da kažemo da se BERT model dubokog učenja pokazao najuspešnije u ovom radu prilikom rešavanja problema ekstraktivne sumarizacije komentara. Najbolje rezultate u pogledu na oba eksperimenta pokazao je prilikom ekstraktivne sumarizacije teksta na 5 rečenica korišćenjem podataka za evaluaciju koje do sada nije video (ne nalaze se u obučavajućem skupu podataka). Njegove performanse ne zavise od toga da li je nad podacima u obučavajućem skupu primenjen algoritam steminga i lematizacije ili nije.

7. ZAKLJUČAK

U ovom radu su implementirana tri pristupa rešavanja problema ekstraktivne sumarizacije komentara korisnika hotela sa poznate internet platforme za rezervaciju smeštaja na Tajlandu. Svaki od tri pristupa koristi model veštačke inteligencije za rešavanje ovog problema – FastText, BERT i GPT 3.5. Iako se u eksperimentima pokazalo da se za rešavanje ovog problema najbolje pokazao BERT model, FastText i GPT 3.5 su se pokazali dovoljno dobro u cilju istog. Zanimljivo proširenje zadatka koje bi skoro sasvim sigurno povećalo uspešnost njegovog rešavanja bi bilo da se ekstraktivna sumarizacija komentara pomoću određenih *prompt*-ova implementira koristeći novije velike jezičke modele kao što su: *GPT 4*, *Antropic Claud*, *Amazon Titan*, *Google Bard*, *Mistral*, *Mixtral* i drugi koji su javno dostupni pomoću svojih provajdera.

8. LITERATURA

- [1] Rada Mihalcea and Paul Tarau, 2004. *TextRank: Bringing Order into Texts* <https://aclanthology.org/W04-3252.pdf>
- [2] Abhishek Kumar, 2023. EXTRACTIVE TEXT SUMMARIZATION <http://14.139.251.106:8080/jspui/bitstream/repository/20079/1/Abhishek%20Kumar%20Mtech.pdf>
- [3] Complete Guide to TensorFlow for Deep Learning with Python by Jose Portilla – Udemy course - <https://www.udemy.com/course/complete-guide-to-tensorflow-for-deep-learning-with-python/>
- [4] What is natural language processing (NLP)? - <https://www.ibm.com/topics/natural-language-processing>

Kratka biografija:



Aleksa Šaršanski rođen je u Novom Sadu 23.09.1998. godine. Osnovne akademske studije završio je 2022. godine, na Fakultetu Tehničkih Nauka u Novom Sadu, smer Primenjeno softversko inženjerstvo. Godine 2022. upisuje master akademske studije na studijskom programu Primenjene računarne nauke, odsek Inteligentni sistemi kontakt: aleksa.berdih@gmail.com

DIZAJN I IMPLEMENTACIJA JEZIKA SPECIFIČNOG ZA DOMEN LOGIČKIH ENIGMATSKIH IGARA**THE DESIGN AND IMPLEMENTATION OF DOMAIN-SPECIFIC LANGUAGE FOR LOGIC PUZZLE GAMES**

Sara Panić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Cilj rada jeste istraživanje, projektovanje i razvijanje jezika koji će omogućiti lakše i brže predstavljanje i analizu različitih vrsta logičkih enigmatičkih igara. Nakon toga implementiran je prototip sistema za implementaciju i izvršavanje logičkih enigmatičkih igara koristeći razvijeni jezik što uključuje razvoj osnovnih modula za rad sa zagonetkama i njihovu specifikaciju i interpretaciju.

Ključne reči: JSD, textX, gramatika, enigmatika, logičke igre

Abstract – The aim of the work is to research, design, and develop a language that will enable easier and faster representation and analysis of various types of logical puzzle games. After that, a prototype system for implementing and executing logical puzzle games was implemented using the developed language, which includes the development of basic modules for working with puzzles and their specification and interpretation.

Keywords: DSL, textX, grammar, enigma, logic games

1. UVOD

Enigmatika obuhvata širok spektar logičkih igara koje zahtevaju rešavanje zagonetki ili mozgalica. Ove igre predstavljaju zanimljiv domen istraživanja u računarstvu i veštačkoj inteligenciji. Ono što ih čini izazovom u ovoj sferi jeste kombinatorika i logička analiza koje su neophodne da bi se igra definisala i rešila, često uz kombinaciju specifičnih strategija i tehnika. Računarske verzije ovih igara omogućavaju korisnicima da ih rešavaju na svojim uređajima, često uz podršku naprednih algoritama za generisanje, interpretaciju i rešavanje istih.

S obzirom na to, ovaj rad se fokusira na razvoj jezika za opisivanje pravila i karakteristika logičkih igara enigmatike. Korišćen je alat textX za definisanje gramatike jezika, što je omogućilo preciznu specifikaciju igara kroz formalne strukture jezika. Praktični deo za cilj ima primenu teorijskog znanja i rezultata istraživanja i predstavlja projekat koji se oslanja na implementaciju veb aplikacije koja koristi definisani jezik za generisanje i reprezentaciju logičkih igara. Aplikacija se sastoji iz korisničkog interfejsa koji omogućava unos novih primera igara, ali i rešavanje već unetih.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Igor Dejanović.

Serverski deo koristi parser zasnovan na textX-u za interpretaciju i generisanje igara na osnovu unetih specifikacija.

2. KLASIFIKACIJA LOGIČKIH IGARA

Enigmatiku definišemo kao veštinu sastavljanja i rešavanja zagonetki. Pored toga što se tretira kao razbibriga i zabava, ima veliku ulogu u testiranju inteligencije, proveru i sticanju opšteg znanja. Često se naziva i "gimnastikom uma i igrom duha" [1]. Razlikujemo više grupa enigmatike: mrežaste zagonetke, zagonetke sa premetanjem slova, logičke igre, rebusi, kriptogrami, matematičke, likovne zagonetke i još mnogo drugih.

Sudoku (eng. *sudoku*) - Sudoku predstavlja popularnu zagonetku koja zahteva veštinu, strategiju i zaključivanje iz pretpostavki sa složenim uslovima [2]. Igra se na tablici od najčešće 9x9 polja podeljenih na 3x3 regiona, gde igrač mora da popuni sva polja ciframa od 1 do 9 tako da svaki red, kolona i svaki 3x3 region sadrže sve cifre bez ponavljanja. Postoje različite varijacije sudoku igre koje uključuju različite rasporede startnih cifara ili dodatna pravila koja usložnjavaju igru (na primer: dijagonalni, geometrijski, parni-neparni, samuraji itd.). Takođe, možemo ih razlikovati i po kompleksnosti, koju najčešće definišemo brojem cifara koje su inicijalno prikazane na poljima. Pravilo koje mora biti zadovoljeno da bi sudoku bio rešiv, bez obzira na kompleksnost, jeste da je inicijalno dato minimum 17 brojeva [3].

Ukrštene reči (eng. *crosswords*) - Ukrštene reči predstavljaju zabavnu i izazovnu mrežastu zagonetku u kojoj igrač treba da pronađe i ukrsti skrivene reči u mreži slova. Ova igra nije samo popularna za rekreaciju, već i za razvoj vokabulara i veštine logičkog razmišljanja [4]. Zahteva od igrača poznavanje opšte kulture i informisanosti i široko znanje iz različitih oblasti. Reči se unose horizontalno ili vertikalno i ukrštaju se najmanje na jednom mestu, deljenjem jednog slova. Po sadržaju ih možemo podeliti na klasične, tematske, simetrične, kriptične i tako dalje, a po strukturi na pravilne (pravougaone) i one koje su nepravilnog oblika.

Osmosmerka (eng. *words search*) – Jednostavna ali efikasna igra koja pruža zadovoljstvo svim uzrastima. Koristi se u edukativne svrhe kako bi pomogla razvijanju jezičkih veština, proširivanju rečnika i unapređenju sposobnosti koncentracije. Za razliku od prethodne dve igre, gde se u prazna polja unose cifre i slova, postavka osmostazne predstavlja već popunjenu mrežu slova. Cilj igre je da igrači pažljivo pregledaju slova i pronađu sve

zadate reči koje su unutar nje skrivene. Reči mogu biti, kako i sam naziv igre govori, napisane u svih osam smerova.

3. JEZICI SPECIFIČNI ZA DOMEN

3.1. Definicija

Jezici specifični za domen su namenjeni ograničenoj oblasti primene. Sastoje se od pojmova koji su relevantni za određeni domen, omogućavajući konciznije i razumljivije opise rešenja za te probleme. Ključna prednost JSD je što omogućava stručnjacima iz određene oblasti da lakše razumeju i učestvuju u razvoju rešenja, čak i ako nisu programeri. Sintaksa ovih jezika je često prilagođena konkretnim konceptima iz domena, što olakšava komunikaciju između inženjera i stručnjaka iz tog područja. JSD su poznati kao "mali jezici" (eng. *little languages*), a uspešnim se smatra onaj koji je fokusiran na uzak, dobro definisan domen i efikasno ga pokriva.

3.2. Prednosti i mane

Ukoliko je JSD dobro definisan, prednosti koje pruža su značajne [5]: uticaj na produktivnost programera, uticaj na kvalitet softvera, duži životni vek aplikacije i ovi jezici su samodokumentujući.

Nasuprot svim prednostima, postoje i izazovi na koje se nailazi pri korišćenju JSD-a, kao što su [6]: cena razvoja, dizajn jezika, održavanje i evolucija, zavisnost od alata i takozvani JSD pakao (nagomilavanje novih umesto korišćenja ili proširivanja već postojećih jezika).

3.3. Faze razvoja i elementi

Prilikom izgradnje svaki proizvod prolazi kroz određene faze razvoja. Proces razvoja jezika specifičnog za domen, prema knjizi "Domain-Specific Languages" autora Martina Faulera, može se opisati sledećim fazama [5]: istraživanje domena, identifikacija ključnih apstrakcija, dizajn jezičkih apstrakcija, implementacija jezika, testiranje i iterativni razvoj, održavanje i podrška.

Svaki softverski jezik, bez obzira da li je opšte namene ili specifičan za domen sastoji se iz apstraktne sintakse, jedne ili više konkretne sintakse i semantike [7].

Apstraktna sintaksa predstavlja strukturu podataka tipa stablo koje sadrži koncepte domena, njihove osobine i međusobne relacije bez detalja notacije. Definiše strukturu i određuje pravila validnosti jezičkih iskaza. Apstraktno sintakso stablo je usmereno, labelirano stablo gde čvorovi predstavljaju instance koncepta apstraktne sintakse. Jezici za definisanje apstraktnih sintaksi jezika se u domenu modelovanja nazivaju meta-metamodeli, odnosno meta-metamodel je apstraktna sintaksa takvog jezika.

Konkretna sintaksa predstavlja vizuelnu reprezentaciju jezika putem koje korisnik obavlja interakciju sa računarom, odnosno može se reći da je interfejs između jezika i korisnika. Definiše izgled iskaza na nekom jeziku i može biti tekstualna, grafička, tabelarna, tipa stabla, hibridna itd. Za jedan jezik dovoljna je jedna konkretna sintaksa ali može postojati i više njih.

Dok je konkretna sintaksa interfejs jezika ka korisniku, apstraktna se ponaša kao API za pristup programima od strane alata za obradu - validatori, transformacije i generatori koda.

Semantika je poslednji element nekog jezika i ona određuje smisao napisanih jezičkih iskaza. Definisanje semantike može se vršiti pomoću više tehnika, ali je u praksi najzastupljenije prevođenje na jezik koji već ima definisanu semantiku putem prevodioca ili interpretira. Prevođenje se obavlja kompajliranjem, odnosno generisanjem koda najčešće na jezike opšte namene i može da ide sve do nivoa mašinskog jezika definisanog u hardveru računara.

3.4. Model i meta-model

Model i meta-model su ključni koncepti u razvoju jezika specifičnih za domen (JSD) i njihovoj implementaciji. Model se često definiše kao opis ili reprezentacija sistema, dok semantički model pruža specifikaciju pojmova unutar jezika, nezavisno od same sintakse jezika.

Meta-model, s druge strane, predstavlja formalni opis elemenata modela i načina njihove povezanosti. U kontekstu jezika specifičnih za domen, meta-model opisuje pojmove iz određenog domena, njihova ograničenja i relacije među njima.

Često se meta-model koristi za definisanje strukture samog jezika, a termin "apstraktna sintaksa" se koristi kao sinonim za meta-model, suprotstavljajući se konkretnoj sintaksi koja opisuje specifičnu reprezentaciju jezika (tekstualnu, grafičku, tabelarnu, itd.).

Formalno definisan meta-model, zajedno sa definicijom konkretne sintakse, omogućava automatsko generisanje parsera ili editora jezika specifičnih za domen.

3.5. Parser

Parseri predstavljaju suštinske komponente u razvoju programskih jezika i interpretatora. Oni igraju ključnu ulogu u obradi ulaznih nizova znakova i simbola u strukture koje su pogodnije za dalju analizu i izvršavanje. Parseri se koriste za konverziju ulaznog teksta u apstraktne strukture podataka, kao što su Sintakso stablo (CST) ili Stablo apstraktne sintakse (AST), na osnovu formalno definisanih gramatika programskog jezika.

Proces parsiranja obično se sastoji iz tri osnovna koraka: leksička analiza (ulazni tekst se razlaže na tokene ili lekseme), sintakсна analiza (proverava i validira redosled tokena u skladu sa definisanom gramatikom i kreira stablo parsiranja) i semantička analiza (dodeljuje značenje parsiranom ulazu, prevodeći ga u oblik razumljiv računaru i izvršavajući semantičke akcije nad čvorovima stabla parsiranja) [8].

4. TEXTX

4.1. Definicija

TextX je alat koji služi za definisanje JSD-a u Python okruženju. Omogućava korisnicima da definišu svoje tekstualne jezike na jednostavan način, čime mogu da kreiraju potpuno nove ili podrže već postojeće tekstualne jezike ili formate datoteka.

Ključne funkcionalnosti textX-a uključuju: automatsko generisanje meta-modela i parsera na osnovu jednog opisa gramatike, automatsko povezivanje referenci između entiteta jezika (engl. *linking*), održavanje odnosa roditelja u modelu, kontrolu parsera u pogledu obrade razmaka i ključnih reči, mogućnost postprocesiranja modela,

podršku za modularizaciju gramatike, kombinovanje više meta-modela (engl. *multi-meta-model*), vizualizaciju kako meta-modela tako i parsiranih, modela radi lakšeg razumevanja i analize strukture jezika [9].

4.2. Gramatika

Sintaksa jezika i meta-model definisani su textX gramatikom. Gramatika predstavlja formalan sistem za opisivanje sintakse ili jezičke strukture, odnosno opisuje pravila koja određuju kako se ispravne rečenice ili izrazi mogu formirati iz komponenti jezika.

Osnovni elementi gramatike uključuju terminalne simbole (osnovne jedinice jezika, koje ne mogu biti dalje razložene) i neterminalne simbole (razvijaju se u skladu sa pravilima gramatike).

U kontekstu računarskih nauka, gramatika opisuje kako se tokeni (simboli programskog jezika) mogu kombinovati da bi se dobili validni programski izrazi ili naredbe. Često se predstavlja u Bakus-Naurovoj formi (BNF) ili sličnim notacijama koje opisuju pravila za razvijanje neterminala u termine.

U slučaju textX-a, koristi se proširena Bakus-Naurova forma (*Extended Backus-Naur Form - EBNF*) gde je moguće navesti pravilo u više redova.

Osim toga, EBNF uključuje mehanizme za unapređenje, definisanje broja ponavljanja, isključivanje alternativa, komentare itd.

Sledećih pet pravila čine osnovu textX tipova i automatski se konvertuju u Python tipove pri instanciranju objekata: ID, INT, FLOAT, BOOL, STRING.

Pored njih podržani su: pravilo prepoznavanja (string ili regex), sekvence (omogućavaju ulančane izraze kojima se precizira očekivani redosled), uređeni izbor, opcioni izraz, ponavljanja (nula ili više, jedan ili više, grupe proizvoljnog redosleda), reference na druga pravila, dodele, sintatički predikati (pozitivni & i negativni !) i drugi.

5. OPIS REŠENJA

5.1. Jezik i gramatika

Nakon istraživanja oblasti jezika specifičnih za domen i textX-a kreiran je jezik za enigmatiku sa ekstenzijom *.lg* (*logic game*). Gramatika jezika započinje *Enigma* pravilom (slika 5.1.) koje služi da objedini sve tipove enigmi u jedan objekat.

```
Enigma:
|   details=EnigmaType
;

EnigmaType:
|   Sudoku | Crossword | WordSearch
;
```

Slika 5.1. Pravila *Enigma* i *EnigmaType*

Pravilo je jednostavno, sastoji se od polja *details* kojem se dodeljuje pravilo *EnigmaType* - uređeni izbor pravila i navodi da polje detalji mora biti definisano nekim od sledećih pravila: *Sudoku*, *Crosswords* ili *WordSearch*. Na slici 5.2. prikazan je primer gramatike za *WordSearch* entitet.

```
WordSearch:
' ' type='wordsearch'
auto_generate?='auto_generate'
('dimensions' ':' dimensions=Dimension)?
'content' ':' content=WordSearchContent
;

WordSearchContent:
('letters' ':' letters+=MatrixRow+)?
'words' ':' words+=STRING (' ' words+=STRING)*
;

MatrixRow:
'R' ':' row+=LETTER+ ','
;

LETTER:
/[a-zA-Z]/
;

Dimension:
INT 'x' INT
;
```

Slika 5.2. Primer gramatike za entitet *WordSearch*

5.2. Parser

Pomoću gramatike koja se prosleđuje funkciji *metamodel_from_file* iz Python paketa *textx.metamodel* kreira se meta-model ukoliko gramatika nema grešaka, u suprotnom se vraća izuzetak. Na osnovu meta-modela, textX može da izgradi parser koji čita i interpretira svaki deo ulaznog teksta u saglasnosti sa specifikacijama jezika. Takođe, na ovom nivou registruju se procesori koji će izvršiti dodatne validacije i modifikacije nad pročitanim objektima. U sklopu ovog rešenja korišćen je procesor objekata koji se poziva u paraleli sa parsiranjem nad svakim uspešno pročitanim objektom. Koristeći opcije procesora, izvršavaju se validacije nad sudokvom i osmosmerkom koje nije izvodljivo obaviti na nivou gramatike: provera da li sudoku ima 81 karakter, da li je inicijalno otkriveno minimum 17 cifara kako bi sudoku bio rešiv, ispitivanje da li postoje ponovljene cifre u istom redu, koloni ili kvadratu, provera da li je u sklopu osmosmerke izabrana opcija *auto_generate* (generisanje mreže slova od strane sistema) i ukoliko jeste da li su unete dimenzije mreže.

5.3. Server

Server obezbeđuje infrastrukturu za dalju obradu i slanje podataka na korisnički interfejs. Razvijen je koristeći Python jezik i njemu kompatibilne Uvicorn veb server i FastAPI radno okruženje.

FastAPI pruža moderan okvir za razvoj API-ja, baziran na korišćenju i unapređenju mogućnosti tipiziranih nago-veštaja koje su uvedene u novijim verzijama Python-a, počevši od verzije 3.5 [10]. Ovim je omogućeno defini-sanje, validacija, automatska serializacija i deserijaliza-cija, kao i korišćenje tipova podataka u svojim operacija-ma API-ja. Odlikuje se brzinom, efikasnošću i jedno-stavnim definisanjem veb ruta i operacija korišćenjem Python dekoratora.

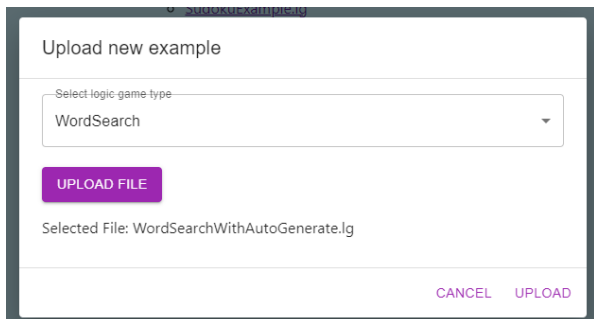
Uvicorn je moćan veb server posebno dizajniran za Python aplikacije koje koriste asinhronu obradu i bazirane su na ASGI (*async server gateway interface*) specifikaciji [11]. Izuzetno je brz i efikasan, što ga čini idealnim izborom za pokretanje modernih veb aplikacija. Podržava asinhrono izvršavanje koda i više radnih procesa. Funkcioniše na bazi osluškivanja soketa, zatim prima konekciju, procesuiru je i prosleđuje.

5.4. Klijentska aplikacija

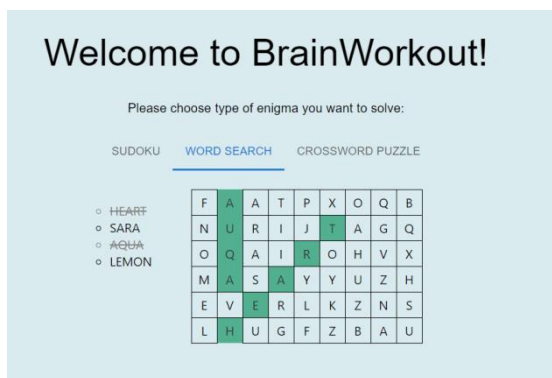
Klijentska aplikacija kreirana je u obliku SPA aplikacije pomoću React-a i predstavlja alat putem kog korisnici mogu da uvezu svoje primere enigmatskih igara (slika

5.3.), a zatim njih ili primere koje su drugi uvezli, rešavaju (slika 5.4.).

React je biblioteka za razvoj korisničkog interfejsa (engl. UI) u JavaScript programskom jeziku. Dizajnirana je za izgradnju efikasnih, lakih za održavanje i visoko performantnih veb aplikacija. Koristi deklarativni pristup za opisivanje komponenti koje se mogu renderovati u odgovoru na promene stanja aplikacije.



Slika 5.3. Dijalog za uvoz novih primera



Slika 5.4. Primer igre – sudoku

6. ZAKLJUČAK

Primenom jezika specifičnog za domen na datom primeru dokazana je jedna od osnovnih prednosti ovih jezika, a to je prilagodljivost korisnicima koji ne poseduju prethodno tehničko znanje. Uočeni nedostaci postojećih rešenja bili su motivacija za razvoj sažete i jednostavne sintakse, pogodne za upotrebu osobama iz različitih sfera. Čak i kao takvo, rešenje ostavlja dosta prostora za unapređenje i dalji razvoj. Neke od ideja za dalji razvoj bi mogle biti:

- Podrška za veći broj igara, odnosno proširivanje jezika da obezbedi generisanje većeg opsega enigmatskih igara.
- Omogućiti dodatne tipove već podržanih igara: naprednije verzije sudokua, ukrštene reči nepravilnog oblika, ukrštene reči sa zagonetnom rečju ili tematske osmo-stazne.
- Unaprediti interakciju krajnjeg korisnika sa korisničkim interfejsom, na primer korišćenje biblioteke
- OpenLayers koja bi vizuelnim i interaktivnim elementima omogućila bolje korisničko iskustvo.
- Implementacija baze podataka omogućila bi značajnu optimizaciju rešenja.

- Nadogradnja sistema tako da podrži registraciju korisnika, praćenje njihovih rezultata i uvođenje vremenskog ograničenja.

- Uvođenje multiplayer režima gde bi igrači mogli da rešavaju enigme zajedno ili se takmiče jedni protiv drugih u sklopu određenih turnira.

- Integracija sa određenim softverima koji bi omogućili obrazovni aspekt kao što su logičko razmišljanje, matematički koncepti ili drugi obrazovni ciljevi.

7. LITERATURA

- [1] Wikipedia – enigmatika, <https://sr.wikipedia.org/sr/Енигматика>, na mreži, pristup: april 2024.
- [2] Rationality - Constraints and Contexts, T.-W. Hung and T.J. Lane , Academic Press, 2016
- [3] Sudoku Space, <https://sudoku-space.com>, na mreži, pristup: april 2024.
- [4] Crossword Puzzles as a Learning Tool For Vocabulary Development, Wiwat Orawiwanakul, Language Institute, Bangkok University, 2017
- [5] M. Fowler, Domain-specific languages, Addison-Wesley, 2010
- [6] M. Völter, „DSL Engineering: Designing, Implementing and Using Domain-Specific Languages”, 2013.
- [7] I.Dejanović – Jezici specifični za domen, - „Uvod“ <https://www.igordejanovic.net/courses/jsd/01-uvod/>, na mreži, pristup: april 2024.
- [8] I.Dejanović – Jezici specifični za domen – „Tekstualne sintakse“, <https://www.igordejanovic.net/courses/jsd/03-tekstualne-sintakse/>, na mreži, pristup: april 2024.
- [9] I.Dejanović, R. Vadera, G. Milosavljević, Ž. Vuković, „TextX: A Python tool for Domain-Specific Languages implementation”, Knowledge-Based Systems, 2017
- [10] FastAPI, <https://fastapi.tiangolo.com/>, na mreži, pristup: april 2024
- [11] Uvicorn, <https://www.uvicorn.org/>, na mreži, pristup: april 2024

Kratka biografija:

Sara Panić rođena je 21. marta 1997. godine u Štutgartu, u Nemačkoj. Osnovnu školu "Vuk Karadžić" završila je sa Vukovom diplomom, u Zrenjaninu 2012. godine. Iste godine upisuje, a 2016. završava opšti smer u "Zrenjaninskoj gimnaziji". Na jesen postaje student Fakulteta tehničkih nauka na Univerzitetu u Novom Sadu, smer Primenjeno softversko inženjerstvo. Diplomirala je u septembru 2020. godine nakon čega obrazovanje nastavlja na master akademskim studijama Primenjene računarske nauke i informatika, smer Softversko inženjerstvo. Paralelno sa pohađanjem master studija, od marta 2021. godine, godinu dana radila je kao Asistent - Saradnik u nastavi na osnovnim studijama Primenjeno softversko inženjerstvo.

**РАЗВОЈ МОДУЛА ЗА ИНТЕГРАЦИЈУ ПРОЦЕСОРА У КОНФИГУРАБИЛНОЈ
HALOWONE™ WI-FI IP ПЛАТФОРМИ****DEVELOPMENT OF THE PROCESSOR INTEGRATION MODULE IN THE
CONFIGURABLE HALOWONE™ WI-FI IP PLATFORM**

Душан Радишић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У овом раду приказан је развој омотача (енг. wrapper) процесора који обухвата и *xTensa* и *RISC-V CV32E40P* са акцентом на омотачу који обухвата *CV32E40P*. Рад описује све кораке потребне да би се овакав систем развио, од почетних истраживања архитектура и функционалности различитих процесора до интеграције процесорског омотача у *HaLowOne™* платформу.

Кључне речи: Wi-Fi HaLow, RISC-V, процесор, станица, приступна тачка

Abstract – This paper represents the development of the processor wrapper that encapsulates both *xTensa* and *RISC-V CV32e40p* with emphasis on the wrapper with the *CV32E40P* processor. The paper describes all the steps required to develop such a system, from the initial research of architectures and functionalities of different processors to the integration of the processor wrapper into the *HaLowOne™* platform.

Keywords: Wi-Fi HaLow, 802.11ah, RISC-V, processor, Station, Access point

1. УВОД

У домену бежичних комуникационих технологија, тежња за сталним напретком у циљу побољшања перформанси и ефикасности система остаје веома важна. *Wi-Fi HaLow™*, интегрални аспект *Wi-Fi* стандарда, нуди ниску потрошњу енергије и широк опсег преноса података, што захтева адаптације унутар постојећих хардверских архитектура, посебно у дизајну система на чипу (енг. *System on chip- SoC*) како би се интегрисала подршка за ову технологију беспрекорно [1].

Wi-Fi CERTIFIED HaLow™, ознака за сертификоване производе који укључују технологију *IEEE 802.11ah*, повећава *Wi-Fi* радећи у спектру испод 1 GHz како би понудио дужи домет и нижу снагу повезивања. *Wi-Fi HaLow™* испуњава јединствене захтеве за Интернет ствари (*IoT*) како би омогућио различите случајеве употребе у индустријским, пољопривредним, паметним зградама и паметним градским срединама [1].

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Вук Врањковић, ванр. проф.

Како овај рад буде напредовао, биће спроведена свеобухватна анализа карактеристика *Wi-Fi HaLow™* технологије, пружајући кључни увид у њену оперативну динамику. Поред тога, биће спроведено детаљно испитивање архитектура *xTensa* и *RISC-V* процесора у контексту *MAC* функционалности, разјашњавајући њихове снаге и компатибилност. Паралелно, рад ће се бавити сложеним стратегијама имплементације за универзални омотач процесора, истражујући начине за оптимизацију његове ефикасности и свестраности. Коначно, перформансе предложеног решења биће процењене комбинацијом симулација и емпиријског експериментисања, пружајући увид заснован на подацима у његову применљивост и делотворност у стварном свету.

Један од главних циљева омотача јесте лако пребацивање *MAC Wi-Fi* процесора између *xTensa* и *RISC-V* језгра.

Постојећа *HaLowOne™* платформа, која је свеобухватна, високо конфигурабилна *Wi-Fi HaLow™ IP* платформа која се састоји од *Wi-Fi HaLow™* модема и опционог интерног подсистема *host-a*, намењена је да буде погодна за надоградњу на следећу генерацију.

Један од главних задатака који треба остварити јесте креирање омотача процесора како би прелаз између два или више процесора био што лакши, као и коришћење истог омотача не само као *MAC* процесора већ и такође опционо као интерни систем *host-a*. То значи омогућавање инстанцирања више процесорских омотача у вишепроцесорском систему са истим или различитим параметрима.

Иако постизање транзиције процесора без додатних системских промена није могуће, захтеви су дизајнирани тако да исход пројекта буде што је могуће ближе идеалном сценарију. Циљ је минималне промене у систему и начину на који окружење омотача процесора комуницира са њим.

2. ЗАХТЕВИ

- **32/64-битни меморијски интерфејс** – Меморијски интерфејс између омотача и меморије треба да буде параметризован за ширину података. Једна од највећих разлика између наших циљних процесора је ширина приступа меморији – *xTensa* је 64-битна, док је *CV32E40P* 32-битна. Ширина линије података остаје параметар у омотачу - 32 бита ако се користи *CV32E40P* или 64 бита ако се користи *xTensa*.

- **Програмски интерфејс** – *AXI-full* интерфејс служи за програмирање *MAC* процесора. У систему који користи интерни хост, он је повезан са омотачем преко *AXI* инфраструктуре као *AXI master*.

- **32 линије за прекиде** - 32 линије прекида сигнала директно повезане са улазом за прекиде на процесору или преко контролера прекида. Овај улаз омогућава пренос сигнала прекида од система до процесора, што је кључно за одређивање функционалности процесора као и читавог система.

- **Механизам за тражење грешака (*debug*)** - Омотач треба да има интерфејс за отклањање грешака. У овом пројекту, *JTAG* порт је изабран као стандардни порт за отклањање грешака.

- **Test enable** - Омогућава тестирање система.

- **AXI Master порт** - Неопходно је имати *AXI master* на који је повезан процесор. Ово омогућава процесору да пише и чита из других делова система. Повезује се директно на *AXI-interconnect*, који на основу адресе инструкција одређује где треба да буде рутиран.

- **Self clock gating верификација** - Пошто *CV32E40P* процесор има јединицу за режим спавања уграђену у њега, неопходно је верификовати да ли извршава своју функционалност исправно. Битно је проучити како функционише и под којим условима процесор прелази у режим спавања.

- **Конфигурабилна архитектура дизајна**

- **Меморијска мапа** - Мора бити добро дефинисана, и границе сваког блока морају бити добро дефинисане и верификоване, укључујући и граничне случајеве.

3. ЦИЉЕВИ

- Лака промена *Wi-Fi MAC* процесора између *xTensa* и *RISC-V* језгра на *HaLowOne™* платформи.

- Специфицирање јединственог *омотача* који одговара за интегрисање и *xTensa* и *RISC-V* процесора у *HaLowOne™* платформу.

- Дизајнирање и имплементација омотача за *HaLowOne™* платформу.

- Демонстрација омотача у *HaLowOne™* платформи са *xTensa* процесорским језгром.

- Демонстрирати омотач у подсистему направљеном са *RISC-V CV32E40P* језгром.

- Коришћење и валидација напредних алата за пројектовање и верификацију дигиталних система – *Sigasi*.

4. ОМОТАЧ

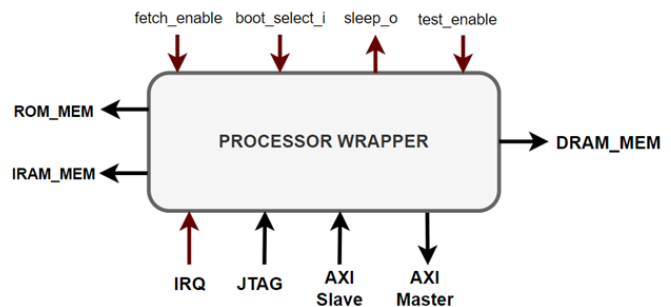
Истраживање је спроведено на више *RISC-V* процесора, а фокус је стављен на *CV32E40P* и *xTensa* процесоре. Главни задатак овог пројекта био је креирање омотача тако да се миграција са *xTensa* у *RISC-V* може урадити са најмање могуће застоја и грешака. Два процесора су веома различита, због чега је омотач креиран да ради на вишем нивоу хијерархије, омогућавајући систему да комуницира и интерагује са њим беспрекорно, без обзира да ли је *RISC-V* или *xTensa* инстанцирана у њему.

Развој јединственог скупа интерфејса омотача најкомпатибилнији је са *xTensa* процесором. Ово је урађено

са циљем да се постојећа платформа (која је у функцији) брзо замени омотачем конфигурираним са *xTensa* процесором, и на тај начин демонстрира да тестови пролазе успешно као када је инстанцирана само *xTensa* (без омотача).

Након завршених испитивања, закључено је да платформа са омотачем, када је у питању *xTensa* процесор, функционише исто као и без омотача. Закључујемо да омотач као концепт има смисла, и да се може даље наставити са радом на њему.

Следећи део пројекта укључује интеграцију *RISC-V CV32E40P* процесора у омотач и проверу да и омотач и процесор ефикасно обављају своју функционалност.



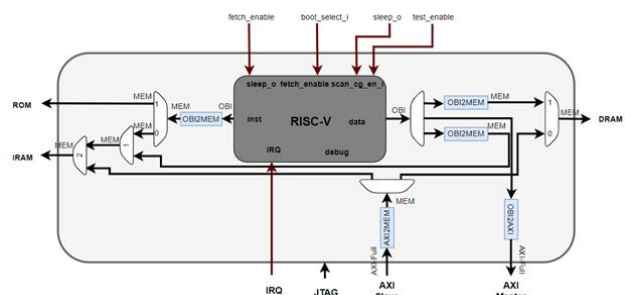
Слика 1. Top „Processor wrapper“

Сваки од сигнала/интерфејса приказаних на слици 1. има своју улогу у комуникацији околног система са омотачем.

Овај омотач има две имплементације, једну са *xTensa* процесором и другу са *CV32E40P*. Као што је раније поменуто, при одабиру интерфејса омотача, фокус је првенствено био на интеграцији *xTensa* процесора у омотач и уграђивању таквог система (процесорски омотач са *xTensa* унутра) у постојећу *HaLowOne™* платформу која већ постоји.

Разлог за то јесте што је познато да платформа ради са *xTensa* процесором и да је овај процесор у потпуности верификован, као и платформа.

У овој фази добијен доказ концепта. Сви тестови су прошли успешно, као да је инстанцирана сама *xTensa* без процесора. То нам омогућава да пређемо на следећи корак, а то је интеграција *RISC-V* процесора у омотач (слика 2.).



Слика 2. Процесорски омотач са *OpenHW RISC-V CV32E40P*

На слици 2. приказане су следеће компоненте:

- *MEM* мултиплексер
- *OBI* мултиплексер

- MEM адресни декодер
- OBtoMEM конвертор
- AXItoMEM конвертор
- Машина стања (FSM)

Након дизајнирања процесора и верификације његових основних функционалности, извршена је синтеза и имплементација. Vivado је генерисао извештај у коме су наведени сви ресурси који се користе инстанцирањем једног процесорског омотача са CV32E40P. Информације о заузећу ресурса приказане су у табели 1. Синтеза и имплементација извршени су на ADRV9361-Z7035 плочи.

Табела 1. Заузеће ресурса процесорског омотача

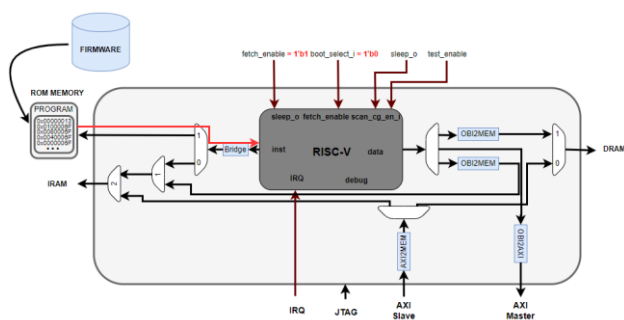
	Slice LUTs	Slice Registers	F7 Muxes	F8 Muxes	Slice	LUT as Logic	LUT as Memory	Block RAM Tile	DSPs
Available resources	171900	343800	109300	54650	54650	171900	70400	500	900
Utilization by the Processor wrapper	4322	2110	281	0	1486	4322	0	0	5
Utilization by the CV32E40P	4114	2089	281	0	1440	4114	0	0	5
CV32E40P percentage of Processor wrapper utilization	95.18%	99.00%	100%	100%	96.90%	95.19%	100%	100%	100%

Из табеле може да се закључи да процесорски омотач заузима веома малу количину ресурса.

5. ПОКРЕТАЊЕ (енг. BOOTING) ПРОЦЕСОРА

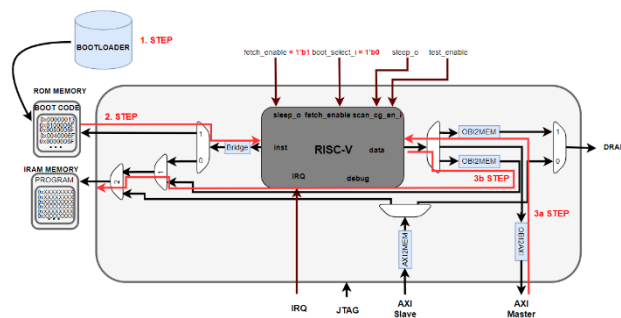
Покретање представља уписивање фирмвера који треба да буде извршен у системској меморији и почетно иницијализовање података у DRAM-у који треба почетно бити присутан тамо. Објашњене су три методе покретања.

Покретање из ROM меморије - фирмвер је већ уписан у ROM меморију процесора, и извршење програма је спремно. Сlike 3. 4. 5. 6. ближе објашњавају процесе покретања.

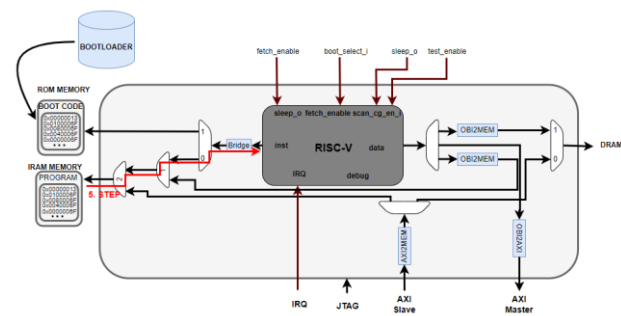


Слика 3. Покретање из интерне ROM меморије

Покретање преко AXI-Master порта (процесор сам себе покреће) - Фирмвер је смештен у меморији изван система процесора. Додељен је одређени простор у меморији, и процесор може приступити њему преко свог AXI master порта и AXI interconnect-a. Ова меморија може бити системска меморија која се налази унутар платформе или спољна flash меморија која се приступа преко периферија.

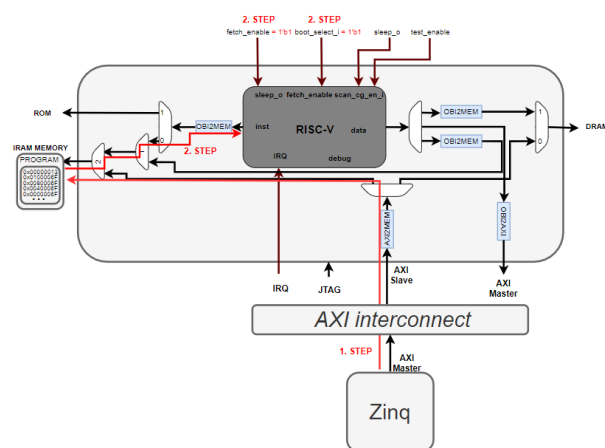


Слика 4. Покретање из екстерне меморије, кораки 1, 2, 3 и 4



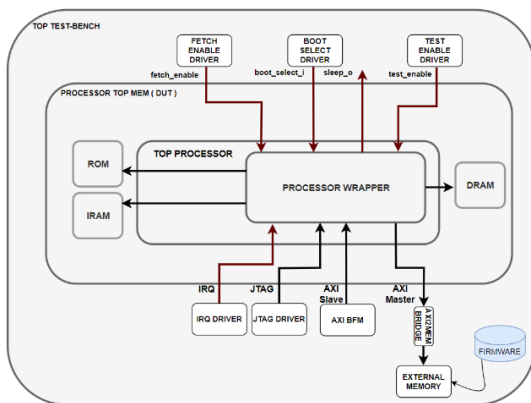
Слика 5. Покретање из екстерне меморије, корак 5

Покретање преко AXI-Slave порта – Приликом овог начина покретања процесор не контролише учитавање програма, па спољни контролер система мора обезбедити да се све инструкције упишу у IRAM пре покретања преузимања инструкција. Предност је што је лако ажурирати или променити фирмвер који процесор треба да изврши.



Слика 6. Покретање преко AXI-Slave-a

Сви начини покретања процесора су верификовани, као и меморијска мапа. Направљено је верификационо окружење и пуштени су тестови како би се осигурала добра функционалност омотача. Блок дијаграм верификационог окружења приказан је на слици 7.



Слика 7. Top testbench wrapper-a

6. МЕМОРИЈСКА МАПА

Да би се осигурала максимална компатибилност и лакоћа интеграције, било је потребно прилагодити меморијску мапу тако да одговара меморијској мапи *xTensa* процесора као да је интегрисан у омотач.

Из перспективе дигиталног дизајна, меморијска мапа је дефинисана као параметар унутар декодера адреса и демултиплексера. Зависно од адресе трансакције, трансакција се рутира ка једном од излаза демултиплексера. Постоје одређени опсези где се трансакција прослеђује даље. Ако трансакција промаши опсег, враћа се "*default response*" како би се обезбедило да процесор добије неки одговор. Ако процесор затражи адресу која не постоји, то указује на непотпуну меморијску мапу, што изазива прекид рада процесора.

Меморијска мапа је значајна за три порта у претходно приказаном омотачу. То су:

- CV32E40P порт за податке (OBI протокол)
- CV32E40P инструкциони порт (OBI протокол)
- AXI-Slave порт

7. ИМПЛЕМЕНТАЦИЈА И ИНТЕГРАЦИЈА МОДЕМА НА FPGA ПЛОЧИ

Bitstream добијен из процеса синтезе успешно је учитан на *FPGA*, и тестирање је започето како би се осигурало да резултати одговарају онима добијеним током верификације. Очекивани исход теста је приказивање на терминалу. Исправно извршавање дела програма везаног за *UART* указује на то да процесор исправно функционише.

Постоји комуникација преко *UART*-а што значи да је тест успешно извршен, и порука је приказана на терминалу, што указује да интегрисани омотач са *RISC-V* процесором на *Methods2Business HaLowOne™* платформи исправно врши своју функционалност. Овај корак служи као доказ концепта (енгл. *proof of concept*).

8. ЗАКЉУЧАК

Рад је почео проучавањем процесора, где је био потребан детаљнији увид у његове компоненте, унутрашњост и функционисање. Након тога, изазов је био написати цео омотач, поново користити постојеће компоненте и креирати нове по потреби како би се осигурала добра функционалност. После тога, провера свих путања и отклањање грешака у мом коду били су кључни кораци. Коначно, главни део је био постављање на плочу, где сам се даље упознао са скриптама за синтезу и имплементацију.

Током писања *SystemVerilog* кода, многи алати и ресурси су били од помоћи, посебно *Sigasi* - напредни алат за пројектовање и верификацију дигиталних система. Уз помоћ овог алата писање *HDL* кода је постало лакше, праћење компоненти, хијерархије, а цео процес дизајнирања је значајно убрзан.

Закључак рада јесте да смо показали да *RISC-V* процесор може да ради без грешке унутар *HaLowOne™* платформе. Добро се интегрише и исправно обавља своје функције. Додатно, поред тога, интегрисан и тестиран је још један начин отклањања грешака у процесору, коришћењем порука послатих преко *UART*-а.

Процесорски омотач, у садашњем стању, може послужити као основа за друге *RISC-V* процесоре. На пример, истраживање је показало да би интеграција *SCR1* и *Ibex RISC-V* процесора у створени омотач захтевала мање времена, што је био један од циљева самог рада.

4. LITERATURA

- [1] <https://www.wi-fi.org/discover-wi-fi/wi-fi-certified-halow> (приступљено у фебруару 2024.).

Kratka biografija:

Душан Радишић рођен је у Новом Саду 1999. год. Дипломски рад на Факултету техничких наука из области Електротехнике и рачунарства – Ембедед системи и алгоритми одбрао је 2022. године.
контакт: dusanr14@gmail.com

IMPLEMENTACIJA MIKROSERVISNE ARHITEKTURE KROZ VEB APLIKACIJU ZA PRODAJU AUTOMOBILA**IMPLEMENTING A MICROSERVICE ARCHITECTURE THROUGH A CAR SALES WEB APPLICATION**Vladimir Vrbica, Milan Vidaković, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – Ovaj rad se bavi realizacijom aplikacije za prodaju automobila upotrebom mikroservisne arhitekture. Ova aplikacija je prvobitno implementirana upotrebom monolitne arhitekture. Klijentski deo aplikacije je razvijen uz pomoć Angular alata, uz dodatak Bootstrap biblioteke. Serverski deo aplikacije je razvijen u programskom jeziku Java, korišćenjem Spring okruženja. Aplikacija je deploy-ovana putem Docker alata, dok je u aplikaciju integrisan i Stripe sistem za plaćanje.

Ključne reči: Java, ActiveMQ, Docker, Rest, Stripe, mikroservisi

Abstract – This paper demonstrates a microservice architecture that was implemented through a car sales web application. This application was originally created as a monolithic one. The client part of the application was developed using Angular tools, with the addition of the Bootstrap library. The server part of the application was developed in the Java programming language, using the Spring environment. The application is deployed using the Docker tool, while the Stripe payment system is also integrated into the application.

Keywords: Java, ActiveMQ, Docker, Rest, Stripe, microservices

1. UVOD

U ovom radu biće opisana implementacija mikroservisne arhitekture [1] kroz veb aplikaciju za prodaju automobila. Prvobitna monolitna aplikacija je unapređena u mikroservisnu tako što je implementirana po uzoru na sisteme koji će biti predstavljeni u drugom odeljku. Kada je reč o aplikaciji, klijentski deo je Angular aplikacija, dok je serverski deo Java aplikacija u Spring okruženju.

2. PREGLED SLIČNIH SISTEMA

Inspiracija za odabir ovakvog tipa aplikacije pronađena je u sajtovima Polovni automobili [2] i Mobile.de [3]. Sajtovi su međusobno slični i predstavljaju najpopularnije i najposećenije sajtove za prodaju automobila. Imajući u vidu sličnost, jednako se prikazuju sve neophodne usluge i informacije na jednom sajtu za prodaju automobila.

NAPOMENA:

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

Neke od tih usluga jesu oglašavanje automobila za prodaju putem oglasa, pretraživanje oglasa, uvid u prethodna iskustva drugih kupaca, postupak registracije vozila i slično.

3. KORIŠĆENE TEHNOLOGIJE

Za implementaciju klijentskog dela aplikacije korišćen je Angular [4] alat uz dodatak Bootstrap [5] biblioteke, dok je serverski deo aplikacije napisan u programskom jeziku Java [6], korišćenjem Spring [7] okruženja.

Spring

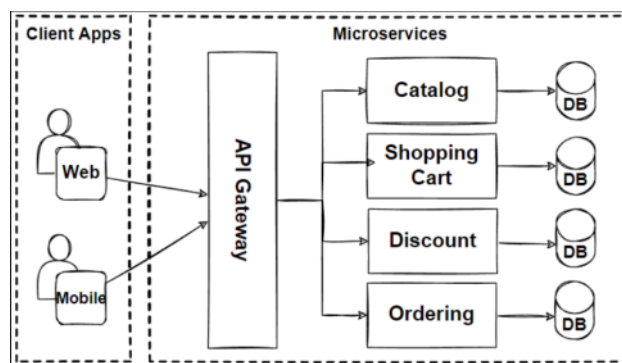
Spring je okruženje koje služi za jednostavniji razvoj poslovnih aplikacija u Javi. Trenutno je najpopularnije i najuticajnije okruženje za razvoj Java veb aplikacija. Nudi podršku za dependency injection mehanizam, veb aplikacije, rad sa podacima iz baze podataka, razmenu poruka itd. Uključuje REST arhitekturu, klijent-server arhitekturu koja koristi HTTP komunikacioni protokol i nudi svoje metode GET, POST, PUT, DELETE.

Angular

Angular je razvojna platforma bazirana na TypeScript-u. Uključuje okvir zasnovan na komponentama za izgradnju veb aplikacija koristeći HTML. Komponente definišu prikaze koji su zapravo skupovi elemenata koje Angular može da bira i menja u skladu sa programskom logikom.

Mikroservisna arhitektura

Mikroservisna arhitektura podrazumeva postojanje većeg broja manjih servisa koji su međusobno slabo povezani, slika 1.



Slika 1 – Mikroservisna arhitektura

Svaki servis ima svoju bazu podataka, svoje servise koji manipulišu tim podacima, kao i svoje kontrolere koji

predstavljaju pristupne tačke svakog servisa. Zahvaljujući tome, servisi mogu da se implementiraju kao nezavisne, manje aplikacije, koje mogu biti napisane različitim jezicima i u različitim okruženjima.

Docker

Docker [8] je otvorena platforma za razvoj i pokretanje aplikacija. Pruža mogućnost pakovanja aplikacije u labavo izolovanom okruženju koje se zove kontejner. Moguće je istovremeno pokrenuti više kontejnera na datom hostu. Svaki projekat ima svoj Dockerfile, dok se za pokretanje više projekata u zasebnim kontejnerima kreira docker-compose.yml fajl.

```
user:
  container_name: user
  build:
    context: UserApplication
    dockerfile: Dockerfile
  environment:
    - DB_URL=jdbc:postgresql://db1:5432/user-service
  service
    - DB_USERNAME=postgres
    - DB_PASSWORD=postgres
  ports:
    - "8081:8081"
  depends_on:
    db1:
      condition: service_started
      restart: true
```

Listing 2 – Primer kreiranja kontejnera user

Stripe

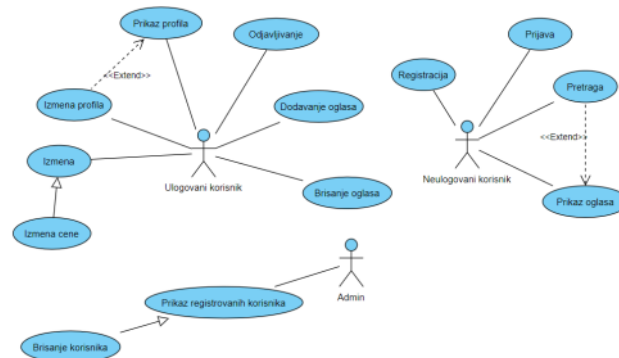
Stripe jeste provajder platnih usluga koji omogućava trgovcima da prihvate kreditne i debitne kartice ili druga plaćanja. Njegovo rešenje za obradu plaćanja je najpogodnije za preduzeća koja većinu svoje prodaje ostvaruju putem mreže, odnosno za kompanije koje se bave onlajn prodajom. Podržava plaćanje u različitim valutama i prihvata plaćanje različitim karticama.

4. SPECIFIKACIJA

Zadatak samog rada jeste refaktorisanje monolitne veb aplikacije za prodaju automobila, kreiranje više manjih servisa po uzoru na mikroservisnu arhitekturu. Između servisa uspostavljena je sinhrona (REST) odnosno asinhrona (ActiveMQ) komunikacija. Zahtevi koje klijent šalje ka serveru gađaju API Gateway, odakle se zahtevi prosleđuju odgovarajućim mikroservisima. Aplikacija je deploy-ovana korišćenjem Docker platforme. Svi servisi, zajedno sa odgovarajućim bazama podataka, smešteni su u kontejnere i organizovani u Docker okruženju. Uz aplikaciju je integrisana i Stripe platforma koja omogućava bezbedan i siguran vid plaćanja karticom putem interneta.

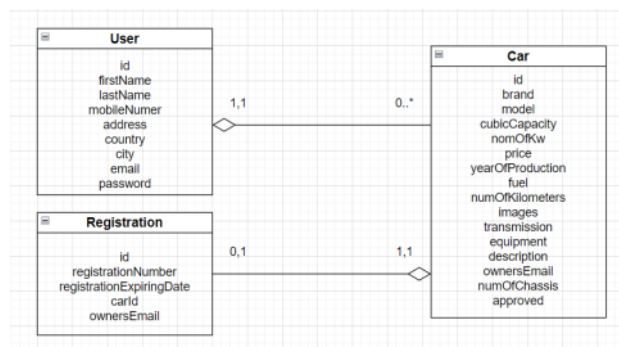
Aplikacija reprezentuje sajt za prodaju automobila. Korisnicima nudi mogućnosti oglašavanja automobila na prodaju, pretraživanja oglasa kao i ažuriranja postojećih oglasa. Na početnoj stranici je omogućeno višestruko

pretraživanje postojećih oglasa, kao i prijava odnosno registrovanje na sistem. Prijavljeni korisnik ima mogućnost objavljivanja novog oglasa. Kada popuni obavezna (marka, model, godina proizvodnje..) i opcionalna polja, oglas se šalje administratoru sistema koji može da odobri oglas, nakon čega će on biti vidljiv u sistemu, ili da ga odbije. Pored prikaza pretraživanih automobila, korisnik ima mogućnost pregleda profila svakog automobila, gde su prikazane osnovne informacije o automobilu, slike, opisi kao i podaci o kontaktu prodavca. Opisane usluge prikazane su na dijagramu slučajeva korišćenja prikazanom na slici 3.



Slika 3 – Dijagram slučajeva korišćenja

Sam domen aplikacije je veoma jednostavan i sastoji se iz tri klase koje su karakteristične za ovaj sistem, a to su Car, User i Registration. Kao što im samo ime kaže, Car klasa se odnosi na automobile i sadrži polja koja predstavljaju sve informacije koje se nalaze na jednom oglasu. Za nju je vezana klasa Registration koja reprezentuje registracionu oznaku automobila. Oznaka je vezana za samo jedan automobil, ali ona ne mora uvek da bude navedena. Poslednja, User klasa sadrži informacije o korisniku. Slika 4 prikazuje opisani dijagram klase.



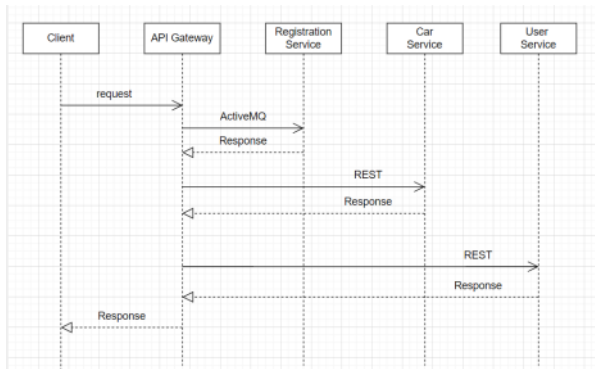
Slika 4 – Dijagram klase

Na slici 5 prikazan je dijagram sekvenci koji predstavlja arhitekturu aplikacije. Zahtev koji stiže sa klijentske strane, pogađa prvo API Gateway odakle se, analizom pristiglog zahteva, on prosleđuje ka odgovarajućem mikroservisu odakle se nakon obrade vraća odgovor.

5. IMPLEMENTACIJA

Aplikaciju čine dve celine, klijentski i serverski deo. Serverski deo implementiran je u programskom jeziku Java, uz pomoć Spring okruženja, dok je klijentski deo aplikacije implementiran pomoću Angular alata uz

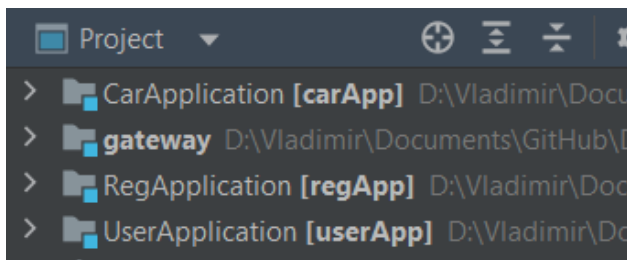
dodatak Bootstrap biblioteke. Serverski deo je mikroservisno orijentisan i deploy-ovan pomoću Docker alata. Takođe, uz aplikaciju je integrisan i sigurnosni sistem za plaćanje pod imenom Stripe.



Slika 5 – Dijagram sekvenci

Serverski deo

Serverski deo aplikacije, kao što je već napomenuto, je mikroservisno orijentisan i sastoji se iz četiri projekta, a to su Gateway, CarApplication, UserApplication i RegistrationApplication (slika 6). Aplikacije međusobno komuniciraju sinhronim (REST) odnosno asinhronim putem ActiveMQ.



Slika 6 – Struktura aplikacije

Zahtevi koji pristužu iz klijentskog dela aplikacije pogađaju Gateway kontroler, odakle se nakon obrade prosleđuju odgovarajućim servisima na dalju obradu. Kako je aplikacija deploy-ovana putem Docker-a, svaki projekat sadrži svoj Dockerfile u kom se navodi jasna konfiguracija projekta koja će se preneti na Docker image. Da bi se aplikacija uspešno pokrenula, sva četiri projekta su objedinjena unutar docker-compose.yml fajla uz pomoć kog se kreiraju kontejneri koji će pokrenuti svaki servis zasebno.

Svaki servis ima svoj kontroler, servis i repozitorijum, kao i svoju bazu podataka. Kontroleri predstavljaju pristupnu tačku svakog mikroservisa, oni primaju zahteve, obrađuju ih i prosleđuju servisima. Dalje, servisi komuniciraju sa repozitorijumima koji dobavljaju podatke ili ažuriraju bazu podataka.

Nakon toga, odgovor se vraća na istu lokaciju odakle je zahtev došao. Metode unutar kontrolera anotirane su, u zavisnosti od tipa zahteva koji obrađuju, sa @GetMapping, @PostMapping ili @PutMapping, gde se u zagradi navodi URL putanja na koju treba zahtev proslediti. Metodama, anotiranim pomenutim anotacijama, se parametri prosleđuju kroz putanju (@PathVariable), zahtev (@RequestBody) ili URL (@RequestParam).

Stripe server

Kao što je već pomenuto, uz aplikaciju je integrisan i sigurnosni sistem za plaćanje pod imenom Stripe. Podaci o kartici putem koje se vrši plaćanje, šalju se u okviru tokena sa klijentske strane ka serveru, kako bi se plaćanje evidentiralo. Tajni ključ obezbeđuje da se plaćanje izvrši što bezbednijim putem, i on je deklarisan na serverskoj strani. Na listingu 7 prikazan je način evidentiranja uspešnog plaćanja.

```

@Value("${api.stripe.sk}")
private String secretKey;

@PostConstruct
public void init() {
    Stripe.apiKey = secretKey;
}

public StripeChargeDTO
chargePayment(StripeChargeDTO
chargeRequest) throws StripeException {
    chargeRequest.setSuccess(false);
    Map<String, Object> chargeParams =
    new HashMap<>();
    chargeParams.put("amount", (int)
(chargeRequest.getAmount() * 100));
    chargeParams.put("currency",
"usd");
    chargeParams.put("source",
chargeRequest.getStripeToken());

    Charge charge =
    Charge.create(chargeParams);

    if(charge.getPaid()){
        chargeRequest.setChargeId(charge.getId());
        chargeRequest.setSuccess(true);
    }

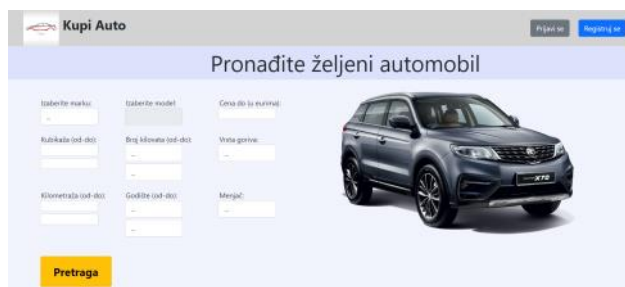
    return chargeRequest;
}
  
```

Listing 7 – Evidentiranje plaćanja

Klijentski deo

Klijentski deo aplikacije čini view komponenta u arhitekturi aplikacije. Zadatak ove komponente jeste prikaz podataka korisniku. Različite stranice definisane su u okviru komponenti, uz pomoć .html, .css i .ts fajlova, gde se kreira sam sadržaj stranice, kao i pozadinska, programska logika.

Početna stranica aplikacije nudi mogućnost pretraživanja oglasa, kao i prijavu odnosno registrovanje novog korisnika na sistem (slika 8).



Slika 8 – Početna stranica

Pretraga se može vršiti po jednom ili više parametara. U slučaju da korisnik ne navede parametre, prikazuju se svi dostupni oglasi na sajtu.

Korisniku se prikazuju rezultati pretrage na drugoj stranici, gde je moguće otvoriti svaki oglas ponaosob. Profil oglasa nudi informacije o automobilu, kao i o kontaktu prodavca. Takođe, ukoliko ih poseduje još, korisnik može da izlista sve registrovane automobile postavljene od strane tog kupca.

Dodavanje novog oglasa je moguće od strane registrovanog korisnika, i ono podrazumeva da se osnovne informacije o automobilu (marka, model, kubikaža..) obavezno navedu, dok su ostale informacije opcione.

Stripe biblioteka

U okviru index.html stranice, u klijentski deo aplikacije ubačena je i Stripe.js biblioteka koja nudi formu za onlajn plaćanje. Kako se svaki oglas naplaćuje, završni korak ka uspešnom registrowanju novog oglasa na sistem jeste popunjavanje forme podacima o kartici. Nakon popunjavanja svih polja, podaci se pakuju u token, i uz javni ključ šalju ka serverskom delu aplikacije gde se plaćanje i evidentira.

6. ZAKLJUČAK

Prvobitno kreirana aplikacija predstavljala je monolitnu aplikaciju za prodaju automobila implementiranu po uzoru na sisteme opisane u drugoj sekciji. Za potrebe rada, aplikacija je refaktorisana i organizovana kao skup mikroservisa koji međusobno komuniciraju. Mikroservisna arhitektura implementirana je sa ciljem da ukloni svaku zavisnost između modula aplikacije. Kao takva, omogućava nezavisno implementiranje, razvoj kao i testiranje, u različitim jezicima i okruženjima. Pored mikroservisne arhitekture, aplikacija je deploy-ovana putem Docker-a, a uz aplikaciju je integrisan i sigurnosni sistem za plaćanje pod nazivom Stripe.

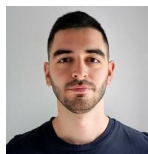
Iz svega pomenutog, može se zaključiti da je projekat razvijan u tehnologijama koje su u današnje doba veoma popularne i sklone unapređivanju. Samim tim, i aplikacija je sklona daljem unapređenju i usavršavanju.

Kombinacija mikroservisne arhitekture i Docker-a predstavlja jedan čvrst i stabilan temelj za budući rast i razvoj i omogućava aplikaciji da se lako prilagođava promenama i zahtevima tržišta.

7. LITERATURA

- [1] Mikroservisna arhitektura – <https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/microservices>
- [2] Polovni automobili - <https://www.polovniautomobili.com/>
- [3] Mobile.de – <https://www.mobile.de/>
- [4] Angular – <https://angular.io/>
- [5] Bootstrap – <https://getbootstrap.com/>
- [6] Java – <https://docs.oracle.com/javase/8/docs/api/>
- [7] Spring – <https://spring.io/>
- [8] Docker - <https://docs.docker.com/get-started/overview/>

Kratka biografija:



Vladimir Vrbica rođen je 24.01.2000. u Subotici. Završio je Osnovnu školu "Sonja Marinković" u Subotici, kao i Gimnaziju "Svetozar Marković". Završio je osnovne akademske studije na Fakultetu tehničkih nauka 2022. godine, smer Računarstvo i automatika. Nakon završenih osnovnih studija upisao je master studije na istom fakultetu, smer Računarstvo i automatika, modul Elektronsko poslovanje. Položio je sve ispite predviđene planom i programom.



Milan Vidaković rođen je u Novom Sadu 1971. godine. Na Fakultetu tehničkih nauka u Novom Sadu završio je doktorske studije 2003. godine. Na istom fakultetu je 2014. godine izabran za redovnog profesora iz oblasti *Primenjene računarske nauke i informatika*.

МУЛТИМОДАЛНО ПРЕПОЗНАВАЊЕ ЕМОЦИЈА ПОМОЋУ КОМПРИМОВАНИХ ГРАФОВСКИХ НЕУРОНСКИХ МРЕЖА**MULTIMODAL EMOTION RECOGNITION USING COMPRESSED GRAPH NEURAL NETWORKS**

Тијана Ђуркић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У раду је прво описан један алгоритам за препознавање емоција на основу звука, текста и видео записа, уз коришћење графовских неуронских мрежа. Потом су представљени резултати примене компресије на модел, која је потребна како би се обимни модели могли користити и на мањим уређајима.

Кључне речи: графовске неуронске мреже, препознавање емоција, мултимодални подаци, компресија

Abstract – In this paper, first, an algorithm for emotion recognition based on sound, text and video, using graph neural networks is described. Then, the results of applying compression to the model are present, which is needed so that large models can be used on smaller devices.

Keywords: graph neural networks, emotion recognition, multimodal data, compression

1. УВОД

Различити уређаји постају све доступнији и заступљенији у свакодневном животу. Због тога човек више времена него раније проводи користећи уређаје. Тиме се смањило време које је раније било посвећено разговору са другим људима. Управо је због тога пожељно да интеракција са уређајима што је могуће више личи на комуникацију између два човека.

Пошто се комуникација одвија путем видеа, звука и текста, постоји захтев за моделом који би успешно обрађивао сва три модалитета. Алгоритам који се показао успешним за решавање овог проблема јесте графовске неуронске мреже, GNN (енгл. – *Graph Neural Networks*). Главни задатак овог рада јесте да покаже теоријски преглед овог алгоритма, детаљнију анализу рада који се бавио овом темом, као и резултате компресије параметара. У другом поглављу дате су опште информације о емоцијама и теоријски преглед GNN методе и компресије, у трећем детаљна анализа рада који је послужио као узор, у четвртном допринос и резултати, а у петом закључак и даљи рад.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био доцент др Синиша Сузић.

2. ТЕОРИЈСКИ ПРЕГЛЕД

У овом поглављу ће детаљније бити представљена теорија која стоји иза GNN методе и компресије.

2.1 Емоције

Емоције су одговор на дешавања око нас. Према Нику Фрицци, емоције су исходи човековог приступања свету односно његовог доживљаја околине које касније модификују реакције на дешавања око њега и поступке који се предузимају као одговор на та дешавања [1]. Због значаја које емоције имају у природној комуникацији, пожељно је да буду укључене и у интеракцију човек – машина како би коришћење уређаја било приближније човековој природи.

2.2 Графовске неуронске мреже

Графовске неуронске мреже су подгрупа неуронских мрежа. Основа неуронске мреже јесте перцептрон (неурон) [2]. Узорак се класификује у зависности од тога да ли је вредност излаза испод или изнад одређеног (бинарна класификација). Уместо прага, користи се слободан члан p који се назива померајем (енгл. – *bias*). За сложеније проблеме користи се комбиновање неурона у неуронске мреже.

Корен графовских неуронских мрежа су графови. Граф представља везе између групе ентитета који се називају чворови. Ивице или гране повезују чворове и описују њихову повезаност [3]. Основна идеја GNN алгоритма је да се сваки чвор графа моделује на основу веза са суседним чворовима чиме се обезбеђује да се науче сложене репрезентације чворова и целокупног графа. То се постиже итеративним ажурирањем стања или карактеристика чвора на основу информација из „суседства“. Прво се прикупе информације из околине и комбинују се, најчешће помоћу неке функције. Ти прикупљени подаци се користе за ажурирање стања чвора кроз неку функцију која укључује неуронску мрежу. Врши се комбиновање тренутних карактеристика чвора и информација из суседних.

Најважнији параметри за GNN алгоритам су тежине веза (енгл. – *weights*) и померај (енгл. – *bias*). Тежине представљају јачине веза између чворова у суседним слојевима неуронске мреже, а утичу и на смер преноса информација. Што је утицај једног чвора на други већи, то је и вредност тежинског коефицијента већа. Слично као што је раније поменуто, померај се додаје излазу сваког чвора пре примене активационе

функције. Он доприноси померању излаза дуж осе вредности и тиме даје могућност моделу да уочи структуре у подацима. Без *bias*-а би се вршило само линеарно пресликавање улаза на излаз. Подешавањем оптималних тежина и помераја приликом обуке, модел учи и прилагођава се подацима из скупа за обуку што касније доводи до успешнијих перформанси приликом решавања проблема.

2.3 Компресија

Компликованији задаци изискују више слојева и параметара. Ипак, велики модели у погледу броја параметара нису практични када их је потребно применити у реалном времену због ограничености меморијских ресурса и хардверских компоненти, нпр. у мобилним телефонима. Такође се показало да су ти модели препараметризовани (енгл. – *over-parametrized*), односно да толики број параметара заправо није потребан [4]. Да би се постигла боља ефикасност извршавања, са задовољавајућим очувањем резултата, потребно је вршити компресију.

У овом раду су примењене две методе квантизације и оне ће бити објашњене у наредним поглављима.

3. ОСНОВНИ РАД

Рад који је послужио као основа за израду овог рада заснован је на архитектури названој COGMEN описаној у [5]. Као узорци за препознавање емоција користе се аудио, видео и текст разговора јер се информације из ова три различита модалитета међусобно допуњају и дају потпунији утисак. Оно што издваја овај рад од осталих на сличну или исту тему јесте да се узимају у обзир и утицај контекста разговора (глобалне информације) и локалне информације односно међусобна зависност саговорника, као и зависност од самог појединачног говорника на временски блиске реченице. Циљ је препознати емоцију изражену у једној реченици говорника уз све горе поменуте новитете који су уведени.

Како би се препознао контекст (глобална информација) и његов утицај на сваку појединачну реченицу (узорак) користи се трансформерски кодер [6] који користи механизам пажње за уочавање односа између различитих делова секвенци и њихово обрађивање. Што се тиче локалних информација, емоција изражена у једној реченици је често плод утицаја околних, суседних реченица, па је било потребно утврдити утицај међу саговорницима и утицај саговорника на самог себе. За ове потребе развијен је граф где свака реченица представља чвор, а усмерене ивице различите повезаности где је важан редослед реченица. Битно је напоменути да се једна реченица састоји из аудио, видео и текстуалне репрезентације, па је самим тим и чвор мултимодалне природе. Разликује се усмерена веза између реченица које изговара један говорник и усмерена веза између реченица које потичу од више њих.

3.1 Архитектура модела

На слици 1 приказана је оригинална архитектура модела. Улазне реченице прво наилазе на издвајач контекста, у слободном преводу (енгл. – *context extractor*). Спојена обележја сва три модалитета

(видео, аудио и текст) користе се као улаз за сваку реченицу дијалога. Како би се препознао контекст (глобална информација) и његов утицај на сваку појединачну реченицу (узорак) издвајач контекста користи кодер трансформера [6].

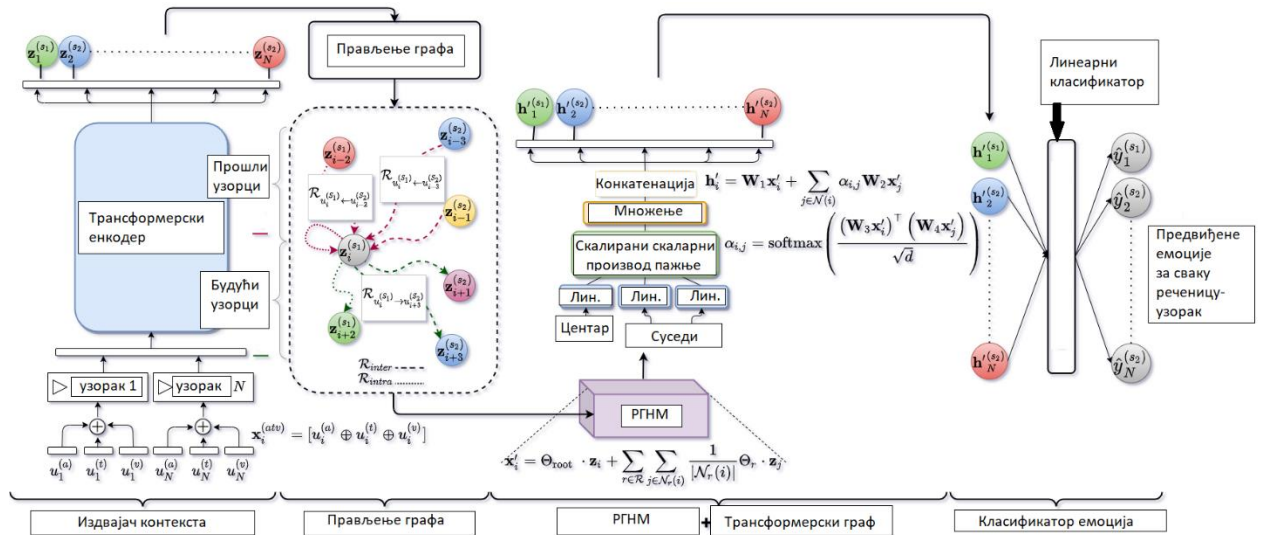
Главни део трансформерског кодера јесте механизам вишеструке пажње (енгл. – *multi-head attention mechanism*) чија је улога да омогући декодеру да искористи делове улазних секвенци који су најреlevantнији додељивањем тежинских коефицијената енкдованим улазним векторима, где се највеће вредности коефицијената приписују најбитнијим узорцима. Последњи слој кодера је мрежа са пропагацијом унапред. На крају се добија вектор обележја за сваку реченицу.

На основу добијених обележја из кодера, прави се граф који уочава везе између реченица дијалога. Као што је већ напоменуто, сваки чвор графа представљен је једном реченицом у сва три модалитета које су повезане усмереним ивицама (гранама). Разликујемо гране које повезују реченице које изговара исти говорник и гране које повезују реченице различитих говорника. Такође, прате се будуће и претходне релације сваке реченице односно за сваку реченицу се зна које су јој све претходиле, а које све следовале.

Наредна целина је релациона графовска неуронска мрежа (енгл. – *Relational Graph Convolutional Network (RGCN)*) која је предложена у ранијем раду [7]. Њена улога је прикупљање трансформација специфичних за однос међу суседним чворовима које зависе од типа и смера грана и то све кроз нормализовану суму. За разлику од класичног GNN алгоритма који све ивице третира подједнако, ова мрежа уочава разлику између типова ивица узимајући у обзир и специфичности веза које те чворове повезују. На тај начин RGCN кодира и информације о самим узорцима, али и њихове међусобне релације.

У раду од интереса, овај модел мреже примењује зависност повезаних реченица од самог говорника и више саговорника. Графовска неуронска мрежа у овом раду има 52 слоја где као параметри фигуришу тежине (енгл. – *weights*) и помераји (енгл. – *biases*). Има 8 слојева који садрже преко милион параметара. За издвајање лабела и обележја из чворова користи се трансформерски граф (енгл. – *graph transformer*) [8] који те информације користи као улаз у пропагацију кроз мрежу. Циљ је да на различите начине приступа суседним чворовима на основу њихових међусобних веза и да тиме уочава структуру графа. Предност трансформерског графа је што уочава дугорочне зависности кроз механизам самопажње (енгл. – *self-attention mechanism*), који омогућава моделу да одреди значај улазне секвенце у односу на друге делове улаза.

Додатне предности су да се паралелним рачунањем смањује време извршавања и обуке, а узимањем у обзир дугорочних зависности нема ограничења у погледу меморијских или рачунарских ресурса. На самом крају целокупног модела налази се класификатор емоција који се састоји из једног линеарног слоја.



Слика 1. Архитектура оригиналног модела [5]

3.2 Резултати оригиналног рада

Перформансе модела проверене су на две базе података.

IEMOCAP (енгл. – *The Interactive Emotional Dyadic Motion Capture*) је мултимодална база података за препознавање емоција где је свака реченица обележена једном од 6 емоција [9]. Она се састоји из снимљених видео записа дијалога глумаца од којих је тражено да одглуме одређене емоције.

MOSEI (енгл. – *Multimodal Opinion Sentiment and Emotion Intensity*) је друга широко распрострањена мултимодална база са 6 емоција: срећа, туга, гађење, страх, изненађеност и љутња [10]. Метрике које су коришћене за евалуацију оригиналног модела су макро средња вредност прецизности, осетљивости и F -мере која износи 82.39%, 81.24% и 81.66% респективно, потом пондерисана средња вредност прецизности, осетљивости и F -мере, од 82.22%, 81.97% и 81.96% респективно и просечна тачност од 81.97% и просечна F -мера од 81.96%.

Показан је велик утицај контекста на одлучивање о присутној емоцији јер се повећањем броја секвенци из дијалога, добија бољу увида у контекст целокупног дијалога, постижу веће вредности F -мере. Такође, уколико се изостави GNN из архитектуре модела примећује се пад перформанси, јер се уз помоћ GNN боље издвајају и групишу реченице у којима су изражене исте емоције. Тиме се даје на значају локалним информацијама које су раније објашњене.

4. ДОПРИНОС ОРИГИНАЛНОМ РАДУ

Као надоградња на постојећи рад, примењена је квантизација на већ истрениран модел чији је циљ смањити величину модела у погледу величине фајла и убрзати његово извршавање смањењем прецизности параметара, пре свега тежина. Тиме се уједно смањује и потрошња енергије. На основу хистограма слојева мреже тешко је било утврдити неку одређену расподелу, па је одлучено да се примени бинарна квантизација као веома лака за имплементирање и $FP8$ (енгл. – *Floating Point 8*) као веома корисна и све коришћенија метода.

4.1 Бинарна квантизација

Бинарна квантизација мења улазну вредност користећи само један бит. Један од изазова бинарне квантизације јесте балансирање између компресије података и очувања квалитета.

Претварање континуираних или вредности широког опсега у само две могуће може довести до губитка битних детаља и информација, што у појединим ситуацијама може бити неприхватљиво.

4.2 Floating point 8

Користећи *Floating Point*, реални бројеви се представљају са фракционим делом. Фракциона компонента су цифре после зареза у децималном запису бројева. *Floating Point* бројеви се састоје из три компоненте: бит знака (енгл. – *the sign bit*), експонент и мантиса (фракција). Бит знака даје информацију о томе да ли је број позитиван или негативан, експонент одређује скалу, а мантиса прецизност или значајне цифре [11].

За $FP8$ важи да је то формат смањене прецизности који користи 8 бита за енковање. Улазни подаци биће квантизовани на вредности које се могу представити овим бројем бита, а на основу оригиналних података и формула које иду уз овај начин квантизације.

Као надоградња на постојећи рад, бинарна квантизација и $FP8$ су прво примењени на слојевима који имају преко милион параметара и њих има 8. Пошто они заузимају највише меморијског простора због свог броја, а један од главних циљева квантизације јесте уштеда меморијског простора, очекивано је да ти слојеви буду први квантизовани. Потом су сви слојеви квантизовани и анализиран је утицај квантизације на резултате предвиђања модела.

4.3 Резултати

Пошто су резултати бољи када се примени квантизација на све параметре, само ће они бити представљени у раду. Може се приметити знатан пад у перформансама модела приликом предвиђања емоција када се примени бинарна квантизација.

Макро средња вредност прецизности, осетљивости и F -мере износи 49.42%, 36.85%, 35.25% респективно, потом пондерисана средња вредност прецизности, осетљивости и F -мере, од 48.13%, 44.75% и 35.25% респективно и просечна тачност од 44.75% и просечна F -мера од 39.54%. Закључено је да неуниформна расподела података над којима је примењена квантизација утиче на пад перформанси [12], што је могуће да је узрок толико слабир резултатима јер се из хистограма види да расподела вредности не одговара Лапласовој расподели око нуле, а у том случају би бинарна квантизација била погодан метод за компресију.

У општем случају, уколико је узрок лошијих резултата неуниформна расподела, тада се врши нормализација током финог подешавања (енгл. – *fine tuning*) или нелинеарна квантизација (нпр. логаритамска). Такође, могуће је да комбинација обичних графовских неуронских мрежа и детерминистичке бинарне функције није предодређена за постизање задовољавајућих резултата.

Овој архитектури модела много више одговара $FP8$ метода за добијање жељених резултата. Могуће објашњење је да је можда оригинални модел могао боље да се истренира, односно да је његова обука заустављена у локалном минимуму. Узрок је то што се $FP8$ методом вредности параметара заокружују на најближе квантизационе нивое који се само мало разликују од оригиналних вредности.

Приликом тренирања модела, вредности параметара се такође мењају у малим корацима. Када се тачност мало побољша, то значи да је хипотетички, примењено даље учење модела јер су вредности параметара мењане за мале вредности које одговарају кораку учења, тако да је и у раду [11] постигнуто побољшање модела.

Макро средња вредност прецизности, осетљивости и F -мере износи 80.90.42%, 81.93%, 81.39% респективно, потом пондерисана средња вредност прецизности, осетљивости и F -мере, од 81.86%, 81.76% и 81.78% респективно и просечна тачност од 81.76% и просечна F -мера од 81.78%.

5. ЗАКЉУЧАК И ДАЉИ РАД

На основу изнетих резултата, може се закључити да компресија модела неком од врста квантизације, на овом раду $FP8$, не мора да има негативан утицај на саму тачност модела, чак може и да је мало побољша. $FP8$ метода се показала као много успешнија од бинарне квантизације.

У даљем раду могло би да се испита каквим проблемима компресије и типовима модела више одговара примена бинаризације. Ово би било значајно јер бинаризација није ни временски ни рачунски захтевна метода, па би се резултати добијали много брже, а ако се покаже да за одређене случајеве не нарушава перформансе, била би јако добра за примену. Такође, могли би се испитати неки други начини компресије или њихова комбинација и проверити да ли то додатно поспешује или оштећује модел.

6. ЛИТЕРАТУРА

- [1] N.H. Frijda, *The emotions*, Cambridge University Press; 1986.
- [2] Т. Носек, Б. Бркљач, Д. Деспотовић, М. Сечујски, Т. Лончар-Турукало, „Практикум из машинског учења“, Факултет техничких наука, Универзитет у Новом Саду, 2020.
- [3] H. Abdi, D. Valentin, & B. Edelman, *Neural networks* (No. 124). Sage, 1999.
- [4] J. O. Neill, An overview of neural network compression. *arXiv preprint arXiv:2006.03669*, 2020
- [5] A. Joshi, A. Bhat, A. Jain, A. V. Singh, & A. Modi, COGMEN: COntextualized GNN based multimodal emotion recognition. *arXiv preprint arXiv:2205.02455*, 2022
- [6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, ... & I. Polosukhin, Attention is all you need. *Advances in neural information processing systems*, 30, 2017
- [7] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, & M. Welling, Modeling relational data with graph convolutional networks. In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15* (pp. 593–607). Springer International Publishing, 2018
- [8] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, Y. Sun, Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*, Sep 8, 2020
- [9] C. Busso, M. Bulut, C. C. Lee, A. Kazemzadeh, E. Mower, S. Kim, ... & S. S. Narayanan, IEMOCAP: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42, 335–359, 2008
- [10] A. B. Zadeh, P. P. Liang, S. Poria, E. Cambria, & L. P. Morency, Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 2236–2246), July, 2018
- [11] Н. Симић, С. Сузић, Т. Носек, М. Вујовић, З. Перић, М. Савић, и В. Делић, Speaker recognition using constrained convolutional neural networks in emotional speech. *Entropy*, 24(3), 414, 2020
- [12] T. Liang, J. Glossner, L. Wang, S. Shi, & X. Zhang, Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, 461, 370–403, 2021

Кратка биографија:



Тијана Ђуркић рођена је у Новом Саду 1998. године, а одрасла у Бечеју. Дипломски рад одбранила је на Факултету техничких наука из области машинског учења. Контакт: tijana.djurkic@gmail.com

RAZVOJ VEB APLIKACIJE U REALNOM VREMENU NA PRIMERU CHAT-a

DEVELOPMENT OF A WEB APPLICATION IN REAL TIME USING THE EXAMPLE OF CHAT

Alisa Milanko, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – Veb aplikacija Chat predstavlja aplikaciju za razmenu poruka između dva korisnika u realnom vremenu. Pored osnovne funkcionalnosti omogućava registrovanje i prijavljivanje korisnika, kao i kreiranje i izmenu ličnog profila. Za implementaciju korišćena je biblioteka SignalR.

Ključne reči: Veb u realnom vremenu, SignalR, ASP.NET Core

Abstract – The web application Chat is an application for exchanging messages between two users in real time. In addition to basic functionality, it enables user registration and login, as well as creation and modification of a personal profile. The SignalR library was used for implementation.

Keywords: Realtime web, SignalR, ASP.NET Core

1. UVOD

Za razliku od desktop aplikacija koje se instaliraju na nekom lokalnom računaru, veb aplikacije su aplikacije koje se izvršavaju na nekom udaljenom računaru. Takvoj aplikaciji pristupa korisnik, odnosno klijent, sa svog računara korišćenjem interneta i pretraživača. Veb server predstavlja računar na kome se izvršava aplikacija. U ovom radu govorićemo o veb aplikacijama koje rade u realnom vremenu, odnosno o komunikaciji u realnom vremenu. Veb je u svojoj ranoj fazi postojanja bio vrlo različit od onoga što danas poznajemo. Jedan od nedostataka bio je taj da je veb stranicu bilo potrebno ponovo učitati kako bi se dobili novi podaci.

Mogućnost dobijanja podataka bez ponovnog učitavanja stranice drastično je promenila način na koji danas koristimo veb. Pod komunikacijom u realnom vremenu smatra se komunikacija dva ili više korisnika računara u kojoj su za vreme komunikacije svi korisnici na mreži.

Danas jedan od glavnih načina pomoću kojih komuniciramo jesu društvene platforme poput Facebook-a, Twitter-a i Google+-a. Ove platforme su izgrađene na veb tehnologijama u realnom vremenu.

Chat je, nesumnjivo, jedan od najpopularnijih, ali i najspecifičnijih vidova komunikacije putem interneta. Reč je o novom obliku komuniciranja, različitom od standardnog govornog i pisanog jezika.

Dok je jezik Email-a mnogo bliži jeziku pisanih žanrova i u nekim svojim karakteristikama podseća na pisanje klasičnih pisama, jezik Chat-a najviše je sličan živom govoru, iako se i on javlja u pisanoj formi.

2. VEB KOMUNIKACIJA U REALNOM VREMENU

Realno vreme (eng. real time), ponekad prevedeno i kao potrebno vreme, može se objasniti kao najduže vreme koje je potrebno za izvršavanje nekog zadatka. Ovaj pojam često se, donekle pogrešno, koristi za vreme u kojem zadatak mora biti izvršen kako bi se ljudima činilo da je izvršen odmah, uglavnom zbog velike količine aplikacija kojima danas to i jeste realno vreme. Realno vreme može biti bilo koje trajanje u zavisnosti od potreba aplikacije. Potrebe veb aplikacija su se znatno izmenile u prvoj deceniji XXI veka. Korišćenje interneta i veb pretraživača (eng. browser) postalo je sinonim, jer se cela paradigma odvija kroz veb pretraživač. Ljudi ih koriste kao primarnu platformu kako bi odradili sve zahteve koje potražuju na internetu. Veb je podeljen na dva različita tipa: statički i dinamički. Statički veb prikazuje isti sadržaj za svakog korisnika. Oni su obično sa fiksnim brojem stranica koje imaju određeni izgled. Kada se stranica pokreće u pregledaču, sadržaj je statičan i ne menja se kao odgovor na radnje korisnika.

Oni jesu ograničeni, ali ih je vrlo jednostavno napraviti. Sa druge strane, dinamički veb, može da prikaže različit sadržaj i pruža korisničku interakciju korišćenjem naprednog programiranja i baze podataka. Popularne socijalne mreže su izgrađene na vrhu web browser platformi i njih danas koristi više od milijardu korisnika u svetu. Glavni razlog popularnosti socijalnih mreža jeste pružanje informacija i poruka od drugih korisnika u realnom vremenu. Sve ovo je dovelo do praktičnog prestanka korišćenja statičkog veba i dovelo do početka razvoja dinamičkih veb aplikacija koje operišu u realnom vremenu.

2.1. HTTP tehnike

HTTP (Hypertext Transfer Protocol) je mrežni protokol koji se koristi za virtuelno dostavljanje podataka (resursa) na vebu. Ovi podaci obično predstavljaju HTML fajlove, slike, rezultate upita ili nešto drugo. Web browser je HTTP klijent koji šalje zahtev (eng. request) prema HTTP serveru koji potom vraća odgovor (eng. response) prema klijentu koji sadrži resurse. Standardni port na kojem HTTP server osluškuje request-e je -80, ali se mogu koristiti i ostali portovi. HTTP se temelji na stalno otvorenoj vezi između klijenta i servera za vreme razmene poruka. Klijent šalje zahtev serveru koji ga obrađuje i

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Željko Vuković, docent.

šalje odgovor. Nakon što klijent primi odgovor, veza između klijenta i servera se prekida. Za svaku sledeću razmenu poruke, odnosno da bi se načinile izmene na veb stranici, potrebno je opet uspostaviti vezu. Ovo dovodi do statičkih veb stranica koje zahtevaju da se browser osveži (eng. refresh) kako bi nove promene bile prikazane. Ova tehnika se zove „serverski pull”(klijent pull-uje podatke sa servera svaki put kada pošalje novi HTTP request.

2.1.1. AJAX

U klasičnoj veb aplikaciji kada korisnik inicira određenu akciju na primer klikom na hyperlink šanje se HTTP zahtev, koji se šalje od strane veb pretraživača na procesuiranje prema serveru. Veb server procesuirao request i šalje nazad response, koji sadrži celu veb stranicu za prikazivanje u korisnikovom web browser-u. Dok se ovo događa, korisnik čeka odgovor i njegova interakcija sa aplikacijom je blokirana. Osvežavanje cele veb stranice nije uvek potrebno, na primer prilikom validacije podataka, a rešenje za to nudi AJAX. Ovo rešenje je bazirano na modelu koji dopušta da se slanje zahteva i primanje odgovora dešava u pozadini. Dakle, rešenje koje zahteva real-time vreme izvršavanja se odvija u pozadini, a aktivnosti koje zahtevaju ogromnu količinu podataka (kao što je čekanje odgovora koji sadrži kompletnu veb stranicu) odvijaju se normalno, sa osvežavanjem stranice [1].

2.1.2. Polling

Glavno oružje za izgradnju veb aplikacija, koje sve više zamenjuju desktop aplikacije, jeste taj što AJAX omogućava učitavanje samo delova stranica bez osveživanja čitavog njenog sadržaja. AJAX konekcija prema serveru nije konstantno otvorena, tako da ukoliko aplikacija zahteva konstantnu promenu podataka, AJAX skripta nema način da sazna da li je došlo do promene podataka. Iz tog razloga potrebno je konstantno slati AJAX upite serveru da li je došlo do promene podataka.

Polling predstavlja najjednostavniju tehniku. Reč polling se može prevesti kao anketiranje, jer klijent kontinuirano anketira server [2].

2.1.3. Long Polling

Long polling tehnika se zasniva na rešenju koje vodi poreklo od desktop aplikacija. Reč je o otvaranju live konekcije između servera i klijenta. Ovom tehnikom klijent šalje zahtev prema serveru ali za razliku od klasične polling tehnike, server ne odgovara odmah. Tačnije, ostvaruje se live konekcija između klijenta koji je poslao zahtev, i servera, sve dok se ne dogodi događaj koji inicira da server vrati odgovor klijentu da je potrebno da se ažurira. Nakon primanja odgovora, klijent se ažurira i ponovo šalje zahtev serveru, i server čeka na novi događaj kako bi odgovorio klijentu. Potrebno je uvideti da se ovom tehnikom postiže manji broj zahteva od strane klijenta prema serveru. Ne postoje odgovori koji ne sadrže nikakvu informaciju, kao u tehnici klasičnog polling-a [3].

2.1.4. Forever Frame

Forever Frame (Hidden iFrame) predstavlja najjednostavniju tehniku za izgradnju dinamičkih veb aplikacija pomoću http streaming-a. Ideja je da se ugradi skriveni HTML element – iFrame. On se postavi da bude

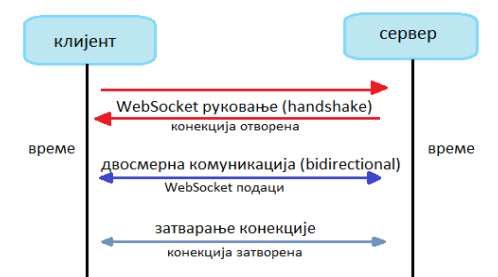
nevidljiv, unutrašnji prozor unutar glavnog prozora koji dozvoljava da se uveže jedan HTML dokument unutar drugog HTML dokumenta. iFrame se šalje prvi put kada klijent pošalje zahtev za veb stranicu serveru. Beskonačan je pa otuda i naziv forever frame. Kada se događaj koji zahteva klijent dogodi, server šalje script tagove u skriveni iFrame i ovaj iFrame se polako napuni tagovima. Ti script tagovi predstavljaju JavaScript naredbe koje pozivaju određene funkcije i unutar kojeg je ugrađen skriveni iFrame u kome se nalaze script tagovi. Svaki script tag se izvršava čim bude poslat od strane servera, a primljen od strane klijenta kroz konekciju koja je i dalje otvorena.

2.1.5. Server Sent Events (SSE)

Server Sent Events je još jedna od tehnika za serverski push podataka prema klijentu. Predstavlja jednostavan i efikasan način za slanje ažuriranih podataka u realnom vremenu od servera prema klijentu, pomoću HTTP veze. On je deo HTML5 specifikacije i podržavaju ga svi moderni veb pretraživači. Zasniva se na jednosmernom toku podataka, gde server šalje poruku klijentu, ali klijent ne može da šalje poruku nazad serveru [4].

2.2. Web Socket

Web socket-i predstavljaju HTML5 specifikaciju koja pruža potpunu dvosmernu (eng. duplex) konekciju između klijenta i servera koja razmenjuje podatke pomoću jedinstvenog socket-a preko veba. Moguće je slati neograničen broj zahteva tako dugo dok je veza otvorena i to sa vrlo malim troškom u resursima aplikacije. Veza koja se otvara WebSocket protokolom je takve prirode da dopušta klijentu i serveru da, nezavisni jedan od drugog, šalju poruke jedan drugom, čak i u isto vreme. Takođe, zbog malog trošenja resursa poput radne memorije, server može imati u isto vreme otvoren veliki broj veza prema više klijenata, ako očekujemo da aplikaciju koristi veliki broj korisnika istovremeno.



Slika 1. – WebSocket komunikacija

2.2.1. WebSocket Handshake

Kako bi se uspostavila WebSocket konekcija između klijenta i servera, i klijent i server se nadograđuju sa HTTP protokola na WebSocket protokol i ovo se naziva inicijalno rukovanje klijenta i servera (eng. handshake) [5].

3. ASP.NET SIGNALR

Danas se od veb aplikacija mnogo očekuje. Prvo, ne samo da pravilno funkcionišu, već to moraju da rade sa odličnim korisničkim iskustvom. Pored toga, zahteva se da budu veoma brze, te da se informacije isporučuju u realnom vremenu.

Prilikom kreiranja veb aplikacije treba uzeti u obzir nekoliko stvari. Aplikacija ima tendenciju da postane veoma poznata kada korisniku omogući interakciju sa drugim korisnicima ili sa ugrađenom bazom podataka. Preduzeće ima tendenciju da poboljša svoju korisničku bazu ako može da razvije aplikaciju koja će pomoći klijentima da komunicira sa njima.

Dobro dizajnirana veb aplikacija takođe može pomoći da se minimiziraju troškovi i vreme. Na primer, kompanija koja se bavi online prodajom može da napravi aplikaciju koja ažurira svoje cene na mreži i šalje poruku zainteresovanim kupcima o promenama cena. Ova vrsta aplikacije će uštedeti ogromnu količinu vremena. Tako veb aplikacija može pomoći preduzeću da redovno bude u kontaktu sa svojim klijentima.

Aplikacija bi trebalo da radi besprekorno na svim platformama. Da bi se to postiglo aplikacija mora da bude kreirana koristeći neke napredne platforme kao što je SignalR.

ASP.NET SignalR je biblioteka otvorenog koda koja omogućava programerima da pojednostave dodavanje veb funkcionalnosti aplikacijama, u realnom vremenu. Takođe, omogućava kodu na strani servera da trenutno prosleđuje sadržaj klijentima.

Dobri kandidati za upotrebu su aplikacije koje zahtevaju visokofrekventna ažuriranja sa servera, na primer: igre, društvene mreže, aukcije, mape i GPS aplikacije. Zatim, kontrolne table i aplikacije za praćenje, kolaborativne aplikacije i aplikacije koje zahtevaju obaveštenja poput društvenih mreža, e-mail-a, aplikacije za časiranje itd.

ASP.NET Core SignalR održava mnogo istih osnovnih koncepata i mogućnosti kao *ASP.NET SignalR*. Glavne razlike između njih su [6]:

- Automatsko ponovno povezivanje (eng. Automatic Reconnects) – klijentska aplikacija mora eksplicitno da pokrene novu vezu kada je to potrebno.
- Razlike kod klijenta – *ASP.NET Core SignalR* klijent je napisan u TypeScript-u. Može se koristiti JavaScript ili TypeScript klijentska biblioteka bez obaveznog upućivanja na jQuery.
- Skaliranje (eng. Scaleout) – Sada postoji mala razlika u skaliranju od kada je Azure SignalR Service počela da podržava *ASP.NET*.
- Podrška protokolu (eng. Protocol Support) – *ASP.NET Core SignalR* podržava JSON protokol, koji nije kompatibilan sa ranijim verzijama, i novi binarni protokol zasnovan na MessagePack-u. Takođe, omogućava kreiranje novih protokola.
- Ubrizgavanje zavisnosti (eng. Dependency Injection) – *ASP.NET* nije imao ugrađen Dependency Injection, a *ASP.NET Core SignalR* ima Dependency Injection koji je ugrađen u okvir (eng. framework) i tako se isporučuje. On se jednostavno može koristiti preko konstruktora, bez dodatne konfiguracije.

3.1. Transports

ASP.NET SignalR pojednostavljuje proces implementacije aplikacija u realnom vremenu. Iako je započeto sa *ASP.NET* veb aplikacijom, može je koristiti i druga .NET klijentska aplikacija. Jedna od ključnih komponenti

SignalR-a jeste transportni medijum, pomoću kojeg komunicira.

Transportni mediji dostupni za *SignalR* su *WebSocket*, *Server-Sent Events (SSE)*, *Forever Frame* i *Long Polling*.

SignalR je obezbedio omotač oko ovih transporta niskog nivoa koji omogućava programerima da se fokusiraju samo na poslovne zahteve umesto da brinu o tome da će se osnovni transport koristiti za slanje i primanje poruka između klijenta i servera. Na osnovu pretraživača i verzije pretraživača koju koristi veb aplikacija u realnom vremenu, *SignalR* u *ASP.NET Core*-u, koji je korišćen u ovom zadatku, automatski odlučuje o osnovnom transportu koji će se koristiti. Rezervni (eng. fallback) mehanizam se koristi za odlučivanje o transportnom sloju koji će koristiti aplikacija.

3.2. Hubs

SignalR koristi čvorove za komunikaciju između klijenta i servera. Čvor (eng. hub) je komponenta koja se nalazi u aplikaciji. On šalje i prima poruke od klijenta koristeći poziv udaljene procedure (eng. remote procedure call) koristeći osnovni transport. U suštini hub je centralna tačka u *ASP.NET Core*-u i on je odgovoran za rutiranje svih komunikacionih poruka u aplikaciji. Klasa koja nasleđuje klasu *Microsoft.AspNetCore.SignalR.Hub* i sadrži metode koje klijenti mogu da pozovu. Ove metode su poznate kao „hubs method”, koje se mogu pozvati iz koda na strani klijenta za slanje podataka serveru ili primanja podataka sa servera. Hubs takođe pružaju način na koji server šalje podatke određenim klijentima [7].

4. APLIKACIJA ZA DOPISIVANJE

4.1 Korišćene tehnologije i alati

Chat aplikacija predstavlja veb aplikaciju koja se sastoji od serverskog i klijentskog dela. Tehnologija koja je korišćena za serverski deo jeste *ASP.NET Core*, a od alata *Microsoft Visual Studio*. Na klijentskom delu aplikacije korišćeni su *Flutter* i *Microsoft Visual Studio Code*. Takođe, za bazu podataka korišćen je *Microsoft SQL Server Management Studio*.

4.1.1. ASP.NET Core

ASP.NET Core je okvir za veb razvoj otvorenog koda koji je dizajniran za izgradnju modernih aplikacija. Nudi brojne prednosti u odnosu na originalni *ASP.NET* okvir, uključujući poboljšane performanse, podršku za više platformi i pojednostavljen razvoj.

4.1.2. Microsoft Visual Studio

Visual Studio je okruženje (IDE – Integrated Development Environment) kompanije *Microsoft* za razvoj softvera: računarskih programa, veb stranica, veb aplikacija, veb servisa i mobilnih aplikacija. Kao alat *Visual Studio* je daleko više od jednostavnog editora namenjenog pisanju izvornog koda. Pored editora i dibagera uključuje kompajlere, alate za dovršavanje koda, grafičke dizajnere i mnoge druge funkcije.

4.1.3. Flutter

Flutter je UI (User Interface) softver otvorenog koda koji je kreirao *Google*. Omogućava da se iz jedne baze koda izgrade izvorno kompajlirane, multiplatformske aplikacije, za bilo koji veb pretraživač *Android*, *Fuchsia*, *iOS*, *Linux*, *macOS* i *Windows*. Programski jezik koji koristi je

Dart koji je optimizovan za brze aplikacije na bilo kojoj platformi.

4.1.4. Microsoft Visual Studio Code

Visual Studio Code je program za uređivanje teksta kompanije Microsoft. Predstavlja integrisano razvojno okruženje (eng. Integrated Development Environment - IDE) koji korisnicima nudi mogućnost uređivanja različitih vrsta teksta i koda.

4.1.5. Microsoft SQL Server Management Studio

Microsoft SQL Server Management Studio je softverska aplikacija koju je razvio Microsoft i koja se koristi za konfigurisanje, upravljanje i administraciju svih komponenti u okviru Microsoft SQL Server-a.

4.2. Arhitektura aplikacije

Arhitektura koja se koristila pri izradi ove aplikacije jeste troslojna arhitektura.

Troslojna arhitektura nastala je evolucijom dvoslojne arhitekture kako bi se prevazišli nedostaci koje je imala dvoslojna.

Aplikacije su podeljene u tri nivoa koji su međusobno labavo povezani i koji međusobno komuniciraju putem interfejsa koje poseduju prezentacioni sloj, sloj poslovne logike i sloj podataka.

4.3. Razmenjivanje poruka pomoću SignalR biblioteke

Ova aplikacija se sastoji od klijenta i servera, tako da je SignalR biblioteku potrebno implementirati na obe strane.

Ono što je prvo potrebno jeste instaliranje SignalR paketa. To se može postići pomoću Nuget menadžera (eng. Package Manager) ili konzole (eng. Package Manager Console). Sledeća stvar koju je potrebno uraditi jeste prijavljivanje servisa na serverskoj strani. Potrebno je kreirati SignalRHub klasu. SignalRHub nasleđuje klasu Hub koja upravlja vezama, grupama i porukama. Ona sadrži samo jednu funkciju SendMessage koja prima parametar SignalRMessage. Parametar predstavlja klasu kreiranu za poruke koje će se razmenjivati među korisnicima. Zatim mapiranje krajnje tačke (eng. endpoint-a) na SignalRHub klasu i prosleđivanje URL-a kako bi korisnici mogli da se konektuju na Hub.

Prelazimo na klijentsku stranu. Kada želimo da rukujemo svim stanjima koja se mogu dogoditi u aplikaciji, možemo koristiti Flutter Bloc. Stanje (eng. state) je stanje aplikacije u bilo kom trenutku. Na primer: kada korisnik dodirne dugme koje ga vodi na drugu stranicu, aplikacija je promenila svoje stanje. Unošenje teksta u prazno polje za tekst takođe dovodi do promene stanja.

Bloc je najbolji način za upravljanje stanjima. Predstavlja šablon dizajna (eng. Design Pattern) koji je kreirao Google da bi pomogao da se poslovna logika odvoji od ostalih slojeva aplikacije.

Cubit predstavlja minimalnu verziju ili podskup Bloc Design Pattern-a koji pojednostavljuje način na koji upravljamo stanjem aplikacije. On je sličan Bloc-u, ali ne zna ništa o događajima i oslanja se na metode za emitovanje novih stanja [8]. U ovom zadatku koristio se Cubit.

Tokom trajanja komunikacije između korisnika, Cubit služi da bi sve vreme slušao stanje aplikacije i tako prikazivao određene poruke. Veza je sve vreme otvorena.

Promenom stanja pozivaju se određene metode jedna za drugom, da bi konačno došlo do uspešnog stanja, kad se iz njega mogu izvući određene informacije koje su nam potrebne za prikaz poruka.

5. ZAKLJUČAK

Zadatak ovog rada bio je da se prikaže komunikacija u realnom vremenu, korišćenjem SignalR biblioteke. Na početku prošli smo malo kroz istoriju veba, tehnika za komunikaciju, zatim objasnili neke pojmove koji su bili bitni prilikom izrade, videli kakva je arhitektura aplikacije, a onda i prikazali to na konkretnom primeru jedne aplikacije za čascanje.

Kao zaključak možemo reći da je WebSocket tehnika za komunikaciju najefikasnija i najprimenljivija tehnika komunikacije u realnom vremenu. Omogućava jedinstvenu konekciju između klijenta i servera koja može da dostavlja podatke u oba smera, što ostale opisane tehnike ne podržavaju. Lakoća implementacije i upravljanje događajima su takođe neke od prednosti ove tehnike. Veb se razvija iz dana u dan, a društvene mreže jesu jedan od primera njegovog rasta i razvoja koje su već neko vreme naša svakodnevnica. Stigli smo do kraja ovog rada, nadam se da sam postigla svoj cilj i približila razumevanje veb aplikacije u realnom vremenu na baš takvom primeru aplikacije za čascanje.

6. LITERATURA

- [1] Jesse James Garrent – *Ajax: A New Approach to Web Applications*
- [2] <https://www.geeksforgeeks.org/what-is-polling-in-ajax/>
- [3] <https://www.pubnub.com/blog/http-long-polling/>
- [4] <https://levelup.gitconnected.com/real-time-updates-made-simple-a-introduction-to-server-sent-events-sse-db78ce3adf23>
- [5] <https://learn.microsoft.com/en-us/aspnet/core/fundamentals/websockets?view=aspnetcore-7.0>
- [6] <https://redwerk.com/blog/asp-net-core-signalr-introduction/>
- [7] <https://learn.microsoft.com/en-us/aspnet/core/signalr/hubs?view=aspnetcore-5.0>
- [8] Dmitrii Slepnev – *State Management Approached In Flutter*

Kratka biografija:



Alisa Milanko rođena je u Kikindi 17.05.1995. Završava osnovnu školu „Đura Jakšić“, a nakon toga srednju Ekonomsko-trgovinsku školu smer Ekonomski tehničar, takođe u Kikindi, završava 2014. godine. Iste godine upisuje Fakultet tehničkih nauka, smer Primenjeno softversko inženjerstvo. Nakon toga 2019. godine upisuje master studije na istom fakultetu, smer Računarstvo i automatika, podsmern Elektronosko poslovanje.

**KOMPONENTE ZA PODRŠKU SISTEMA ZA OPTIMIZACIJU RASPOREDA NASTAVE
FAKULTETA TEHNIČKIH NAUKA****COMPONENTS SUPPORTING THE SCHEDULE OPTIMIZATION SYSTEM OF THE
FACULTY OF TECHNICAL SCIENCES**

Ksenija Prčić, Nikola Luburić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U okviru rada razvijene su komponente u sistemu za kreiranje rasporeda nastave na Fakultetu tehničkih nauka u Novom Sadu. Fokus komponenti je parsiranje značajnih dokumenata za kreiranje rasporeda, transformacija dobijenih podataka i izvoz finalnog rasporeda u različite formate. Rad detaljno istražuje domen problema, uključujući specifične izazove i ograničenja, kao i korisničke zahteve koji su ključni za efikasno funkcionisanje sistema. Opisane su faze prikupljanja i obrade podataka korišćenjem ETL (Extract, Transform and Load) metodologije, arhitektura sistema i implementacija razvijenog rešenja. Predloženi su i potencijalni putevi za unapređenje sistema. Krajnji proizvod ovog istraživanja je funkcionalan sistem za kreiranje i optimizaciju rasporeda nastave, koji se integriše u informacioni sistem fakulteta, omogućavajući korisnicima da upravljaju kompletnim procesom, od obrade ulaznih podataka do konačnog prikaza rasporeda u različitim formatima.

Ključne reči: rukovanje dokumentima, problem planiranja rasporeda, predstava domenskog znanja

Abstract – As part of this paper, components supporting the teaching schedule optimization system at the Faculty of Technical Sciences in Novi Sad were developed. Focus of the components is parsing of input documents significant for teaching schedule creation, transformation of extracted data and exports of the optimized schedule into various formats. The thesis thoroughly investigates the problem domain, including specific challenges and constraints, as well as user requirements essential for the system's effective functioning. The stages of data collection and processing using the ETL (Extract, Transform, and Load) methodology are detailed, along with the system's architecture and implementation. Potential ways for system enhancements are also discussed. The final product of this research is a system that facilitates the creation and optimization of teaching schedules and is also integrated into a broader informational system of the faculty, allowing users to manage the entire process, from processing input data to the final display of schedules in various formats.

Keywords: document management, scheduling problem, representation of domain knowledge

1. UVOD

Problem kreiranja rasporeda predstavlja izazov u različitim aspektima društva, od individualne organizacije vremena do organizacije rada zaposlenih u velikim organizacijama. Fokus istraživanja su izazovi kreiranja rasporeda nastave u obrazovnim ustanovama, gde efikasnost organizacije nastave značajno utiče na kvalitet obrazovanja i zadovoljstvo studenata i nastavnog osoblja. Istraživanje se bavi kreiranjem rasporeda nastave za Fakultet tehničkih nauka u Novom Sadu, gde je uočena potreba za sistemom za optimizaciju rasporeda. Trenutno ne postoji sistematičan pristup prikupljanju i ažuriranju podataka, što dovodi do gubitka informacija.

Rad obuhvata prikupljanje i obradu podataka koji se koriste za kreiranje rasporeda. Prvi korak je identifikacija i prikupljanje podataka koji utiču na kreiranje rasporeda. Na osnovu dostupnih podataka formira se raspored i optimizuje iskorišćenost svih resursa.

Za obradu podataka dizajniran je parser podataka koji adaptira Extract, Transform and Load (ETL) [1] metodologiju. Ovaj proces uključuje ekstrakciju podataka iz različitih izvora dostupnih u raznim formatima, njihovu transformaciju u format pogodan za generator rasporeda i integrisanje transformisanih podataka u sistem za kreiranje i dalju optimizaciju rasporeda. Cilj istraživanja je razvoj sistema za obradu ulaznih podataka različitih struktura i njihovu transformaciju u format pogodan za korišćenje u širem sistemu optimizacije rasporeda. Sistem takođe kreira različita dokumenta za prikaz rasporeda zainteresovanim stranama. Naredno poglavlje razmatra specifične izazove i ograničenja domena problema prilikom izrade rešenja. Poglavlje 3 opisuje arhitekturu sistema. U 4. poglavlju opisana je implementacija sistema. Poslednje poglavlje rada je zaključak koji sadrži sažetak istraživanja i moguća buduća unapređenja.

2. DOMEN PROBLEMA

Rešenje problema optimizacije rasporeda u obrazovnim ustanovama je problem na svetskom nivou, što potvrđuje i postojanje međunarodnog takmičenja posvećenog ovoj temi [2]. U radu [3] opisan je postupak organizacije takmičenja kao i broj timova (preko 490) i država učesnica (preko 66), gde ovi brojevi govore o značajnosti problema.

Rad opisuje postupak rešavanja problema kreiranja rasporeda nastave na Fakultetu tehničkih nauka (FTN) u Novom Sadu sa fokusom na načine prikupljanja i obrade podataka potrebnih za formiranje rasporeda.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio docent dr Nikola Luburić.

Problem optimizacije rasporeda FTN-a u Novom Sadu opisan je u radu [4]. Glavni fokus ovog rada je postupak ekstrakcije podataka iz trenutnog informacionog sistema i korišćenje istih za formiranje novog rasporeda. Oba rada su bazirana na sličnom domenskom modelu koji uključuje: organizacione jedinice fakulteta (departmane i katedre), studentske programe, studentske grupe, predmete, predavače, prostorije, vremenske intervale i termine u okviru rasporeda nastave.

Proces ekstrakcije, transformacije i učitavanja podataka poznat je u svetu softverskog inženjerstva [5], posebno u sferi skladištenja podataka [6]. U radu [7] detaljno je opisan ETL proces primenjen u akademskom okruženju za usklađivanje podataka u distribuiranim bazama. Kako je trenutna struktura informacionog sistema FTN-a distribuirana i sastoji se iz većeg broja različitih komponenti, često dolazi do nekonzistentnosti podataka. U cilju centralizovanog rukovanja podacima značajnim za kreiranje i optimizaciju rasporeda, dizajniran je ETL proces koji prikuplja potrebne podatke, transformiše ih u oblik pogodan za obradu i čuva ih u dokument orijentisanom bazi radi daljeg održavanja [8].

3. SPECIFIKACIJA SISTEMA

U ovom poglavlju su detaljnije prikazani korisnički zahtevi i opisan je način integracije komponenti opisanih u ovom radu u postojeću informacionu infrastrukturu fakulteta. Rad je deo šireg sistema koji ima za cilj razvoj procesa koji obezbeđuje konzistentno rukovanje podacima neophodnim za kreiranje rasporeda. U nastavku je opisan ceo proces, od prikupljanja podataka, preko kreiranja rasporeda, do dostavljanja informacija o rasporedu svim zainteresovanim stranama.

3.1. Zainteresovane strane i izvori ulaznih podataka

Pre razvoja sistema potrebno je analizirati trenutnu strukturu dokumenata FTN-a, način na koji se oni kreiraju, održavaju i koriste u izradi rasporeda. Takođe, važno je utvrditi ko su zainteresovane strane za konačni raspored, kako se on formira, održava i dostavlja zainteresovanim stranama.

Prvi koraci u razvoju sistema su uključivali analizu strukture fakulteta i identifikaciju sektora koji učestvuju u kreiranju rasporeda ili za koje je raspored od značaja za rad. Na osnovu razgovora sa domenskim ekspertom, profesorima u nastavi, kao i sa profesorima koji se pored nastave nalaze i na organizacionim pozicijama u okviru FTN-a izdvojeni su sektori koji utiču na kreiranje rasporeda: studentska služba, kadrovska služba, katedre, departmani, dekanat i informaciono tehnološka (IT) služba.

Na osnovu inicijalnog istraživanja, prikupljeni su dokumenti koji utiču na kreiranje rasporeda. Izvori ovih dokumenata su: IT služba, domenski ekspert i zvanična stranica FTN-a. Pored sektora koji imaju uticaj na kreiranje rasporeda, utvrđene su i zainteresovane strane za finalni raspored: studenti, nastavno osoblje, nenastavno osoblje i domenski ekspert.

3.2. Izvorna dokumenta

Jedan od najvećih izazova koji je primećen već u početnim fazama istraživanja jeste nekonzistentnost

podataka koji potiču iz različitih izvora. Različiti sektori koji učestvuju u organizovanju nastave vode sopstvene evidencije podataka između kojih ne postoji sinhronizacija.

Najznačajniji dokument za kreiranje rasporeda bio je trenutni raspored nastave, prikazan na Slici 3.1. Zbog njegove složene i nepregledne strukture, odlučeno je da će trenutni Excel dokumenti koji sadrži raspored u novom sistemu biti zamenjen bazom podataka.

Učionica	06:30	06:45	07:00	07:15	07:30	07:45	08:00	08:15	08:30	08:45	09:00	09:15	09:30	09:45	10:00	10:15
A1				SVI	SVI	SVI	SVI	SVI	SVI	SVI	SVI	SVI	SVI	SVI	SVI	SVI
A2																
A3																
A4																
101																
102																
103																
104																
105																
106																
107																
108																
108A																
109																
109A																
201																
202																
203																
204																
205																
206																

Slika 3.1. Isečak Excel dokumenta rasporeda nastave

3.3. Slučajevi korišćenja

Rad opisuje komponente koje čine deo većeg sistema za kreiranje i optimizaciju rasporeda. Sistem podržava jedan tip korisnika, administratora, koji je odgovoran za rukovanje ulaznim podacima potrebnim za izradu rasporeda kao i za sam proces kreiranja rasporeda. Komponente imaju tri glavna cilja koja se integrišu u sistem na različitim nivoima:

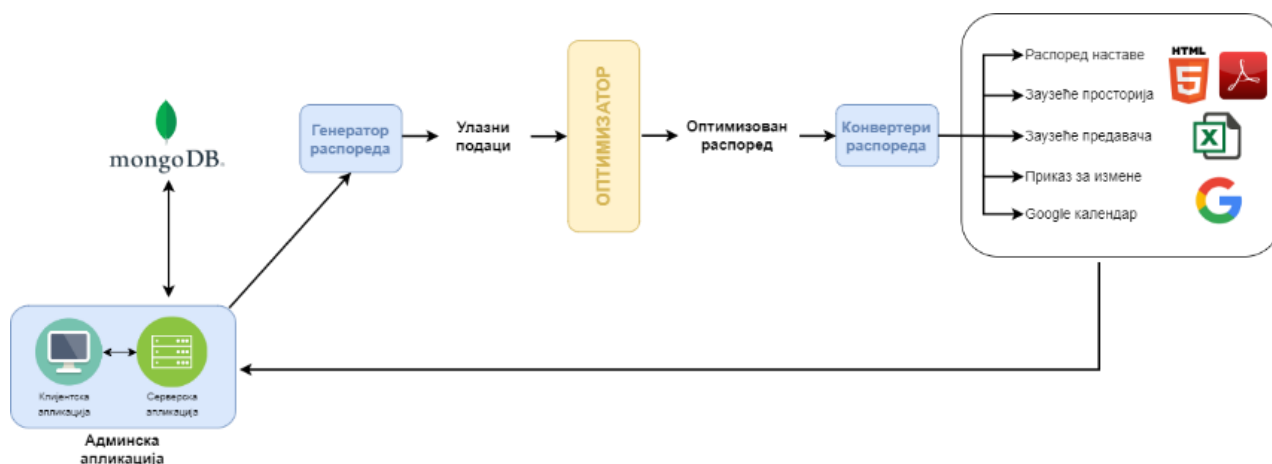
1. Priprema ulaznih podataka:
 - a. Ekstrakcija podataka iz dokumenata bitnih za kreiranje rasporeda,
 - b. Transformacija podataka,
 - c. Učitavanje podataka u JSON formatu.
2. Generisanje termina za raspored na osnovu transformisanih podataka u format kompatibilan sa optimizator komponentom,
3. Izvoz rezultata optimizacije u različite formate prilagođene potrebama svake od zainteresovanih strana:
 - a. Izvoz rasporeda za studente,
 - b. Izvoz rasporeda za predavače,
 - c. Izvoz rasporeda za prostorije,
 - d. Izvoz rasporeda za dalje izmene,
 - e. Izvoz zauzeća predavača u odgovarajućem formatu za uvoz u Google kalendar.

3.4. Arhitektura sistema

Arhitektura sistema prikazana je na Slici 3.2. Rad opisuje komponente sistema koje su međusobno nezavisne i obuhvataju obradu različitih tipova dokumenata i podataka.

Odabrana je odgovarajuća arhitektura kako bi sistem bio efikasan i funkcionalan. Zbog prirode problema, arhitektura sistema koristi više skripti napisanih u Python [9] programskom jeziku koji je izabran zbog dinamičkog tipiziranja i podrške za rad sa Excel i CSV dokumentima kroz biblioteku Pandas [10].

Tokom razvoja korišćeno je Jupyter Notebook [11] okruženje, dok su za produkciju razvijene Python skripte.



Slika 3.2. Arhitektura integracionog sistema

Za upravljanje zavisnostima koristi se alat pip [12], a za generisanje PDF dokumenata alat wkhtmltopdf [13].

3.5. Arhitektura integracionog sistema

Dizajniran je integracioni sistem koji omogućava konzistentno kreiranje rasporeda nastave, od obrade ulaznih podataka, preko njihove optimizacije, do generisanja rasporeda nastave.

Komponente koje čine integracioni sistem su:

- skladište podataka,
- administratorska aplikacija,
- generator rasporeda,
- optimizator i
- konverteri rasporeda.

4. SPECIFIKACIJA SISTEMA

Sistem se sastoji iz tri nezavisne komponente koje su implementirane upotrebom Python programskog jezika. Celokupan izvorni kod komponenti je javno dostupan¹. U daljem tekstu poglavlja opisana je svaka od komponenti, kao i tok podataka kroz njih.

4.1. Struktura parser komponente

Parser komponenta je najkompleksnija od tri opisane u radu i zadužena je za obradu podataka. Kompleksnost je posledica velikog broja ulaznih dokumenata koje je potrebno parsirati i uskladiti. Proces parsiranja je logički podeljen u 6 koraka, gde se u svakom koraku obrađuje određeni set ulaznih dokumenata i njihov izlaz se ulančava u krajnju strukturu podataka koja je optimizovana za skladištenje u dokument orijentisanoj bazi podataka. Održavanje skladištenih podataka se vrši kroz administratorsku aplikaciju.

4.2. Struktura generator komponente

Generator termina je središnja komponenta u toku podataka kroz sistema optimizacije. Nakon parsiranja podataka, na osnovu realizacije nastave, potrebno je kreirati termine za odabrani semestar. Prilikom generisanja ter-

mina poštovana su pravila propisana Pravilnikom fakulteta [14].

Zbog velikog broja studenata i ograničenog kapaciteta prostorija, potrebno je rešiti problem optimalnog kombinovanja grupa u cilju bolje iskorisćenosti prostora. Ovaj problem je poznat kao problem pakovanja u prostoru [15] i spada u grupu NP-teških problema [16][17]. Generisanje termina rasporeda zahteva kreiranje algoritma za kombinovanje studentskih grupa i rešavanje opisanog NP-teškog problema.

Generisani termini koji su izlaz ove komponente se dalje prosleđuju optimizatoru koji je zadužen za dodeljivanje prostorija i vremenskog intervala svakom terminu.

4.3. Struktura konverter komponente

Konverter komponenta je najjednostavnija među tri navedene komponente. Njena uloga je kreiranje prikaza rasporeda u odgovarajućim formatima za sve zainteresovane strane. Ulaz predstavlja raspored kreiran od strane optimizatora, sa informacijama o studijskom programu, grupama, predmetima, predavačima, učionicama i vremenskim intervalima. Konverter komponenta pretvara ovu ulaznu strukturu, predstavljenu JSON objektom, u dokumente pogodne za pregled od strane studenata, predavača i drugih zainteresovanih strana.

5. ZAKLJUČAK

U radu je predstavljen razvoj i implementacija komponenti za podršku sistema za kreiranje rasporeda nastave na Fakultetu tehničkih nauka u Novom Sadu. Motivacija za razvoj sistema je bolja organizacija nastave i veće zadovoljstvo studenata i nastavnog osoblja. Sistem podržava različite scenarije korišćenja i osigurava konzistentnost u kreiranju rasporeda, od obrade ulaznih podataka, preko transformacije podataka i generisanja termina rasporeda, do kreiranja izvoza pogodnih za sve zainteresovane strane. Glavna funkcionalnost je obrada podataka iz različitih izvora korišćenjem ETL procesa, omogućavajući lakšu integraciju sa postojećim informacionim sistemom fakulteta.

Za buduća unapređenja, predlaže se integracija sistema sa adapterima za dinamičko preuzimanje podataka, povećavajući njegovu primenu i potencijal za korišćenje u različitim obrazovnim ustanovama. Takođe, proširenje

¹ <https://github.com/evaj10/raspored-parser>

sistema novom funkcionalnošću za izmenu kreiranog rasporeda, integracijom sa kalendarskim rešenjima, olakšalo bi rukovanje kreiranim terminima u okviru rasporeda nastave.

6. LITERATURA

- [1] Extract, Transform, Load процес, <https://www.ibm.com/topics/etl> [датум приступа мај 2024]
- [2] Интернационално такмичење у оптимизацији распореда наставе за 2019. годину, <https://www.itc2019.org/home> [датум приступа мај 2024]
- [3] Müller, T., Rudová, H., & Müllerová, Z. (2024). Real-world university course timetabling at the International Timetabling Competition 2019. *Journal of Scheduling*, 1-21.
- [4] Janković, E., & Luburić, N. (2023). SISTEM ZA OPTIMIZACIJU RASPOREDA NASTAVE FAKULTETA TEHNIČKIH NAUKA. *Zbornik radova Fakulteta tehničkih nauka u Novom Sadu*, 38(12), 1679-1682.
- [5] Vassiliadis, P. (2009). A survey of extract–transform–load technology. *International Journal of Data Warehousing and Mining (IJDWM)*, 5(3), 1-27.
- [6] Simitsis, A., Vassiliadis, P., & Sellis, T. (2005, April). Optimizing ETL processes in data warehouses. In *21st International Conference on Data Engineering (ICDE'05)* (pp. 564-575). Ieee.
- [7] Yulianto, A. A. (2019). Extract transform load (ETL) process in distributed database academic data warehouse. *APTİKOM Journal on Computer Science and Information Technologies*, 4(2), 61-68.
- [8] Vera, H., Boaventura, W., Holanda, M., Guimaraes, V., & Hondo, F. (2015, September). Data modeling for NoSQL document-oriented databases. In *CEUR Workshop Proceedings* (Vol. 1478, pp. 129-135).
- [9] Званична документација Python програмског језика, <https://www.python.org/doc/> [датум приступа мај 2024]
- [10] Званична документација Pandas Python библиотеке, <https://pandas.pydata.org/docs/> [датум приступа мај 2024]
- [11] Званична документација Jupyter Notebook софтвера, <https://docs.jupyter.org/en/latest/> [датум приступа мај 2024]
- [12] Званична документација pip алата за руковање зависностима за Python програмски језик, <https://pip.pypa.io/en/stable/> [датум приступа мај 2024]
- [13] Званична документација wkhtmltopdf алата за генерисање PDF докумената на основу HTML докумената, <https://wkhtmltopdf.org/docs.html> [датум приступа мај 2024]
- [14] Правилник о изради распореда наставе, ФТН, <http://www.ftn.uns.ac.rs/520569743/pravilnik-o-izradi-rasporeda-casova-nastave-i-rasporeda-ispitana-fakultetu-tehnickih-nauka-od-31-01-2019-> [датум приступа мај 2024]
- [15] Lodi, A., Martello, S., & Vigo, D. (2002). Recent advances on two-dimensional bin packing problems. *Discrete Applied Mathematics*, 123(1-3), 379-396.
- [16] Knuth, D. E. (1974). Postscript about NP-hard problems. *ACM SIGACT News*, 6(2), 15-16.
- [17] Lodi, A., Martello, S., & Vigo, D. (2002). Recent advances on two-dimensional bin packing problems. *Discrete Applied Mathematics*, 123(1-3), 379-396.

Kratka biografija:



Ksenija Prčić рођена је 16. марта 1998. године у Новом Саду. Као носилац дипломе „Vuk Karadžić“ 2017. године завршава гимназију „Јован Јовановић Змај“, након чега уписује Факултет техничких наука, смер Softversko inženjerstvo i informacione tehnologije. По завршетку студија у року, уписује мастер академске студије и исте завршава 2024. године.

HORIZONTALNO SKALIRANJE K8S POD-OVA POMOĆU KEDA AUTOSCALER-A**HORIZONTAL SCALING K8S PODS USING KEDA AUTOSCALER**Marko Stanić, *Fakultet tehničkih nauka, Novi Sad***Oblast - SOFTVERSKO INŽENJERSTVO I INFORMACIONE TEHNOLOGIJE**

Kratak sadržaj - U radu je prikazana implementacija upravljanja konfiguracijom za automatsko skaliranje Kubernetes pod-ova zasnovano na broju poruka u SQS-u. Opisan je princip rada Kubernetes kontejner orkestratora sa KEDA autoscaler-om uz integraciju sa Python bibliotekom za upravljanje cloud servisima i K8s resursima.

Ključne reči: Kubernetes, KEDA, SQS, horizontalno skaliranje

Abstract - The paper presents the implementation of a configuration management software for K8s pods and auto-scaling based on number of messages in SQS. It is described how to use KEDA autoscaler with K8s container orchestrator and Python library for cloud and K8s resource management.

Keywords: Kubernetes, KEDA, SQS, horizontal scaling

1. UVOD

U savremenom softverskom inženjeringu, skaliranje aplikacija postaje sve važniji aspekt koji utiče na njihovu pouzdanost i performanse. Kada aplikacije rastu i korisnički zahtevi se povećavaju, potrebno je da se skalira infrastruktura koja podržava aplikaciju.

Dva osnovna pristupa skaliranju su vertikalno i horizontalno skaliranje. Horizontalno skaliranje podrazumeva dodavanje novih mašina u postojeću infrastrukturu, što omogućava raspoređivanje opterećenja na više mašina i obezbeđuje bolju otpornost na kvarove [1]. U ovom radu će fokus biti horizontalno skaliranje *Kubernetes pod-ova* uz pomoć KEDA autoscaler-a i metrika iz AWS SQS servisa i broja poruka u redu.

2. TEHNOLOGIJE**2.1. Kubernetes**

Kubernetes (K8s) [2] je open-source platforma za automatizovanu upravu kontejnerima.

Kontejnerizacija je tehnologija koja omogućava pakovanje aplikacija i njenih zavisnosti u izolovane kontejnere, što omogućava da se aplikacija izvršava u bilo kojoj okolini koja podržava kontejnere. Kubernetes pruža alate i mehanizme za automatizovanu upravu kontejnerima i njihovim

resursima, kao što su skaliranje, *load balancing*, *selfhealing* i *rollout* aplikacija. Kubernetes ima nekoliko ključnih komponenti koje omogućavaju automatizovanu upravu kontejnerima. Kontejneri se pokreću u Kubernetes klasteru, koji se sastoji od čvorova (eng. *nodes*) i kontrolne ravni (eng. *control plane*). Svaki čvor je fizička ili virtuelna mašina koja izvršava kontejnere, dok kontrolna ravan sadrži alate za upravljanje čvorovima i kontejnerima.

2.2. KEDA

KEDA (Kubernetes-based Event Driven Autoscaler) [3] je open-source komponenta za automatsko skaliranje Kubernetes aplikacija (*Deployment* i *ReplicaSet*) na osnovu različitih događaja i metrika koje se dešavaju u Kubernetes klasteru ili u drugim cloud servisima.

KEDA podržava veliki broj događaja i metrika, uključujući AWS SQS [4], Azure Service Bus, Apache Kafka, RabbitMQ i brojne druge.

2.3. SQS

Amazon Simple Queue Service (SQS) je *fully-managed* servis u okviru Amazon Web Services (AWS) platforme, koji omogućava korisnicima da lako kreiraju i upravljaju redovima poruka u cloud okruženju. SQS je dizajniran da pomogne korisnicima da odvoje komponente svojih aplikacija, tako što će omogućiti različitim delovima aplikacije da komuniciraju asinhrono, bez potrebe za direktnom vezom.

SQS je skalabilan i otporan na greške, što ga čini idealnim izborom za različite aplikacije koje se zasnivaju na cloud tehnologijama i koje moraju da podrže veliki broj korisnika i visok nivo opterećenja. SQS je dostupan u dve verzije: standardna verzija i FIFO verzija (*First-In-First-Out*). Standardna verzija podržava veliki broj poruka u redu, sa minimalnim kašnjenjem u isporuci poruka, dok FIFO verzija omogućava garanciju da će poruke biti isporučene tačno onim redosledom kojim su postavljene u red.

SQS pruža fleksibilnost u upravljanju porukama, uključujući mogućnost slanja, primanja, brisanja i promene prioriteta poruka. Takođe omogućava i definisanje pravila za ponovno slanje poruka koje nisu uspešno isporučene.

SQS se može integrisati sa drugim AWS servisima, kao što su Amazon S3, Amazon EC2, AWS Lambda i mnogi drugi, kako bi se omogućilo jednostavno upravljanje porukama i stvaranje pouzdanih, skalabilnih i otpornih aplikacija u cloud okruženju.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Željko Vuković, docent.

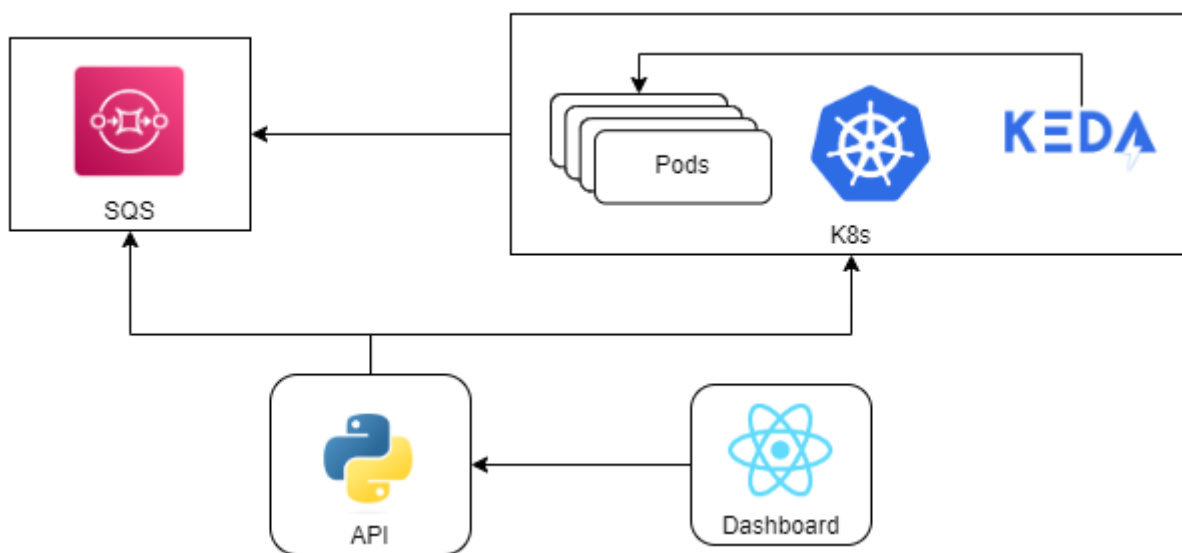
3. ARHITEKTURA

Na slici 1 nalaze se sledeće komponente:

- K8s klaster na kome će biti *deploy*-ovana aplikacija koja će se dinamički skalirati. Podrška za *Kubernetes* je dostupna na gotovo svim većim *cloud* provajderima. Za testiranje u lokalnom okruženju korišćen je *Minikube* [5].
- *KEDA autoscaler* gde će se definisati skalirajući resursi (*Scalers*) koji se koriste za prikupljanje metrika o opterećenju aplikacije. U ovom slučaju to će biti broj poruka koji se nalazi u redu (*queue*)
- Za korišćenje SQS servisa je neophodan AWS korisnički nalog, tako da alternativna za njegovu upotrebu može biti *ElasticMQ* [6] koji ima isti interfejs kao i SQS. To znači da

je moguće testirati lokalno bez ikakvih troškova, a za prelazak na SQS je potrebno samo izmeniti odgovarajuću konfiguraciju.

- *Backend* je *REST Api* koji pruža interakciju sa *Amazon SQS* i K8s klasterom. Sve dostupne funkcionalnosti mogu se pregledati kroz *OpenAPI* dokumentaciju na */docs* putanji.
- *Dashboard* je korisnički interfejs za komunikaciju sa prethodnim *REST* servisom, koji omogućava grafički pregled, dodavanje novih kao i brisanje redova. Takođe u svaki red je moguće dodati navedeni broj poruka ili obrisati sve (*purge*). Kreiranje *Scaled Object* resursa je dostupno popunjavanjem forme.



Slika 1. Arhitektura Sistema

4. IMPLEMENTACIJA

Za pokretanje *minikube* klastera, na *Windows* operativnom sistemu potrebno je izvršiti komandu *minikube start* u *PowerShell*-u. Status je moguće proveriti kroz *Docker Desktop* gde će se pojaviti kontejner sa nazivom *minikube* u stanju *running* (slika 4.1.2) ili kroz *Lens* - radno okruženje za K8s. Instalacija *KEDA autoscaler*-a se vrši pomoću *helm* [7] menadžera paketa za *Kubernetes*. Pravila skaliranja se mogu definisati na osnovu željene maksimalne i minimalne veličine broja replika aplikacije. Na primer, ako se želi imati minimalno dve i maksimalno pet replika aplikacije u *Kubernetes* klasteru, mogu se definisati pravila skaliranja koja će skalirati aplikaciju prema gore kada broj poruka u redu poraste iznad određenog praga, a skalirati aplikaciju prema dole kada broj poruka u redu padne ispod nekog drugog praga.

KEDA ima dve različite faze tokom procesa automatskog skaliranja:

Faza aktiviranja (ili deaktiviranja) je trenutak kada *KEDA* operator mora da odluči da li je opterećenje takvo da treba skalirati od nule ili na nulu. Ova akcija je bazirana na rezultatu funkcije *IsActive* i primenjuje se samo na 0-1 skaliranje. Postoje slučajevi kada su pravila za aktiviranje

potpuno drugačija, na primer sa *Prometheus* skalarom gde vrednosti mogu ići od -x do x.

Faza skaliranja je trenutak kada postoji jedna instanca i tada *Horizontal Pod Autoscaler (HPA)* donosi odluke zasnovane na konfiguraciji koja je generisana na osnovu podataka u *ScaledObject* resursu. Ova faza se primenjuje na 1-n skaliranje.

Web API koji služi za upravljanje redovima i *Kubernetes* resursima je kreiran uz pomoć *FastAPI* radnog okvira u *Python* programskom jeziku.

Za kreiranje novog reda je potrebno proslediti samo naziv u formi koja je prikazana na slici 2.

Akcije koje je moguće izvršiti nad redom su: dodavanje poruka, brisanje poruka, brisanje i kreiranje *ScaledObject* resursa (slika) za koje je potrebno izabrati *deployment* iz odgovarajućeg *namespace*-a.

Nakon uspešnog kreiranja *ScaledObject*-a skaliranje se aktivira i status je moguće videti na slici 4 gde je prikazan *progress bar* koji predstavlja odnos trenutnog i maksimalnog broja instanci, koji su definisani za *ScaledObject*.

Slika 2. Tabela formom za dodavanje reda

Slika 3. Forma za kreiranje ScaledObject-a

Slika 4. Status skaliranja

5. ZAKLJUČAK

U ovom radu je predstavljen primer horizontalnog skaliranja aplikacija u *Kubernetes* klasteru uz pomoć *KEDA autoscaler*-a. Objašnjen je princip rada *Kubernetes*-a kao kontejner orkestratora i način upotrebe *autoscaler*-a zasnovano na događajima, tačnije broja poruka u AWS SQS. Time je omogućena brža obrada podataka ukoliko je to zahtevano na osnovu definisane konfiguracije, a sa druge strane ušteda resursa kada nije neophodno da aplikacija bude aktivna.

Horizontalno skaliranje se može primeniti u nekoliko slučajeva: za povećanje performansi ukoliko nije moguće dodati više resursa na jednoj mašini (fizičkoj ili virtualnoj), kada je potrebno omogućiti maksimalnu dostupnost, u slučaju da dođe do otkaza neke mašine, ostale nastavljaju da rade bez zastoja.

6. LITERATURA

- [1] IBM Data Science - Best Practices
<https://ibm.github.io/data-science-best-practices/scaling.html>

- [2] Kubernetes <https://kubernetes.io/>

- [3] KEDA <https://keda.sh/>

- [4] SQS <https://aws.amazon.com/sqs/>

- [5] Minikube <https://minikube.sigs.k8s.io/docs/start/>

- [6] ElasticMQ <https://github.com/softwaremill/elasticmq>

- [7] Helm <https://helm.sh/>

Kratka biografija:



Marko Stanić rođen je 31.5.1997. godine u Šapcu. Završio je osnovne akademske studije 2020. godine na Fakultetu tehničkih nauka. Upisao je master studije iste godine, studentski program Softversko inženjerstvo i informacione tehnologije

kontakt: marko.stanic97@icloud.com

АНАЛИЗА ЕФЕКТА СРЕДСТАВА ЗА УСПОРАВАЊЕ САОБРАЋАЈА**ANALYSIS OF THE EFFECTS OF SPEED RETARDERS**Милан Милановић, Вук Богдановић, *Факултет техничких наука, Нови Сад***Област – САОБРАЋАЈ И ТРАНСПОРТ**

Кратак садржај – У оквиру рада извршена је анализа ефеката примене опреме за успоравање саобраћаја на територији Инђије и Новог Сада. Измерене су брзине возила приликом преласка преко различитих типова успоривача саобраћаја. Те брзине су међусобно упоређене, а затим су упоређене са брзинама које су остварене на локацији фиксне радарске камере и на локацији где не постоји никакво средство за успоравање саобраћаја. Приказани су ефекти анализираних средстава и одабрано је најделотворније решење за успоравање саобраћаја.

Кључне речи: Брзина, Успоривачи, Упоредна анализа

Abstract – The paper analyzes the effects of speed retarders on territory Indjija and Novi Sad. The speed of vehicles were measured when crossing different types of traffic retarders. Those speeds were compared with each other and then they were compared with the speeds achieved at the location of the fixed radar camera and at the location where there is no means to slow down the traffic. The effects of the analyzed means are shown and the most effective solution for slowing down the traffic is selected.

Keywords: Speed, Retarders, Comparative analysis

1. УВОД

Предмет рада представља анализа утицаја примене различитих успоривача на уличној мрежи Инђије и Новог Сада на брзину.

Успоривачи саобраћаја представљају техничка средства којима се возачи приморавају на кретање мањом брзином. Служе за успоравање, односно смањење брзине кретања моторних возила на појединим деловима путне и уличне мреже.

За потребу израде рада вршена су мерења брзине на следећим врстама успоривача: гумени (пластични) вертикални успоривач висине 7 cm, асфалтирани вертикални успоривач висине 7 cm и успоривачи платформе.

Поред тога, извршено је и мерење брзине на месту постављене фиксне радарске камере, која се такође може сматрати средством, односно техничким решењем које се поставља са циљем успоравања саобраћаја. Такође, вршено је мерење и на месту где је некад био постављен гумени успоривач саобраћаја, који је из одређених разлога уклоњен.

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Вук Богдановић, ред. проф.

Циљ рада јесте да се ефекти утицаја успоривача на брзину, кроз анализу брзина возила приликом преласка преко одређених типова вертикалних успоривача, као и ефеката примене камера за детектовање брзине.

Ефекти утицаја успоривача на брзину утврдиће се упоредном анализом брзина на месту где је некада био постављен успоривач који је уклоњен, са брзинама на постојећем успоривачу.

У циљу анализе ефеката утицаја различитих успоривача на смањење брзине, карактеристичне вредности брзина на вертикалном успоривачу упоређени су са брзинама возила на одсеку на коме се налазе фиксне радарске камере. Резултати анализе могу се користити приликом избора оптималног типа успоривача на уличној мрежи.

2. ТЕОРИЈСКЕ ОСНОВЕ

Поводи за употребу успоривача су различити, најчешће је то потреба да се кроз смањење брзине утиче на смањење броја саобраћајних незгода, односно на жестину и умањење последица. Поред тога, успоривачи се често користе за ограничавање или отежавање приступа свих или појединих врста возила, односно за демотивацију возача да се крећу појединим путним правцима или деловима уличне мреже.

Иако успоривачи представљају практично најефикаснији начин успоравања саобраћаја и смањења брзине, они се не могу поставити на сваком делу путне и уличне мреже, чак и у ситуацијама повећане угрожености учесника у саобраћају. Наиме, успоривачи генерално, у зони њихове примене, утичу на смањење комфора вожње, повећања буке и вибрација растресања терета, успоравање или отежавање приступа возилима хитних служби итд. [1].

Под техничким средствима за успоравање саобраћаја на путу подразумевају се:

1. физичке препреке и то:
 - вештачка избочина,
 - плато (платформа),
 - сужење коловоза,
 - хоризонтално скретање коловоза;
2. шуштећа трака;
3. вибрациона трака

Димензије вештачких избочина зависе од ограничене брзине кретања возила на посматраној деоници и дате су у табели 1 [1].

Табела 1. Димензије вештачких избочина у зависности од ограничења брзине

Ранг	Ограничена брзина кретања (km/h)	Висина вештачке избочине – х (cm)	Ширина вештачке избочине – д (cm)
1	50	3	60
2	40	5	90
3	30	7	120

3. МЕТОДОЛОГИЈА

Методологија рада обухвата:

- избор типова успоривача погодних за истраживање,
- избор локације на терену, и
- спровођење кратке анкете са станарима подручја на којима је вршено истраживање.

3.1. Начин прикупљања података

Подаци су прикупљени тако што је мерена брзина возила приликом нагаза на успоривач, односно непосредно пре тога, што је уједно и минимална брзина на коју возило успорава да би прешло преко вештачке избочине. Наиме, возило пре успоривача успорава на брзину којом гази вештачку избочину и прелази преко ње, након чега наставља да се креће убрзано док не постигне жељену брзину.

Узорак возила чија је брзина мерена износио је око 150 возила, у зависности од локације.

Истраживањем нису обухваћена возила која су се кретала у условима засићеног тока односно возила која су се кретала у колони већ само возила која су се кретала у условима слободног тока. Разлог тога је избегавања мерења брзине у ситуацијама када возач прилагођава брзину условима тока, а не условима постојања успоривача саобраћаја. Такође, у обзир нису узете ни брзине возила која су се непосредно пре или непосредно после локације мерења укључила односно искључила из саобраћајног тока. У случајевима када се непосредно након локације мерења налазио пешачки прелаз у близини кога су се налазили пешаци, брзине возила нису узете у обзир. Разлог тога је да су возачи потенцијално прилагодили брзину пешацима на пешачком прелазу а не успоривачу саобраћаја.

На месту уклоњеног лежећег полицајца, у улици Ђорђа Војновића у Инђији, спроведена је кратка анкета са станарима околних кућа о томе да ли уклоњени лежећи полицајац треба да се врати.

Током спровођења истраживања, становници из непосредне близине локације су самоиницијативно прилазили и давали своја мишљења везана за субјективни осећај безбедности саобраћаја.

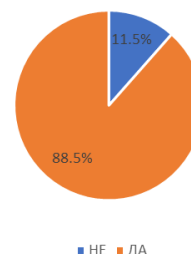
Већина становника у посматраном подручју навела су да ту возила „јуре“, да има нико не поштује ограничење брзине (које је на том делу 50 km/h), да је страшна бука и гужва, верујући да је све то последица преокорачења брзине. Поводом тога, дошло се на идеју да се међу становницима спроведе кратка анкета.

Анкетом је обухваћен узорак од 52 грађана.

У оквиру анкете испитаницима је постављено само по једно питање: „Да ли мислите да је потребно вратити лежећи полицајац?“

На слици 1 приказан је графикон са резултатима спроведене анкете. Чак 88,5% становника који живе у околини успоривача мисли да је потребно вратити лежећи полицајац док се свега 11,5% њих не слаже са тим. То би према приручнику окружног одељења за саобраћај у САД био довољан разлог за постављање успоривача саобраћаја јер је преко 75% станара тог дела улице гласало за његово враћање [3].

Да ли мислите да је потребно вратити лежећи полицајац?



Слика 1 Резултати анкете

Треба имати на уму да је лежећи полицајац за којег 88,5% грађана мисли да треба да се врати био висине 7 cm што према Правилнику о техничким средствима за успоравање саобраћаја на путу одговара ограничењу брзине од 30 km/h а не 50 km/h која је у тој улици [4].

3.2. Локације на којима је вршено истраживање

Локације на којима је вршено истраживање се налазе на територији Града Новог Сада и на територији Општине Инђија. Разлог избора ових градова је што Инђија и Нови Сад користе различите врсте успоривача саобраћаја.

У Новом Саду се у већој мери користе асфалтирани вертикални успоривачи док се на уличној мрежи Инђије користе гумени вертикални успоривачи. Поред тога, у Новом Саду постоје издигнуте платформе за успоравање саобраћаја, а у Инђији се налазе фиксне радарске камере и на двотрачним саобраћајницама на којима се радило целокупно истраживање.

У Инђији је истраживање рађено у улицама: Ђорђа Војновића и Бошка Бухе где се налазе гумени вертикални успоривачи истих димензија (ширине 1,2 метра и висине 7 центиметара) и у улици Цара Душана где се налази фиксна радарска камера. Поред тога, у улици Ђорђа Војновића, рађено је мерење брзине и спроведена кратка анкета на другој локацији на којој је некада постојао вертикални успоривач који је уклоњен.

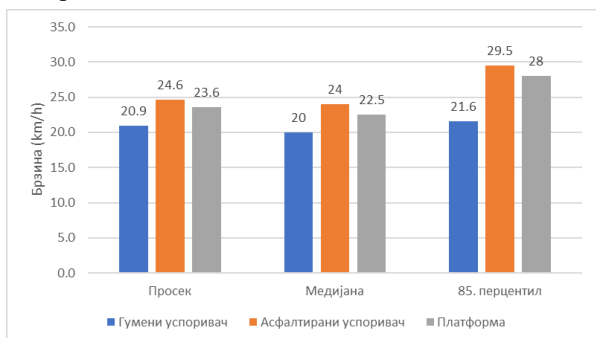
У Новом Саду је рађено истраживање на две локације на булевару Деспота Стефана. Од тога је једно било на издигнутој платформи на којој се налазе два пешачка и један бициклистички прелаз код Лиманског парка а друго код студентског дома Цар Лазар на асфалтираном вертикалном успоривачу. Следећа локација на којој је рађено истраживање је улица 1300 каплара где је мерена брзина на асфалтираном вертикалном успоривачу.

Поред тога истраживање је рађено у свеже реновираној Крајишкој улици у Петроварадину. У току прве половине 2023. године у овој улици је спроведена реконструкција и убачене су платформе

на којима се налази пешачки прелаз као средства за успоравање саобраћаја.

4. РЕЗУЛТАТИ ИСТРАЖИВАЊА

На слици 2 је приказано поређење све 3 врсте успоривача на којима је вршено истраживање. Поређене су на основу 3 основна параметра а то су: просечна брзина, медијална брзина и 85. перцентил брзине. Пошто је истраживање обухватило по два успоривача од сваке врсте, за потребе дијаграма је узета аритметичка средина просечне брзине, медијалне брзине и 85. перцентила два успоривача исте врсте.



Слика 2. Поређење све три врсте успоривача

Иако су се асфалтирани вертикални успоривачи налазили у зони ограничења брзине на 30 km/h, а гумени и платформе у насељу где влада опште ограничење брзине од 50 km/h, резултати истраживања указују на чињеницу да су највеће брзине остварене управо приликом преласка возила преко асфалтираног вертикалног успоривача, иако су сва 3 облика успоривача висине 7 cm.

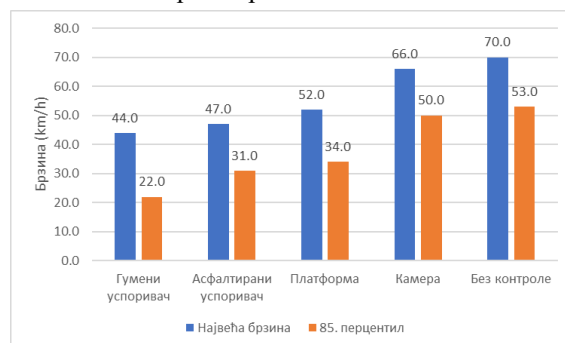
Резултати истраживања упућују на закључак да ограничење брзине у околини успоривача није имало утицаја на брзину преласка преко њега. Брзина преласка преко успоривача саобраћаја зависи пре свега од облика, врсте и димензија.

Убедљиво најниже вредности и просечне брзине и медијалне брзине и 85. перцентила максималне брзине су забележене код возила која су прелазила преко гуменог вертикалног успоривача. Највеће брзине у све 3 категорије су забележене код асфалтираног вертикалног успоривача. Резултати истраживања показали су да је највећа брзина измерена на платформи на булевару Деспота Стефана у све 3 анализирани категорије. Међутим, с обзиром да је узета у обзир аритметичка средина брзина на платформама у булевару Деспота Стефана и Крајишкој улици, средња вредност брзине је умањена. Наиме, вредности брзина измерених у Крајишкој улици су утицале на смањење вредности просечне брзине, медијалне брзине и брзину 85. перцентила, која је уједно и максимална брзина за категорију платформи.

Слика 3 приказује односе између максималних забележених брзина на различитим локацијама на којима је вршено истраживања и брзине 85. перцентила.

На слици су приказане упоредне вредности брзина на свим карактеристичним пресецима где је мерена брзина а то су: гумени успоривач („лежећи

полицајац“), асфалтирани успоривач („лежећи полицајац“), платформа, фиксна радарска камера и место где је некада постојао успоривач, па тренутно представља локацију на којој се не управља брзинама. Што се тиче асфалтираних успоривача, гумених успоривача и платформи, као што је наведено, брзина је мерена на две локације. Из разлога што је рађено на по две локације, узета је у обзир локација где је забележена већа брзина, односно максимална брзина која је забележена по типу успоривача. На локацији на којој је забележена већа максимална брзина, утврђен је и већи 85. перцентил. За локацију са радарском камером и локацију без контроле, узета је максимална измерена брзина.



Слика 3. Поређење брзина по мерама за успоравање саобраћаја

Најмање брзине генерално се остварују приликом преласка возила преко гуменог вертикалног успоривача. Ту је максимална забележена брзина била 44 km/h док 85. перцентил износи само 22 km/h. Значи да су брзине изнад те вредности од 22 km/h биле ретке.

После гуменог успоривача најмање брзине се постижу приликом преласка преко асфалтираног успоривача која износи 47 km/h за максималну и 31 km/h за 85. перцентил. Треба напоменути да је једино асфалтирани успоривач био на локацији где је ограничење брзине 30 km/h, али су на њему ипак постигнуте веће брзине него код гумених успоривача који се налазе на делу уличне мреже са ограничењем брзине од 50 km/h.

Разлике у брзинама на истом типу успоривача највеће су на платформама. Наиме, да се посматрала само платформа у Крајишкој улици, максимална измерена брзина би била на другом месту, а асфалтни успоривач на трећем месту на слици број 3. Међутим, на платформи у булевару Деспота Стефана су забележене значајно веће вредности брзина у односу на Крајишку улицу. Наиме, максимална измерена брзина у приликом преласка возила преко платформе на булевару Деспота Стефана износи 52 km/h за максималну брзину и 34 km/h за 85. перцентил брзина.

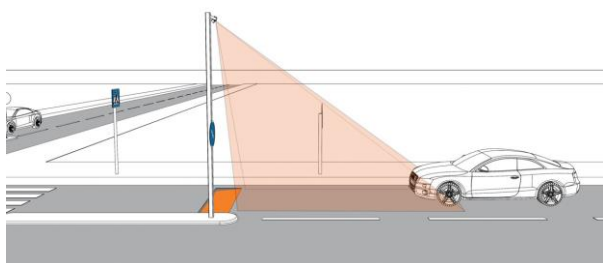
Очекивано, највеће брзине су измерене на месту где не постоји контрола брзине, односно радарска камера или техничко средство за успоравање саобраћаја. Резултати истраживања су показали да било која техничка средства за успоравање саобраћаја или контрола уз помоћ камера утичу на возаче да смање брзину, односно возе спорије. Међутим, 85. перцентил брзине на местима без контроле, односно

на локацији без успоривача или камере, брзине износи 53 km/h, што је само 3 km/h више од општег ограничења. Максимална брзина на овим локацијама је 70 km/h и то је уједно и највећа брзина снимљена приликом истраживања. Међутим, укупни резултати мерења показали су да релативно мали број возача прекорачује опште ограничење за више од 15%.

5. ПРЕДЛОГ МЕРА

У скандинавским земљама се све више користи системи за успоравање саобраћаја који реагује само у ситуацијама када детектује возила која прекорачују брзину [2]. Резултати истраживања су показали да прекорачење брзине на уличној мрежи не остварују сви возачи већ појединци. У улици Ђорђа Војновића у Инђији су постављени успоривачи саобраћаја ради смањења прекорачења брзине, а резултати истраживања су показали да је да је просечна брзина свих возила која су прошла локацијом где не постоји контрола брзине, за време истраживања испод ограничења брзине, односно 46.7 km/h. 85. перцентил брзине је тек нешто изнад ограничења, али унутар толеранције од ± 4 km/h који су у Шведска користи као усвојена толеранција брзине у оваквим ситуацијама. Познато је да физичке препреке за успоравање саобраћаја имају негативне утицаје на склопове на возилима, посебно у дужем временском периоду, што се избегава применом савремених система за успоравање.

Дакле, резултати истраживања су потврдили да мали проценат возача значајније крши ограничење брзине. У том смислу потребно је да се применом различитих мера контроле открију и санкционишу возачи прекршиоци или да се успоре њихова возила, применом савремених система за успоравање. Наиме, нерационално је због малог броја несавесних возача постављати успориваче саобраћаја, којима ће се успоравати комплетан саобраћајни ток. Из тог разлога је потребно користити савремене, интелигентне системе за успоравање саобраћаја, који ће за разлику од класичних успоривача, успоравати само возила која прекорачују брзину. Управо зато се у примени нашао систем приказан на наредној слици, који се активира само на појединачна возила док возилима која се крећу у границама дозвољене брзине пружа могућност неометаног проласка.



Слика 4. Паметни успоривач саобраћаја [2]

Овај систем се састоји од фиксног радара и платформе која је у равни са коловозном, али која има могућност спуштања за неколико центиметара. Функционише тако што радар детектује брзину возила у доласку. Ако је брзина мања од ограничења, помична платформа остаје у равни коловоза и пружа

неометан пролазак возилу. Ако радар детектује возило које је прекорачило брзину, систем спушта платформу неколико центиметара ниже и ствара успоривач саобраћаја по принципу улегнућа на путу. Тада возило прелази преко успоривача и принуђено је спушта брзину.

Резултат у оба случаја је утицај на смањење брзине возилима која не поштују ограничења [2]. Овај систем захтева већа улагања, али омогућава оптималан протока возила, односно не успорава возила без стварне потребе.

6. ЗАКЉУЧАК

Брзина преласка зависи од врсте и типа успоривача више него од ограничења брзине. На асфалтираним вертикалним успоривачима, на локацијама у зони 30 измерене су веће брзине него на гуменим вертикалним успоривачима на локацији на делу уличне мреже са општим ограничењем брзине од 50 km/h, иако су и једни и други висине 7 cm.

Закључак је да поред висине, облик, изглед и материјал вертикалног успоривача утичу на брзину преласка. Могуће је и да ширина асфалтираног вертикалног успоривача омогућава возилима бржи прелазак у односу на гумени вертикални успоривач.

Овакав закључак у складу је са брзинама измереним на платформи у булевару Деспота Стефана, на коме су измерене значајно веће брзине у односу на другу платформу. Наиме, ова платформа је око 2,5 пута шири него платформа у Крајишкој улици и изведена је на другачији начин, који је по нашем мишљењу основни узрок мање брзине преласка.

У улици Ђорђа Војновића примећен је недостатак обележених пешачких прелаза који би у комбинацији са платформом или асфалтираним вертикалним успоривачем мање висине били добро решења и адекватна алтернатива постојећим успоривачима.

На основу резултата истраживања у улици Ђорђа Војновића али на одсеку на коме је уклоњен успоривач, закључено је да не постоји објективна потреба за поновним постављањем успоривача. Наиме, мали број возила се креће већом брзином, а постоји објективна могућност да би успоривач негативно утицао на услове одвијања саобраћаја, посебно у облику у ком је постојао.

Фиксна радарска камера показала се као најбоље решење на улицама са интензивним саобраћајним током јер возачи одржавају брзину у границама од 45 km/h - 50 km/h, тако да нема непотребног успоравања на екстремно ниске брзине што утиче на појаву негативних ефеката.

7. ЛИТЕРАТУРА

- [1] <http://www.ftn.uns.ac.rs/ojs/index.php/zbornik/issue/archive>
- [2] <https://highways.today/2022/03/28/active-speed-bump->
- [3] <https://www.putisaobracaj.rs/index.php/PiS/article/download/7/1/>
- [4] <https://www.paragraf.rs/propisi/pravilnik-o-tehnickim-sredstvima-za-usporavanje-saobracaja-na-putu.html>

Кратка биографија:



Милан Милановић рођен је у Београду 1999. год. Живи у Инђији. Мастер рад на Факултету техничких наука из области Саобраћај и транспорт - Пројектовање и организација одбранио је 2024. год.

Контакт: 22milan07@gmail.com



Вук Богдановић рођен је у Сремској Митровици 1966. год. Докторирао је на Факултету техничких наука 2005. год., а од 2017. је у звању редовног професора.

Захвалница:

Резултати приказани у овом раду су део истраживања пројекта *"Савремени трендови и иновације у развоју курикулума у области саобраћаја и транспорта"*, основаног од стране *Департмана за саобраћај, Факултета техничких наука у Новом Саду, Универзитета у Новом Саду, Република Србија*.

ZNAČAJ POŠTANSKIH USLUGA U LANCIMA VREDNOSTI ELEKTRONSKE TRGOVINE U SRBIJI**IMPORTANCE OF POSTAL SERVICES IN VALUE CHAINS OF E-COMMERCE IN SERBIA**

Andela Vasić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – POŠTANSKI SAOBRAĆAJ I TELEKOMUNIKACIJE

Kratak sadržaj – U ovom radu opisan je značaj poštanskih usluga u lancima vrednosti elektronske trgovine.

Ključne reči: *Elektronska trgovina, Poštanske usluge, Rok dostave, Paketomat, Reklamacija robe*

Abstract – *In this paper, the importance of postal services in value chains of electronic commerce is described.*

Keywords: *Electronic commerce, Postal services, Delivery deadline, Parcel machine, Complaint of goods*

1. UVOD

U uslovima dinamičnog okruženja i intenzivne konkurencije, digitalizacija ne predstavlja samo osnovu za rast i razvoj, već je i faktor opstanka organizacije. Organizacije koje su osposobljene da na promene odgovore inovacijama i uvođenjem tehnologije su uspešnije. U okviru poštanskog saobraćaja i digitalne transformacije, od velikog je značaja obezbediti odgovarajućeg poštanskog operatera i osigurati dvosmernu komunikaciju s obzirom da on u velikoj meri utiče na vrednost elektronske trgovine.

Jedan od većih problema u poštanskom saobraćaju jeste nedostatak transparentnosti u pogledu praćenja pošiljaka. U tom slučaju je potrebno redovno ažuriranje podataka o statusu pošiljke i pružanju informacija o očekivanom vremenu dostave, mogućim kašnjenjima i koracima koji se preduzimaju kako bi se nastali problem rešio. Od velikog značaja jeste investiranje u tehnologijama koje mogu da obezbede praćenje pošiljaka, kao i ulaganja za optimizaciju ruta dostave. To može uključivati globalni navigacioni sistem, određene tehnologije za praćenje vozila, softvere za upravljanje zalihama i logistikom. Na taj način poštanski operateri pružaju određenu sigurnost korisnicima da će pošiljka biti uručena u ispravnom stanju i u dogovorenom vremenskom roku.

Cilj rada jeste da se ukaže na značaj poštanskih usluga u lancima vrednosti elektronske trgovine u Srbiji.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Dragana Šarac, red. prof.

2. LANAC VREDNOSTI ELEKTRONSKE TRGOVINE I POŠTANSKIH USLUGA

Elektronska trgovina predstavlja trgovinu robom, uslugama i informacija putem računarskih mreža. Ona uključuje sve oblike poslovnih transakcija kao što je distribuiranje, elektronski transfer novca, prodaja, marketing, elektronska kupovina, elektronska pošta.

2.1 Pojam i karakteristike elektronske trgovine

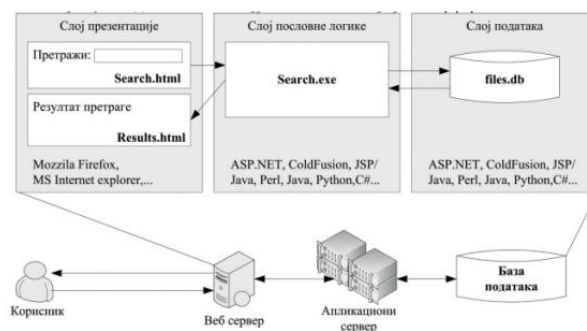
Današnji model elektronske trgovine podrazumeva kreiranje virtuelne potrošačke korpe koja je namenjena korisnicima za korišćenja elektronske trgovine, dok su se ranije koristile veb stranice kao jedinstvene prezentacije.

Tehnologije koje se koriste zasnovane su na troslojnoj arhitekturi i različitim programskim jezicima:

- Sloj prezentacije - Odnosi se na korisnički interfejs i prikazuje izlazne rezultate aplikacije.

- Sloj poslovne logike - Kontrolise funkcionalnost aplikacije i obavlja detaljnu obradu procesa. Na ovom sloju mora da postoji veb-server, čiji je zadatak da obezbedi HTTP protokol za komunikaciju s klijentom. Veb-server generiše gotove rezultate (HTML strane ili XML dokumenta), koji se prebacuju do klijenta preko HTTP protokola i kod njega se izvršavaju.

- Sloj podataka - Obuhvata servere baza podataka. Podaci su izolovani od ostalih slojeva, tako da se poboljšavaju skalabilnost i ukupne performanse sistema [1].



Slika 1. Troslojna arhitektura

2.2 Platni sistem i ponašanje kupca u elektronskoj trgovini

U okviru elektronskog plaćanja postoje tri učesnika, odnosno kupac, trgovac i banka. Kupac je osoba koja plaća elektronskim novcem, trgovac je osoba koja je plaćena elektronskim novcem, dok banka izdaje elektronski novac koji je namenjen za plaćanje.

Načini plaćanja u elektronskoj trgovini su: model digitalne trgovine, plaćanje putem digitalnog čeka, plaćanje putem platnih kartica, plaćanje po modelu naloga, plaćanje pouzecom.

Na proces donošenja odluke o elektronskoj kupovini utiču faktori okruženja, kao i individualne karakteristike. Pojedini korisnici ne preferiraju kupovinu putem interneta zbog straha nedovoljne bezbednosti i sigurnosti u kvalitet, krađe podataka na kartici.

2.3 Bezbednost elektronske trgovine i zaštita podataka u digitalnoj poštanskoj komunikaciji

Digitalna sigurnost i privatnost podataka su od visokog značaja u bilo kojem preduzeću, nezavisno od delatnosti kojim se ono bavi. U cilju čuvanja podataka, Ustav Republike Srbije (RS) nalaže tajnost pisama i ostalih sredstava komunikacije. Odstupanja su moguća onda kada sud donese takvu odluku, u slučaju vođenja krivičnog postupka ili zaštite bezbednosti RS, na način koji je definisan zakonom. Dalje, uređivanje odnosa u oblasti poštanskih usluga u skladu sa Zakonom o poštanskim uslugama temelji se na načelu nepovredivosti poštanskih pošiljaka i tajnosti podataka. Svaki poštanski operator ima zaduženje da kada vrši poštanske usluge postupa u skladu sa svojim specifičnim uslovima, obezbedi nepovredivost poštanskih pošiljaka i tajnost podataka [3].

Onda kada se ne postupa u skladu sa posebnim uslovima, ili se ne obezbedi nepovredivost poštanskih pošiljaka i tajnost podataka, javni, ili drugi, poštanski operator, ili drugo pravno lice, će snositi konsekvence, odnosno biti kažnjeno za prekršaj novčanom kaznom [2].

2.4 Prevencija sajber napada na poštanske sisteme

Sajber bezbednost štiti koji ima zadatak da odgovori na različite vrste pretnji koje mogu doći iz virtuelnog sveta, poput malicioznih softvera, fišinga, neovlašćenih pristupa i krađa podataka. Konkretno kako bi se poštanski sistemi zaštitili od sajber napada neophodno je da koriste šifrovanje, zaštitne zidove (engl. firewall), antivirusne programe, softverske sisteme, autentifikaciju, autorizaciju korisnika, sigurnosne politike i procedure koje se primenjuju na organizacijskom nivou [2].

2.5 Uloga poštanskih usluga u lancima vrednosti elektronske trgovine

Elektronske poštanske usluge omogućavaju komunikaciju putem informaciono-komunikacionih tehnologija i u okviru njih spadaju sledeće usluge: elektronsko poštansko sanduče, javni pristup internetu u prostorijama pošte, elektronski žig, elektronska poštanska marka, elektronski potpis, elektronski telegram, elektronska razglednica, onlajn direktna pošta, registrovana elektronska pošta i hibridna pošta, kao i mnoge druge.

2.6 Razvoj elektronskih usluga i potreba za brзом i efikasnom dostavom u JP „Pošta Srbije“

Transformacija PTT sistema, u skladu sa savremenim tendencijama dovela do organizacionog i ekonomskog osamostaljenja telekomunikacija. Tada je pošta morala da komercijalizuje svoje poslovanje kako bi se obezbedili preduslovi za transformaciju i uspeh na tržištu.

Pošta Srbije pruža veliki broj elektronskih usluga kao što su:

1. Sistem za elektronsko praćenje poštanskih pošiljaka (Track and Trace) - Omogućava korisniku poštanskih usluga da u svakom trenutku sazna gde mu se pošiljka nalazi, odnosno njen status (uručena, vraćena, preusmerena).

2. Traženje lokacija poštanskih jedinica – Ova usluga daje mogućnost korisnicima da dobiju podatke o poštanskim jedinicama na osnovu poštanskog broja ili naziva pošte.

3. Traženje poštanskog adresnog koda (PAK-a) – Omogućava korisnicima da unosom naziva grada, opštine ili naselja, naziva ulice i kućnog broja saznaju podatke o PAK-u.

4. Usluge geografskog informacionog sistema Pošte - Omogućava korišćenje GIS tehnologija, upotrebu poštinih prostornih podataka i kvalitetniji prikaz područja na kome se pružaju usluge (<http://www.postagis.rs/>).

5. Usluge Hibridne pošte - Usluge koje su kombinacija tradicionalne poštanske usluge i elektronske pošte, kroz tri objedinjena tehnološka procesa.

3. DELFI METODA

Kako bi se sagledala ključna pitanja u vezi sa značajnim karakteristikama poštanskih usluga u lancima vrednosti elektronske trgovine, upotrebljena je Delfi metoda.

Formirana je grupa eksperata za poštanske usluge u elektronskoj trgovini koji uključuju stručnjake iz oblasti logistike, elektronske trgovine, tehnologije i marketinga. Na bazi upitnika, sagledan je značaj pojedinih karakteristika usluge, kao što su brzina, pouzdanost, određište, alternativni načini dostave itd. Na bazi ovih rezultata, sačinjen je upitnik koji je prosleđen korisnicima elektronske trgovine.

4. METODA ANKETE KORISNIKA

Upitnik za korisnike elektronske trgovine sačinjen je na bazi mišljenja 10 eksperata. Upitnik je sadržao 30 pitanja i prosleđen je korisnicima elektronske trgovine i poštanskih usluga.

U anketi je učestvovalo 291 ispitanika.

5. REZULTATI

U okviru ovog istraživanja postavljene su tri hipoteze.

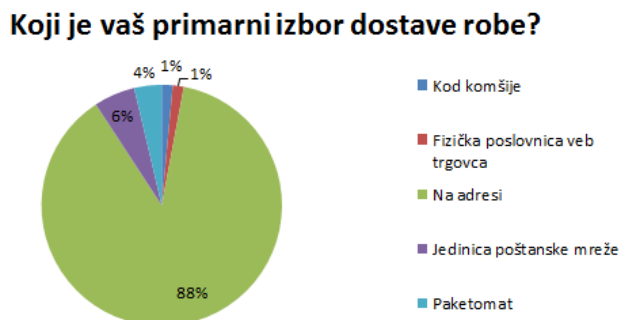
Prva hipoteza (H1)- „Ako su dostave brže i pouzdanije, tada će korisnici biti skloniji korišćenju usluga dostava, što će rezultirati povećanje ukupne potražnje za dostavom robe“. H1 je potvrđena jer većini (121) ispitanika je prihvatljivo da pošiljka bude isporučena u roku od 4 dana.

Brža dostava povećava zadovoljstvo korisnika što rezultira povećanom potražnjom za dostavom robe.



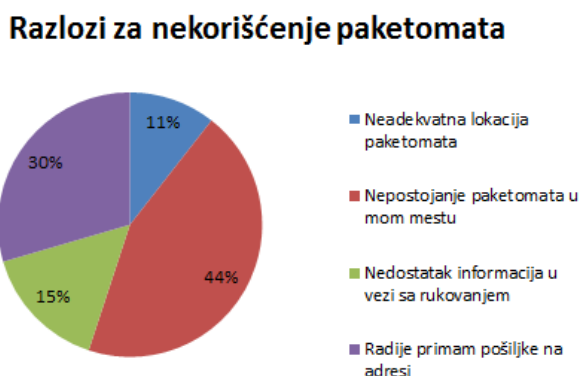
Slika 1. *Koja su vaša očekivanja u pogledu rokova za dostavu robe?*

Druga hipoteza (H2) - „Ako se alternativni način dostave pošiljaka na kućnu adresu pokaže kao efikasan i praktičan, korisnici će imati veću sklonost korišćenju ovog načina dostave zbog komfora i ograničenost vremena“ je potvrđena jer 88% ispitanika biraju ovakav vid dostave zbog sigurnosti da će pošiljka biti uručena lično njima.



Slika 2. *Koji je vaš primarni izbor dostave robe?*

Treća hipoteza (H3) - „Dostava robe putem paketomata predstavlja pouzdan i praktičan način preuzimanje robe i pruža korisnicima veću fleksibilnost u pogledu preuzimanja pošiljaka“ je odbačena zato što većina ispitanika nemaju paketomat u svom mestu ili se nalazi daleko od njihove lokacije. Smatraju da postoji velika mogućnost da se dogodi greška prilikom preuzimanja pošiljaka, kao i nedostatak informacija u vezi sa rukovanjem paketomata.



Slika 3. *Razlozi za nekorišćenje paketomata*

6. ZAKLJUČAK

Na osnovu izvršenog istraživanja koje se odnosi na ponašanje korisnika u elektronskoj trgovini, došli smo do zaključka da su korisnici uglavnom mlađa populacija i da je neophodno obezbediti pružanje podrške i adekvatnu obuku za korišćenje tehnologija starijim osobama. Svaka elektronska prodavnica bi trebala da ima definisanu tabelu veličine obuče i odeće sa svim odgovarajućim podacima kako bi se svaki kupac bio zadovoljan svojom robom i samim tim bi se smanjila potreba za vraćanjem robe i izbegli nepotrebni troškovi povratne poštarine.

Takođe, od velikog značaja je i izbor poštanskog operatora kako bi se izvršila brza i pouzdana dostava, i kako bi korisnici bili zadovoljni. Poštanski operatori sa širokom mrežom distributivnih centara i dostavnih ruta mogu pokriti veću teritoriju i dostaviti pakete čak i na udaljene ili teško dostupne lokacije, što je od izuzetnog značaja za onlajn prodavnice koje žele da obuhvate širi geografski krug kupaca.

Većina ispitanika nema priliku da koristi paketomat jer ne postoji u njihovom mestu ili u njihovoj blizini, pa je od velikog značaja postaviti paketomate na žižnim tačkama kao što su tržni centri, autobuske i železničke stanice, fakulteti.. Oni pružaju niz pogodnosti, kako za radnike pošte tako i za korisnike, kao što su smanjenje redova čekanja u pošti, ušteda vremena korisnika, dostupnost. Svako preduzeće treba da ima mogućnost izbora pakovanja od recikliranog materijala koji pored doprinosa zaštiti životne sredine i smanjenju ekološkog afekta, može uticati na pozitivne efekte brenda, troškove i percepciju korisnika.

Svakom od nas koji smo u ulozi korisnika ili klijenta bilo kog oblika poslovanja, digitalna sigurnost i privatnost podataka je na prvom mestu. Korisnici su i dalje skeptični kada je u pitanju vršenje onlajn plaćanja, ili jednostavno ostavljanje ličnih podataka na određenim sajtovima pri registraciji.

7. LITERATURA

- [1] <https://www.elab.fon.bg.ac.rs/wp-content/uploads/2015/07/Elektronsko-poslovanje.pdf> (posećeno dana 07.05.2024)
- [2] Tomić-Petrović, N., Petrović, Lj., Zaštita privatnosti u pružanju poštanskih i drugih usluga, 34. Simpozijum o novim tehnologijama u poštanskom i telekomunikacionom saobraćaju - PosTel 2016, Beograd <https://postel.sf.bg.ac.rs/simpozijumi/POSTEL2016/RAD/OVI%20PDF/Menadzment%20procesa%20u%20postanskom%20i%20telekomunikacionom%20saobraćaju/6.%20Tomic-Petrovic-Petrovic.pdf> (posećeno dana 07.05.2024)
- [3] A1, Uvod u sajber bezbednost, USAID od američkog naroda, dostupno: <https://www.preduzmi.rs/wp-content/uploads/2023/05/Uvod-u-sajber-bezbednost-https://www.posta.rs/index.aspx11052023.pdf> (posećeno dana 07.05.2024)

Kratka biografija:



Andela Vasić rođena je u Vranju 1999.godine.

Zvanje diplomirani inženjer saobraćaja stekla je 2022. godine na Fakultetu tehničkih nauka i iste upisala master akademske studije na smeru Poštanski saobraćaj i telekomunikacije.

Master rad na temu " Značaj poštanskih usluga u lancima vrednosti elektronske trgovine u Srbiji" odbranila je 2024. godine.

Kontakt: andjelavasic909@gmail.com

1326

fleš tipa, dok je memorija za podatke na bazi brzih SRAM memorijskih ćelija. Veličina programske memorije iznosi 32 kilobajta, od kojih treba oduzeti 512 bajtova namenjenih smeštanju tzv. bootloader programa koji omogućava programiranje mikrokontrolera na računaru. Memorija za podatke sadrži 32 osmobarbitna registra opšte namene, 64 U/I registra, 160 registara za proširenu U/I memoriju i dva kilobajta namenjena smeštanju podataka potrebnih za izvršavanje programa iz programske memorije. Na čipu se nalazi i EEPROM (Electrically Erasable Programmable Read-Only Memory) veličine jednog kilobajta koji može da zadrži podatke koji ostaju i nakon isključivanja struje [1].

3.2 SIM900 GSM GPRS štit

GPRS/GSM (General Packet Radio Service/Global System for Mobile communication) štit omogućava primanje podataka sa udaljenih lokacija korišćenjem usluga GSM mobilnih mreža (slika 2).



2. Prikaz GPRS/GSM štita

GPRS štit se konfiguriše i kontroliše preko njegove UART(Universal Asynchronous Receiver Transmitter) komunikacije koristeći jednostavne AT (ATtention) komande. Zasnovan je na SIM900 modulu kompanije SIMCOM. Pored komunikacionih karakteristika, GPRS štit ima 12 GPIO (General Purpose Input Output), 2 PWM (Pulse Width Modulation) i jedan ADC (Analog Digital Converter) [2].

3.3 Senzor temperature i vlažnosti vazduha DHT22

DHT-22 (AM2302) senzor temperature i vlažnosti vazduha je kapacitivni senzor koji je namenjen za praćenje i očitavanje temperature i relativne vlažnosti vazduha. Sastoji se od kapacitivnog senzora vlažnosti vazduha, termistora za merenje temperature i elektronike za komunikaciju sa mikrokontrolerom. Fabrički je kalibrisan, pa ne zahteva dodatne komponente. Dolazi u plastičnom kućištu (koje ujedno služi i kao mehanička zaštita), a pri vrhu se nalazi i otvor za montažu. Automatski se radi kompenzacija vlažnosti u zavisnosti od temperature vazduha. Senzor može da meri temperaturu do 80°C, a vlažnost vazduha u opsegu od 0-100 % [3].

3.4 Dvokanalni AD konvertor – HX711

Dvokanalni AD (Analog Digital) konvertor sa 24 – bitnim konvertorom visoke preciznosti je specijalno namenjen za digitalne vage. Povezivanjem HX711 sa jedne strane na

senzor težine, a sa druge strane na mikrokontroler, dobija se mogućnost očitavanja precizne težine, kontrole nekog procesa ili mogućnost detekcije prisustva uz relativno malo podešavanje kalibracionih parametara. HX711 sadrži dva analogna ulazna kanala i ima integrisani programabilni pojačavač nivoa ulaznog signala sa predefinisanim pojačanjima od 32/64/128 (namenjen za merenje izuzetno niskih vrednosti signala) [4].

3.5 Merna ćelija CZL - 642

Merna ćelija pretvara silu u električni signal koji se može meriti. Električni signal se menja proporcionalno primenjenoj sili. Konkretno za potrebe ovog projekta je korišćena merna ćelija za naprezanje CZL – 642 (slika 3) nosivosti do 200 kg i preciznosti od 0,02 kg.



Slika 3. Merna ćelija CZL – 642

Merne ćelije za merenje naprezanja se sastoje od metalne šipke sa pričvršćenim merilima naprezanja. Merač naprezanja je električni senzor (merna traka) koji meri silu ili naprezanje na objektu. Otpor merača naprezanja varira kada se na objekat primeni spoljna sila, što rezultira deformacijom oblika objekta (u ovom slučaju metalne šipke). Otpor merača naprezanja je proporcionalan primenjenom opterećenju, što omogućava izračunavanje težine predmeta [5].

3.6 DC – DC konverter XHM404

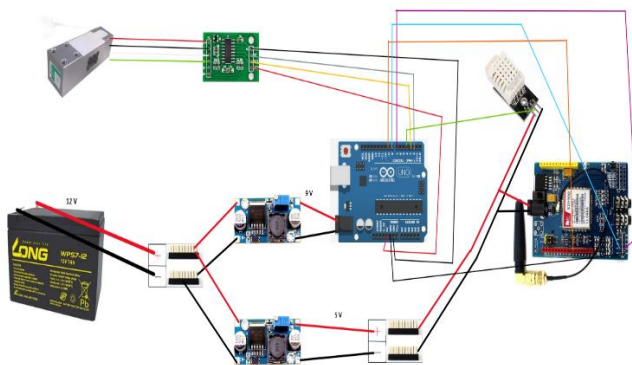
XHM404 DC-DC step-down regulator napona (ispravljač) je uređaj koji stabilise i spušta DC napon preko fino podesivog potencijometra, s time da ulazni napon mora biti veći od izlaznog. Frekvencija rada mu iznosi 180 KHz, što omogućava stabilan i brz rad. Ulazni napon je 4-40 V, izlazni napon: 1,25-36 V, izlazna struja: nominalna struja 8 A [6].

3.7 Baterija WPS7 – 12

Bateriju WPS7 – 12 proizvodi tajvanska firma Kung Long Batteries Industries. Ona pripada grupi olovnih VRLA (Valve Regulated Lead Acid) baterija. VRLA baterija (olovna baterija sa regulacijom ventila) je vrsta akumulatorske punjive baterije. Zbog svoje konstrukcije mogu se montirati u bilo kojoj orijentaciji i ne zahtevaju stalno održavanje. Ova vrsta punjive baterije ne zahteva formiranje, jer ih je uradio već proizvođač pri prvom punjenju baterije neposredno pri proizvodnji iste. Iako se pun kapacitet olovne VRLA baterije dostiže posle 50 – 100 ciklusa, nikako nije preporučljivo da se baterija podvrgava bespotrebnom punjenju/pražnjenju, jer je radni vek baterije usko povezan sa brojem punjenja/pražnjenja. Olovne VRLA baterije po strukturi su gotovo iste kao i baterije u automobilu koje nije potrebno formirati [7].

4. POVEZIVANJE KOMPONENTI SISTEMA ZA PRAĆENJE TEŽINE KOŠNICE

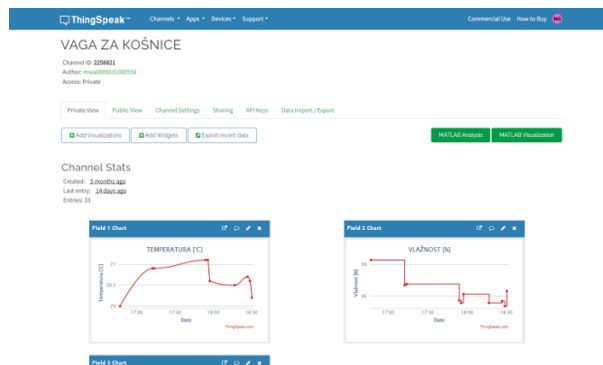
Napajanje sistema pruža baterija WPS7 – 12. Nju je potrebno povezati sa DC – DC konverterom kako bih se spustio napon od 12 V na 9 V, sa kojim se napaja centralna upravljačka jedinica Arduino Uno. Potom je potrebno povezati AD konvertor sa Arduinoom sa jedne strane i mernom ćelijom sa druge. Tako povezan sistem ima sposobnost merenja težine. Da bi se proširile mogućnosti sistema, potrebno je na njega dodati još senzor temperature i vlažnosti vazduha, kao i GSM GPRS štit. Senzor temperature i vlažnosti vazduha (DHT22) povezuje se na napajanje od 5 V, koje se dobija uz pomoć drugog DC-DC konvertera, i na Arduino Uno. GSM GPRS štit se takođe povezuje na isto napajanje od 5 V sa jedne strane, dok se sa druge strane povezuje sa Arduinoom. Tako povezan sistem ima sposobnosti merenja težine, temperature i vlažnosti vazduha, kao i sposobnost uspostavljanja internet konekcije. Šema povezivanja sistema je prikazana na slici 4.



Slika 4. Šema povezivanja sistema

5. KONFIGURACIJA CLOUD SERVERA

Podatke koje prikupi vaga potrebno je poslati na neki udaljeni server (cloud), gde će se oni skladištiti i biti dostupni u svakom trenutku. Za cloud server je odabran besplatan ThingSpeak server koji pruža usluge skladištenja podataka i prikazivanja u vidu dijagrama. Potrebno je pristupiti sajtu ThingSpeak-a (<https://thingspeak.com/>) i registrovati novi korisnički nalog.



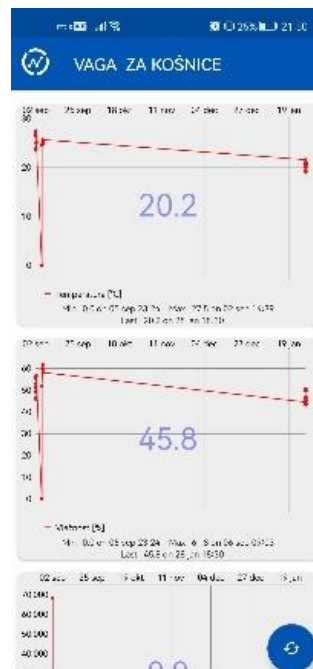
Slika 5. Prikaz izgleda stranice na serveru ThingSpeak-a

U slučaju ovog projekta kanal je imenovan kao VAGA ZA KOŠNICE. Pristupanjem samom kanalu dobija se različiti niz opcija za konfiguraciju samog kanala, kao što se može videti na slici 5. U sekciji Private View dat je

prikaz osnovnih podataka o kanalu. Tu se nalaze svi dijagrami u koje smeštamo određene grupacije podataka. U ovom slučaju prikazani su dijagram za temperature, vlažnost i ispod njih se nalazi dijagram težine. Pored tih podataka mogu se videti i informacije o vremenu kada je sam kanal napravljen, kada je poslednji put ažuriran, kao i koliko je poslato poruka na sam kanal. Svaki od prikazanih dijagrama se može konfigurisati prema željama korisnika.

6. APLIKACIJA NA MOBILNOM TELEFONU

Pored načina koji zahteva pristup računaru, da bi se logovali na sajt od ThingSpeaka, postoji i druga (brža) opcija za uvid u podatke koje šalje vaga. To je aplikacija na mobilnom telefonu koja je uvezana sa samim cloud serverom preko internet konekcije. Za realizaciju ovog projekta je korišćen Android mobilni telefon, pa je samim time za instalaciju neke aplikacije potrebno pristupiti aplikaciji Google Play. Pomoću nje se preuzima željena aplikacija pod imenom ThingView–ThingSpeak viewer. Izgled same aplikacije dat je na slici 6.



Slika 6. Izgled otvorene aplikacije ThingSpeak viewer na mobilnom telefonu

7. REALIZACIJA KONAČNOG REŠENJA

Konačno rešenje sistema vage za košnice je realizovano kao na slici 4. Pre samog puštanja sistema u rad, dok je još u test fazi potrebno je podesiti određene parametre u samom kodu da bi sistem radio kao što se to od njega očekuje. Prvi parametar koji je potrebno podesiti je parametar za kalibraciju same vage, da bi vaga mogla da precizno meri težinu. To se radi tako što se uzme 5 predmeta poznate težine i za svaki od njih je očitana sirova vrednost sa ADC - a. Treba birati predmete koji su teški od nekoliko stotina grama, pa sve do nekoliko desetina kilograma, kako bih se dobila što tačnija očitavanja na svim težinama. Dobijena očitavanja sa ADC-a je zatim potrebno podeliti sa težinom od koje su dobijene da bi se dobio kalibracioni parametar za tu težinu. Konačno, od dobijenih 5 vrednosti parametara je

potrebno naći srednju vrednost i ubaciti je u kod. Još jedan parametar koji je potrebno podesiti je broj telefona od samog vlasnika vage. Broj telefona se dodaje zbog mogućnosti vage da uputi poziv vlasniku u slučaju krađe same vage. Ako dođe do nagle promene težine vage će pozvati vlasnika (sa periodom trajanja poziva od 20 sekundi), kako bi onda on mogao pravovremeno da reaguje na nastalu situaciju. Nakon podešavanja parametara, sledi spuštanje koda na Arduino Uno. To se radi tako što se ploča poveže sa laptopom preko USB kablova i spusti sam kod. Nakon spuštanja koda potrebno je povezati Arduino Uno sa ostatkom sistema i pustiti sistem u rad. Izgled konačnog sistema nakon spajanja svih komponenti prikazan je na slici 7.



Slika 7. Izgled konačnog sistema

Sva elektronika samog sistema je stavljena u plastičnu kutiju koja treba da je zaštiti od uslova iz spoljašnje sredine. Sistem se pokreće preko prekidača koji omogućava napajanje čitavog sistema. Još jedan vid zaštite od spoljašnjih uticaja, kao i u svrhe kamuflaže, predstavlja drveno postolje (slika 8) koje se postavlja na samu vagu i na koje potom ide košnica.



Slika 8. Izgled drvenog postolja stavljenog na samu vagu

8. ZAKLJUČAK

Kroz ovaj rad uspešno je izvršeno upoznavanje sa svim komponentama sistema za praćenje težine košnice sa

distribuirane lokacije, načinom njihovog povezivanja i funkcionisanja. Pomenuti su neki osnovni pojmovi pčelarstva, alati koji se koriste u tom poslu, kao i mala istorija ovog starog zanata. Dat je prikaz samog sistema za praćenje težine košnice sa distribuirane lokacije, način povezivanja njenih komponenti i neki osnovni parametri i informacije o sistemu. Svaka komponenta sistema za koju je smatrano da treba da bude pomenuta prilikom opisa sistema je predstavljena time što su date neke njene osnovne karakteristike, uloga u samom sistemu i sam prikaz komponente.

Prednosti ovakvog sistema su što širok krug ljudi, a pre svega sam pčelar, može da ima konstantan uvid u trenutno stanje određenih parametara (temperature, vlažnosti, težine) na samom pčelinjaku, bez odlaska na isti. Pored mogućnosti pružanja podataka o određenim parametrima, u sam sistem je implementiran jedan vid zaštite protiv krađe. Ukoliko dođe do nagle promene težine na samoj vagi, vaga će istog momenta uputiti poziv vlasniku, koji onda može da reaguje na adekvatan način.

Sistem za praćenje težine košnice sa distribuirane lokacije bi se mogao unaprediti dodavanjem još pojedinih funkcionalnosti. Jedna od tih funkcionalnosti bi mogla biti GPS (Global Positioning System) lokacija pčelarske vage. Pomoću nje bi vlasnik vage mogao da ima uvid u trenutnu lokaciju same vage, pa u slučaju krađe bi mogao tačno da prati kretanje vage i da dođe do samog počinioca. Takođe, mogla bi se dodati i funkcionalnost praćenja ispražnjenosti baterije. Tada bi pčelar imao konstantan uvid u nivo napunjenosti baterije, pa bi mogao da reaguje usled niskog nivoa napona baterije zamenom ili dopunjavanjem baterije.

9. LITERATURA

- [1] <https://www.sk.rs/arhiva/clanak/13678/arduino-uno>
- [2] <https://www.tinyosshop.com/datasheet/GSM%20Shield%20Datasheet.pdf>
- [3] https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf
- [4] http://www.kelm.ftn.uns.ac.rs/literatura/mut/LV09_Procitati_pre_vezbe.pdf
- [5] <https://elektromodul.rs/senzor-vlaznosti-dht22-am2302/>
- [6] <https://www.techtonics.in/xh-m404-dc-4-40v-8a-voltage-regulator-module-digital-pwm-adjustable-dc-dc-step-down-voltage-regulator-dc-xl4016e1>
- [7] <https://www.akumulator-shop.rs/baterije-za-ups/baterija-long-wps7-12-12v-7ah>

Kratka biografija:

Nebojša Gradinski rođen je u Somboru 1999. god. Master rad na Fakultetu tehničkih nauka iz oblasti Mehatronika, robotika i automatizacija odbranio je 2024.god.
kontakt: ngradinski26@gmail.com

BIOMETRIJSKE TEHNOLOGIJE U NAPREDNIM SISTEMIMA ZA POMOĆ VOZAČIMA**BIOMETRIC TECHNOLOGIES IN ADVANCED DRIVER ASSISTANCE SYSTEMS**

Adrijana Delić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MEHATRONIKA

Kratak sadržaj – U ovom radu prikazana je primena uređaja koji mogu da olakšaju vozačima vožnju posebno u situacijama pospanosti. S obzirom da je vožnja postala sastavni deo svakodnevnog života mnogih ljudi, zahteva njihovu punu pažnju. Prikazan je primer korišćenja „Stay awake“ uređaja, kao i kamere i algoritma koji prate budnost vozača. Na kraju rada opisan je „Eye tracker“ uređaj koji bi takođe mogao da se iskoristi kao uređaj za praćenje pogleda vozača tj njegove pažnje tokom vožnje.

Ključne reči: budnost vozača, „Stay awake“, „Eye tracker“, uređaj za praćenje pogleda vozača

Abstract – This paper shows the application of devices that can make it easier for drivers to drive, especially in situations of drowsiness. Since driving has become an integral part of many people's daily lives, it requires their full attention. An example of using the "Stay awake" device, as well as a camera and an algorithm that monitor the driver's vigilance, is presented. At the end of the paper, the "Eye tracker" device is described, which could also be used as a device for monitoring the driver's gaze, i.e. his attention while driving.

Keywords: driver's vigilance, "Stay awake", "Eye tracker", a device for monitoring the driver's view

1. UVOD

Prema studijama koje su sprovedene 2008. godine oko 20% svih saobraćajnih nesreća uzrokovano je skretanjem pogleda i umorom vozača. Ovaj procenat je zabrinjavajući pa podstiče detaljno proučavanje ovog problema i realizaciju i primenu „Eye tracker“, „Stay awake“ i drugih uređaja. Najnoviji izveštaji o globalnom stanju bezbednosti na putevima ukazuje na to da saobraćajne nezgode godišnje odnesu živote preko 1 350 00 ljudi. Na svake 24 sekunde jedna osoba izgubi život u saobraćajnim nezgodama na putevima. Mnogi vozači koji su zaspali za volanom u toku vožnje, priznaju da to nisu ni osetili.

2. „STAY AWAKE“ UREĐAJ

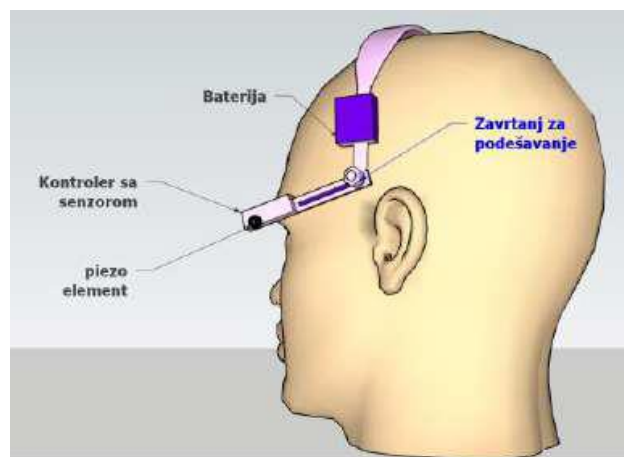
Kao što samo ime uređaja govori „ostani budan“ je uređaj koji će na vreme upozoriti vozača da je umoran i da mu je hitno potreban odmor. Način rada ovog uređaja je da prati očne kapke vozača i primeti i delimičnu zatvorenost oka.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Platon Sovilj, red. prof.

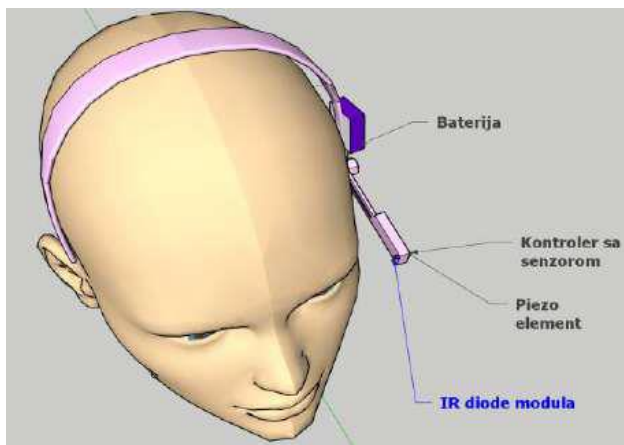
„Stay awake“ uređaj je instaliran na glavi vozača po principu slušalica. Težina uređaja je 78 g, tako da je lagan i ne predstavlja problem kod dugotrajnog nošenja. Senzor je podešen da stoji na samoj ivici oka i svojom pozicijom ne ograničava vidno polje vozača, čak ni kod perifernog vida. Napaja se baterijom od 9 V i koristi se kontroler Arduino NANO Atmega 328P. Ovo je kontroler vrlo malih dimenzija i pogodan je za primenu u ovakvim uslovima gde smo ograničeni gabaritom i težinom uređaja. Senzorski modul prilagodljiv je ambijentalnom svetlu. Ima infracrvene diode koje emituju i primaju svetlost, a ta svetlost nije u vidnom rasponu čoveka, tako da ne ometa vozača. Domet detekcije senzora se može menjati podešavanjem potencijometra od 2 do 30 cm. Pored malih dimenzija i preciznosti u radu, ovaj uređaj je daleko konkurentniji i sa cenom od ostalih uređaja. Njegova cena je manja od 30e.

Kao što je prikazano na slici 2.1 uređaj se može podesiti u odnosu na dimenziju glave korisnika. Ukoliko je potrebno uređaj se može zakačiti i na naočare korisnika.

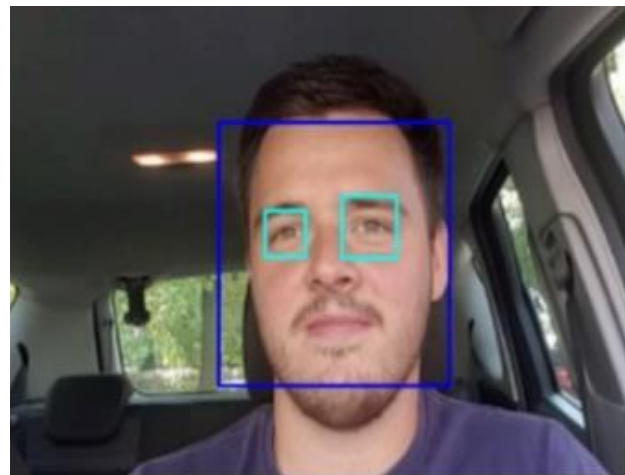


Slika 2.1. „Stay Awake“ sa posebnim nosačem

Uključivanjem uređaja piezo element se oglašava pištanjem u periodu od 1500 ms i time potvrđuje da je uređaj spreman za rad. Pored podešavanja dimenzija nikakva druga podešavanja nisu potrebna. Dioda moraju biti usmerene ka kapku oka. Testiranje uređaja se obavlja tako što se uređaj podesi, a oko je otvoreno. Zaklapanjem kapka u periodu od 1000 ms, piezo element se oglašava pištanjem i tako stavlja do znanja da je uređaj ispravan.



Slika 2.2. Usmeravanje diode



Slika 3.1. Uspešna detekcija lica i očiju

Ukoliko u toku vožnje, vozač zaklopi očne kapke u periodu od 1000 milisekundi piezo element će se oglasiti „pištanjem“ i na taj način upozoriti vozača da treba da stane i odmori se. Prema istraživanjima trajanje treptaja je između 100 i 400 milisekundi, pa je iz tog razloga uzeto 1000 ms kao granica, jer se nakon 1000 ms vodi da je vozač zapao u mikrosnivanje. Zbog individualnosti vozača morala je biti veća granica od 400-500 ms.

Stay Awake uređaj se pokazao kao veoma pouzdan uređaj i u dnevnim i u noćnim uslovima, kao i u obe varijante nošenja. Cena ovog uređaja je veoma pritupačna što dobro utiče na sam status uređaja na tržištu. Pored osnovnog rešenja postoji i dodatak vibro motora na uređaj, kako bi se vozač dodatno razbudio. Vibro motor podiže cenu uređaja, pa je ostavljen izbor kupcu da li želi uređaj sa ili bez vibro motora.

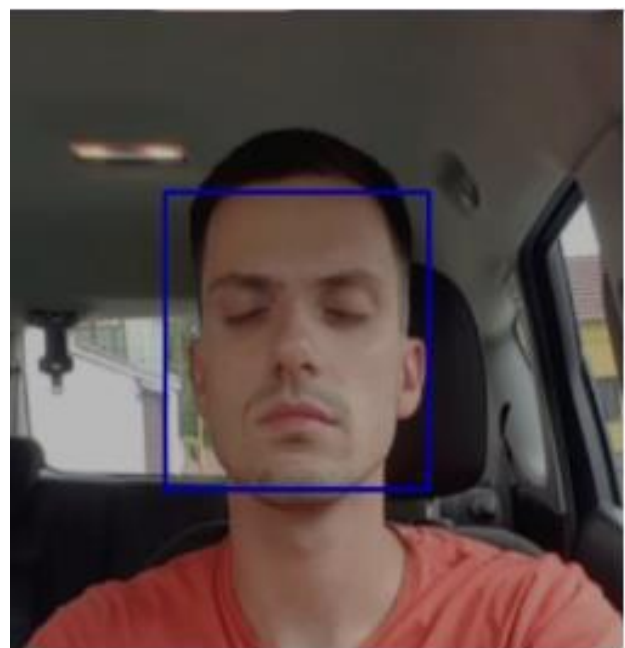
3. PRAĆENJE STANJA VOZAČA POMOĆU KAMERE U VOZILU

Pri izvođenju ovog projekta korišćene su metode za prepoznavanje objekta koji su razvili *Viola i Jones*. Nakon detektovanja objekta, u ovom slučaju lica i očiju, proverava se da li vozač žmuri, gleda u stranu i spušta glavu. *Viola Jones* algoritam zasniva se na mašinskom učenju. Konkretno algoritam za ovaj projekat koristi se Haarovim karakteristikama, integralnom slikom, *AdaBoost*-om i kaskadnim klasifikatorom.

Algoritam se sastoji iz dva dela. Jedan se odnosi na prednju stranu lica, a drugi na profil lica, jer karakteristike profila i prednjeg dela lica nisu iste.

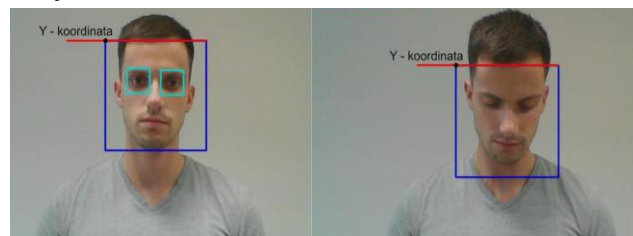
Nakon prvog kadra u kojem se pojavljuje lice, poziva se funkcija koja je zadužena za detekciju lica. Veličina glave se izračunava na osnovu visine i širine pravougaonika koji je označio lice. Taj okvir se kasnije koristi za proveru da li je glava spuštena. Nakon otkrivanja lica, algoritam pokušava da detektuje oči unutar okvira, kao što je prikazano na slici 3.1.

Svaki put kada algoritam detektuje lice, a oči ne, poziva se funkcija koja proverava da li su oči korisnika zatvorene. U algoritmu postoji brojač koji aktivira alarm da su vozaču oči predugo zatvorene tj nedetektovane.



Slika 3.2. Uspešna detekcija lica i očiju kada vozač žmuri

Nakon detektovanja lica i očiju proverava se spuštenost glave. Proverava se y-koordinata okvira detektovanog lica. Poredi se početna y-koordinata gornjeg levog ugla sa trenutnom y-koordinatom gornjeg levog ugla pravougaonika. Kako vozač spušta glavu tako se i y-koordinata menja.



Slika 3.3. Prikaz detekcije prilikom spuštanja glave

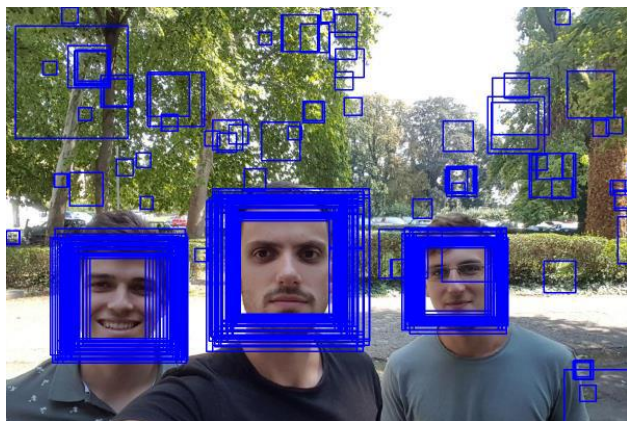
S obzirom da postoji mogućnost da se vozač udalji od ili da se približi kameri tokom vožnje, postoji mogućnost promene y-koordinate. Rešenje ovog problema sprovedeno je računanjem površine označenog okvira.

Algoritam funkcioniše tako da, ukoliko ne može da se otkrije prednji deo lica, pokušava da otkrije profil lica. Ovaj deo funkcioniše tako da ako se detektuje profil lica, uključuje se brojač i meri se koliko dugo vozač skreće pogled. Postoji određeni prag nakon kog se aktivira alarm upozorenja koji obaveštava vozača da vrati pogled napred.

Detektor je određen samo za jednu stranu, pa je upotrebljen efekat ogledala, kako bi mogla da se detektuje i leva i desna strana profila vozača.

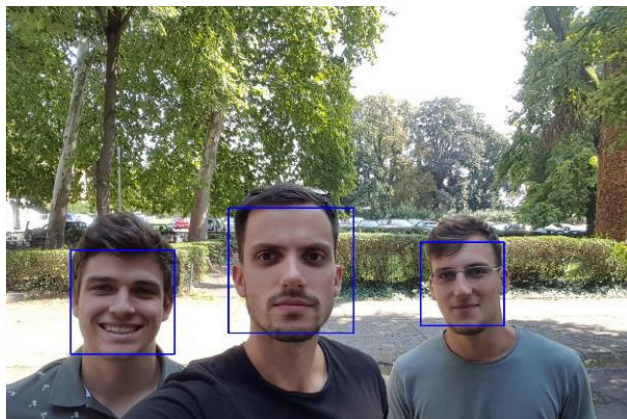
Ovaj algoritam testiran je u automobilu sa kamerom mobilnog telefona. Testovi su se obavili preko dana, jer način rada algoritma nije predviđen za noćne uslove. Nakon testiranja utvrđeno je da je na tačnost rezultata najviše uticala funkcija koja vrši detekciju.

Pored toga jako bitan faktor bio je nivo svetlosti. Svetlost je uticala na samu detekciju. Postoje određeni parametri funkcije koji su posebno uticali na detekciju, poput parametra minimalni broj suseda (eng. *minNeighbors*). Ovaj parametar radi po principu kliznog pod-prozora. Primenom ovog kliznog pod-prozora preko slike, traži se željeni objekat i prolazi kroz sliku. Proces se ponavlja sve dok se veličina prozora više ne može promeniti. Nakon ovog procesa, gde svaka iteracija menja veličinu prozora i ponovo prolazi kroz sliku, detektuje se mnogo lažno-pozitivnih lica, kao što je prikazano na slici 3.4.



Slika 3.4. Detekcija sa parametrom minimalni broj suseda=0

Da bi se lažno-pozitivna lica uklonila primenjuje se kocept minimalni broj suseda. Ako je otkriven objekat u blizini drugih prozora onda je u redu. Ovaj broj određuje koliko je suseda potrebno za detekciju lica.

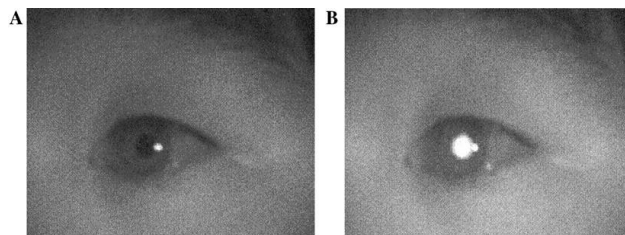


Slika 3.5. Detekcija sa parametrom minimalni broj suseda=3

Broj suseda je dosta veći kada je u pitanju prednja strana lica u odnosu na profil lica. Ovaj pristup daje tačnije rezultate, ali slabiju detekciju u situacijama kada osvetljenje nije idelano. Detekcija stanja zavisi od detekcije oka što zavisi od toga da li je lice otkriveno ili ne. Detektor za traženje očiju obučen je da obeleži samo otvoreno oko. Tokom testiranja uočena je greška da se ponekad detektuju oči i kada su zatvorene. Iz tog razloga je minimalni broj suseda povećan na veću vrednost.

4. EYE TRACKER UREĐAJ

Tehnike praćenja pogleda na osnovu kamere oslanjaju se na neka svojstva odnosno karakteristike oka koje može otkriti i pratiti kameru ili druge opričke ili fotosenzibilne uređaje. Limbus i zenica se koriste za praćenje. Limbus je granica oka gde se iris (obojeni deo oka) i beonjača spajaju. Zenicu je teže otkriti zbog kontrasta između granice zenice i irisa, ali tehnika praćenja zenice imaju bolju tačnost jer nisu prekrivene kopcima (osim za vreme treptanja). Da bi se poboljšao kontrast koriste se infracrveni izvori svetlosti. Pošto infracrvena svetlost nije u vidljivom spektru čoveka ne odvlači pažnju korisnika. Pošto kamera sada „vidi“ svetlo koje je reflektovano sa zadnje strane oka, slično efektu crvenih očiju pri fotografiji pomoću jakog blica, fotoaparat vidi svetlu zenicu kao što se može videti na slici 4.1.



Slika 4.1. Prikaz tamne (a) i svetle (b) zenice

5. ZAKLJUČAK

Kao neko ko svakodnevno provodi dva sata vozeći automobil na posao, mogu da potvrdim koliko umor, pospanost i skretanje pažnje može loše uticati na kvalitet same vožnje. Uređaji koji su prikazani u ovom projektu, zaista mogu da pomognu kako bi statistika saobraćajnih nesreća uzrokovana skretanjem pogleda vozača, bila smanjena. *Stay awake* uređaj pokazuje veliku tačnost i zaista ga je lako prilagoditi samom korisniku. Njegova težina i cena mogu da doprinesu njegovom statusu na tržištu. Što se tiče postavljanja kamere na vetrobran auta da prati pogled vozača, rezultati su nešto lošiji, jer na samu kameru utiču svetlost i udaljenost vozača od kamere.

Ovaj uređaj bi mogao dobro da se iskoristi, ali da se izvrše određene promene i poboljšanja koja bi doprinela tačnosti usled promene svetlosti, promene položaja glave, udaljenosti i sl.

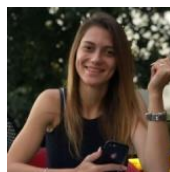
Na samom kraju spominje se *Eye tracker* uređaj kao predlog za praćenje same zenice oka. *Eye tracker* uređaj je zaista koristan u mnogim sferama. Ja sam ga koristila u oblasti sporta, gde se pokazao veoma korisno, pri mnogo

bržim pokretima čitavog tela, pa samim tim i glave i oka, u odnosu na pokrete tokom vožnje. Nisu izvršena testiranja, ali ne sumnjam da bi *Eye tracker* uređaj značajno doprineo poboljšanju pažnje vozača.

6. LITERATURA

- [1] World Health Organization, Global status report on road safety, 2009, dostupno na: http://www.who.int/violence_injury_prevention/road_safety_status/2009/en/
- [2] The National Sleep Foundation, Sleep in America™ Poll, 2009, dostupno na: <https://sleepfoundation.org/sites/default/files/2009%20POLL%20HIGHLIGHTS.pdf>
- [3] Cypress, Advanced Driver Assistance Systems (ADAS), dostupno na: <http://www.cypress.com/solutions/advanced-driver-assistance-systems-adas>
- [4] Grand View Research, Advanced Driver Assistance Systems (ADAS) Market Size, Share & Trend Analysis Report By Solution (Adaptive Cruise Control, Blind Spot Detection), By Component, By Vehicle, And Segment Forecasts, 2018 – 2025, dostupno na: <https://www.grandviewresearch.com/press-release/global-advanced-driver-assistance-systems-adas-market>
- [5] Matthew Walker, professor of neuroscience and psychology and the director of the Center for Human Sleep Science at the University of California (2017). "Why We Sleep", dostupno na: <https://www.simonandschuster.com/books/Why-We-Sleep/Matthew-Walker/9781501144325>
- [6] W. Vanlaar, H. M. Simpson, D. Mayhew, and R. Robertson, „Fatigued and drowsy driving: attitudes, concerns and practices of Ontario drivers“, Traffic Injury Research Foundation, 2007, dostupno na: <http://tirf.ca/publications/fatigued-drowsy-driving-attitudes-concerns-practices-ontario-drivers/>
- [7] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features”, In Conference on Computer Vision and Pattern Recognition, 2001, dostupno na: www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdfhttps://docs.opencv.org/3.4.1/dc/d88/tutorial_train_cascade.html
- [8] NVIDIA, ADVANCED DRIVER ASSISTANCE SYSTEMS (ADAS), dostupno na: <https://www.nvidia.com/en-us/self-driving-cars/adas/>
- [9] BlackBerry QNX, Advanced driver assistance systems, dostupno na: <http://blackberry.qnx.com/en/products/adas/index#benefits>
- [10] Automotive Machine Vision ALPHA reference board, dostupno na: http://www.rt-rk.com/download/rt-rk_ALPHA_ADAS_board.pdf
- [11] R. Kothari, J. Mitchell, Detection of eye locations in unconstrained visual images, in: Proc. of the International Conference on Image Processing, dostupno na: www.pixelmaven.com/jason/articles/Kothari96_DetectionOfEyeLocationsInUnconstrainedVisualImages.pdf
- [12] R. Jacob: What you look at is what you get, dostupno na: <https://ubicomp.net/wp/wp-content/uploads/2017/10/p62-jacob.pdf>
- [13] Y. Ebisawa, S. Satoh, Effectiveness of pupil area detection technique using two light sources and image difference method, in: A. Szeto, R. Rangayan (Eds.), Proc. of the 15th Annual Internat. Conf. of the IEEE Eng. in Medicine and Biology Society, dostupno na: www.witpress.com/Secure/elibrary/papers/VID95/VID95009FU.pdf

Kratka biografija:



Adrijana Delić rođena je 1996. god u Vrbasu. Master rad na Fakultetu tehničkih nauka iz oblasti Mehatronika-Robotika i automatizacija, odbranila je 2024.god.
kontakt: delic.adrijana@gmail.com

SISTEM PAMETNE KUĆE

SMART HOME SYSTEM

Srđan Bijelić, Vladimir Rajs, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MEHATRONIKA

Kratak sadržaj – Ovaj projekat predstavlja jedan od mogućih načina izrade sistema pametne kuće gdje je poseban naglasak stavljen na jednostavnost, pouzdanost, fleksibilnost, modularnost i niske troškove izrade i održavanja. Sistem je zasnovan na centralnim kontrolerima Arduino nano i WeMos D1 mini mrežnim kontrolerima (WiFi). Interakcija korisnika sa sistemom vrši se putem Android ili iOS aplikacije, kao i direktnim putem pomoću tastature, tastera, rotirajućeg enkodera i displeja.

Ključne reči: Mikrokontroler, senzor, aktuator, WiFi, internet stvari, server, Blynk.

Abstract – This project represents one of the possible ways to create a smart home system, with a particular emphasis on simplicity, reliability, flexibility, modularity, and low construction and maintenance costs. The system is based on central controllers such as Arduino Nano and WeMos D1 Mini network controllers (WiFi). User interaction with the system is carried out through an Android or iOS application, as well as directly using a keyboard, buttons, a rotary encoder, and a display.

Keywords: Microcontroller, sensor, actuator, WiFi, Internet of Things, server, Blynk.

1. UVOD

Zadatak sistema pametne kuće jeste da korisnika liši zamornih, monotonih i repetitivnih poslova i odluka, te da mu boravak u stambenom objektu bude prijatniji, konformniji i bezbjedniji. Zbog kompleksnosti se ovakvi sistemi dijele na više modula, koji se bave različitim procesima i, u manjoj ili većoj mjeri, samostalno djeluju na njega. Moguća je i potpuna autonomija sistema kada su neki segmenti djelovanja u pitanju, ali preduslov za to je implementacija „machine learning“ modela. Ono što karakteriše ovakav projekat svakako je IoT (eng. *Internet of Things* – internet stvari) tehnologija, koja obezbjeđuje efikasno povezivanje digitalnog i fizičkog svijeta, tj. povezivanje senzora i aktuatora iz stvarnog svijeta sa internetom [1]. Sa jedne strane imamo kontrolere sa sensorima i aktuatorima (module) povezane na WiFi ruter u svojoj lokalnoj mreži, a sa druge imamo korisničke uređaje kao što su telefon, tablet, računar, smart TV... (Slika 1). Komunikacija funkcioniše u oba smjera, tako da korisnik na svojim uređajima očitava podatke koje sistem šalje, ali istovremeno može i da upravlja sistemom.

NAPOMENA:

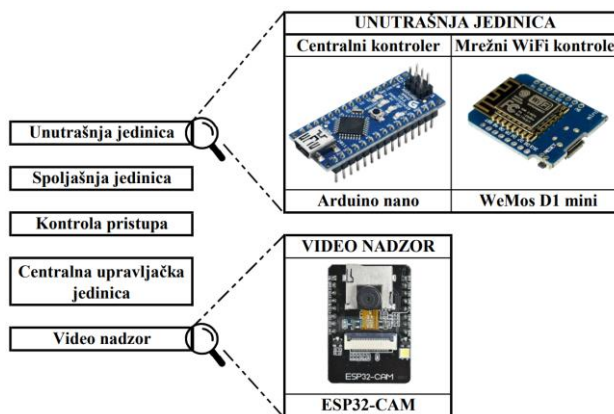
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Vladimir Rajs, vanr. prof.



Slika 1. Način funkcionisanja IoT tehnologije

2. ANALIZA PROBLEMA

Sistem je zbog svoje kompleksnosti, a na osnovu karakteristika i funkcionalnosti, podijeljen na pet uređaja, odnosno modula. Svaki od njih pomoću ugrađenog mrežnog WiFi modula komunicira sa web serverom. Kako je jedan od početnih uslova projekta bio jednostavnost i niska cijena, urađena je detaljna analiza dostupnih kontrolera, njihovih karakteristika, podrške, zajednice i cijene, pa je tako svaki modul, osim video nadzora, zasnovan na Arduino nano centralnom kontroleru, koji radi u sprezi sa mrežnim kontrolerom WeMos D1 mini (slika 2). Osnova modula za video nadzor je kontroler ESP32-CAM.

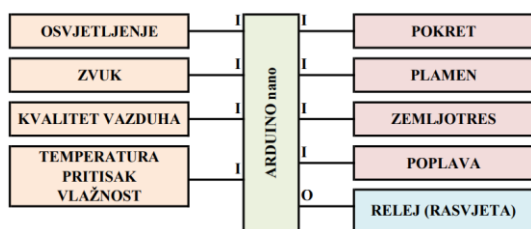


Slika 2. Moduli sistema i kontroleri na kojima su zasnovani

Svaki od navedenih modula obavlja njemu svojstvene funkcionalnosti, a one su u direktnoj zavisnosti od lokacije na kojoj se ovi moduli ugrađuju. Opremljeni su različitim krajnjim uređajima - sensorima, aktuatorima i perifernim uređajima. Kako se cilj ovog projekta ogleda u predstavljanju mogućnosti jednog ovakvog sistema, u realnim okolnostima bi se koristili neki kvalitetniji krajnji uređaji, boljih karakteristika i performansi u vidu bolje osjetljivosti, dometa, otpornosti na smetnje i slično. U slučaju ovakvih korekcija, program unutar kontrolera se ne bi značajno mijenjao, što je izrazito važno kada je riječ o modularnosti, kao jednom od početnih uslova projekta.

2.1. Unutrašnja jedinica

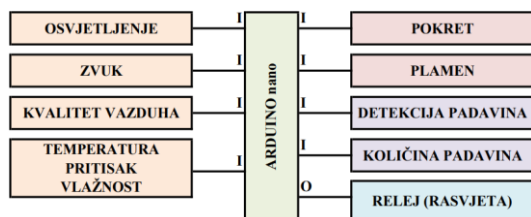
Osnovna funkcija ovog modula je da prikuplja ambijentalne podatke iz prostorije u kojoj se nalazi i da ih šalje web serveru, odnosno Android aplikaciji za nadzor ili dalju obradu i upravljanje sistemom. Opremljen je digitalnim i analognim senzorima (slika 3), a jedini izlazni signal namjenjen je za upravljanje rasvjetnim aktuatorom, odnosno LED trakom putem jednokanalnog releja. Predviđeno je da se modul montira na plafon prostorije čije parametre želimo da nadziramo, a razlog se ogleda u tome što će se u gornjem nivou prostorije najprije detektovati prisustvo štetnih gasova, i iz tog ugla će se najlakše uočiti plamen, kretanje ili zemljotres. Upravo te informacije uz detekciju poplave kreiraju alarmne statuse ovog modula.



Slika 3. Senzori i aktuatori unutrašnje jedinice

2.2. Spoljašnja jedinica

Riječ je o modulu koji se ni konstruktivno ni po funkcionalnosti ne razlikuje od unutrašnje jedinice. Jedina razlika se ogleda u podacima koje prikuplja i u lokaciji na koju se montira, a to je sa spoljašnje strane objekta - na trijem, balkon i slično. Predstavlja uprošteni vid meteorološke stanice, sa zadatkom da prikuplja podatke vezane za spoljašnje vremenske uslove i da ih šalje web serveru. Kao i unutrašnja jedinica, ima definisane alarmne statuse, a to su detekcija pokreta i plamena, dok se informativna obavještenja na Android aplikaciji kreiraju detekcijom padavina. Sastoji se od senzora koji predstavljaju ulazne podatke (slika 4), a jedini izlazni signal ima istu funkciju kao i u prethodno opisanom modulu – upravljanje jednokanalnim relejom, odnosno rasvjetnim aktuatorom.



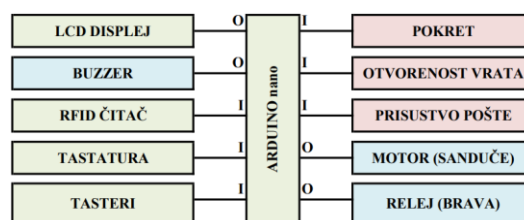
Slika 4. Senzori i aktuatori spoljašnje jedinice

Ideja o automatizaciji sistema ogleda se u poređenju određenih parametara ulazne i izlazne jedinice, te bi sistem na osnovu njih donosio odluku o daljim koracima i o načinu upravljanja određenim aktuatorima. Samo neke od mogućnosti su:

- Otvaranje/zatvaranje prozora u zavisnosti od nivoa buke i temperature unutar i van objekta,
- Uključivanje/isključivanje ventilacije u zavisnosti od kvaliteta vazduha unutar i van objekta,
- Podizanje/spuštanje roletni u zavisnosti od nivoa osvjetljenja unutar i van objekta (dan/noć) i slično.

2.3. Kontrola pristupa

Predstavlja dio sistema koji upravlja ulaznim vratima, dozvoljavajući pristup objektu samo onim licima koji imaju odgovarajuće kredencijale, lozinku, ID karticu ili tag, pa se tako ovaj modul sastoji od spoljašnjeg i unutrašnjeg dijela. Ostavljena je mogućnost da korisnik putem aplikacije upravlja elektronskom bravom ulaznih vrata. Zbog karakteristične lokacije pridružen mu je sistem za nadzor i upravljanje poštanskim sandučetom (slika 5). Spoljašnji dio ovog modula obuhvata senzore i periferne uređaje za interakciju sa korisnikom, kao i aktuatora za upravljanje ulaznim vratima i poštanskim sandučetom. Unutrašnji dio se sastoji od senzora pokreta i tastera za otključavanje vrata osobama koje žele da izađu iz objekta.

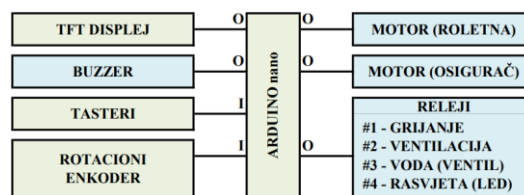


Slika 5. Senzori, aktuatori i periferni uređaji modula kontrole pristupa

Ostavljena je mogućnost da se sistem „naoruža“, odnosno da se postavi u stanje pripravnosti - kada senzori pokreta kreiraju alarmna stanja.

2.4. Centralna upravljačka jedinica

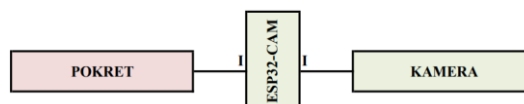
Jedini zadatak ovog modula je da upravlja kućanskim aparatima, odnosno aktuatorima, i to preko višekanalnih relejnih modula ili putem PWM signala (servo motori). Zamisao je da ovaj modul bude u sprezi sa drugim modulima iz sistema i da pomoću aktuatora automatski mijenja ambijentalne uslove na osnovu informacija kojima raspolaže. Njegovi periferni uređaji (slika 6) obezbjeđuju korisniku direktnu interakciju sa sistemom, dok je i upravljanje putem Android aplikacije ostavljeno kao opcija.



Slika 6. Senzori, aktuatori i periferni uređaji centralne upravljačke jedinice

2.5. Kamera

Funkciju video nadzora obavlja jednostavni ESP32-CAM, modul sa ugrađenom kamerom, u sprezi sa senzorom pokreta (slika 7). Isti može da se implementira u kombinaciji sa tasterom za zvono i slično, a ti dodatni elementi imaju ulogu „okidača“ za slikanje.



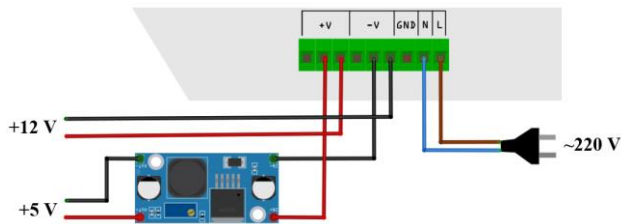
Slika 7. Sastavni dijelovi modula kamere

3. HARDVER

Radi lakše analize i planiranja obavljena je kategorizacija uređaja od kojih se sastoji sistem. Uređaji su grupisani u napajanja, kontrolere, senzore, aktuatora i periferne uređaje. Za njihovo povezivanje je bilo neophodno dizajnirati i izraditi prilagodne provodne pločice tehnikom toner transfera. Kućišta su izrađena tehnikom 3D štampe.

3.1. Napajanje

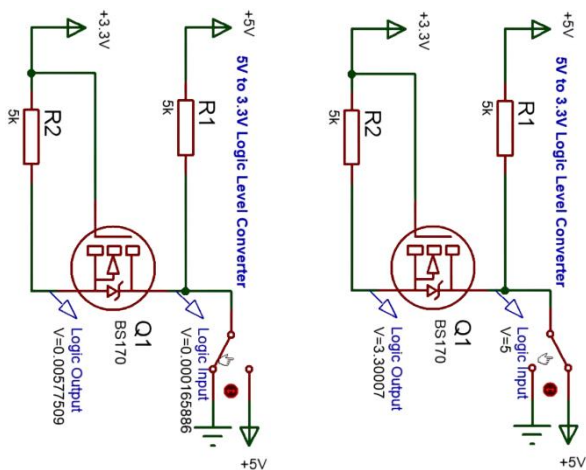
Korišteni su prekidački ispravljači izlaznog napona +12 V u kombinaciji sa step-down modulima, a razlog se ogleda u njihovoj jednostavnosti, malim dimenzijama, energetske efikasnosti i niskoj cijeni. Ulaz step-down konvertora se povezuje na izlaz ispravljača (slika 8) i podešavanjem ugrađenog potencijometra na njegovom izlazu se dobija stabilnih +5 V, koje dalje sistem koristi za napajanje senzora, kontrolera i pojedinih aktuatora.



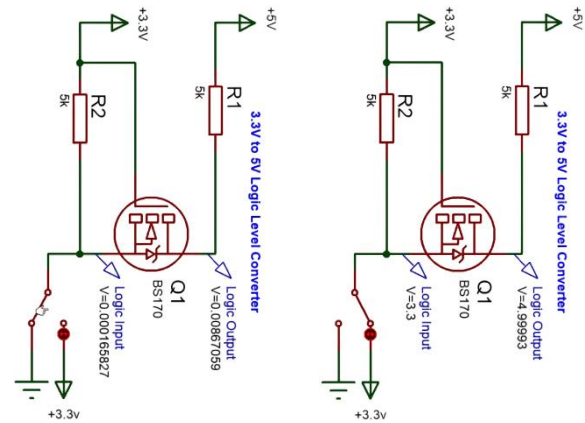
Slika 8. Povezivanje ispravljača i step-down modula

3.2. Kontroleri

Centralni kontroleri primaju i obrađuju podatke sa senzora i prosleđuju ih mrežnom kontroleru korištenjem komunikacionog UART protokola. Mrežni kontroler je povezan na lokalni WiFi ruter, te na taj način web serveru dostavlja programom definisane pakete podataka. Web server na zahtjev korisnika šalje te pakete na njegov WiFi ruter, i korisnik ih prima na svom uređaju putem Android, iOS ili web aplikacije. Sa druge strane korisnik može sistemu da pošalje određene zahtjeve za upravljanjem aktuatorima, i ta komunikacija se odvija u suprotnom smjeru u odnosu na opisanu. Korišteni centralni i mrežni kontroleri rade na različitim naponskim (logičkim) nivoima, tako da je za neometanu obostranu razmjenu podataka bilo neophodno uvesti konvertore logičkih naponskih nivoa. Na tržištu postoje već formirani bidirekcionni višekanalni konvertori, ali je njihov način funkcionisanja najlakše shvatiti kroz rad (jednokanalnog) tranzistora MOSFET BS170 (slika 9 i 10) [2].

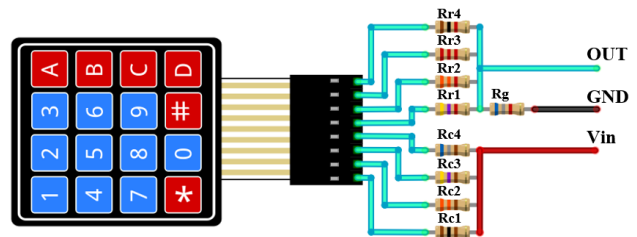


Slika 9. Konverzija višeg na niži logički naponski nivo



Slika 10. Konverzija nižeg na viši logički naponski nivo

Najveći problem sa resursima (ulazno-izlaznim izvodima) centralnog kontrolera primjetan je kod modula kontrole pristupa. Iz tog razloga je za LCD 1602 displej korišten prilagodni I2C modul, čime je smanjen broj potrebnih ulaznih izvoda sa sedam na dva (SCL i SDA). Drugi vid optimizacije ogleda se u implementaciji metode analognog multipleksiranja (slika 11) na tastaturi, gdje je potrebnih osam ulaznih izvoda svedeno na jedan analogni.



Slika 11. Metoda analognog multipleksiranja

Primjer proračuna izlaznog signala za taster „8“ vidljiv je u izrazu (1) [3]:

$$V_{out8} = \frac{V_{in} \cdot R_g}{R_{c3} + R_{r2} + R_g} \quad (1),$$

nakon čega se proračunava vrijednost signala koji će kontroler očitati na svom analognom ulazu (2):

$$AnalogRead8 = \frac{V_{out8}}{V_{in} \cdot 1024} \quad (2).$$

4. SOFTVER

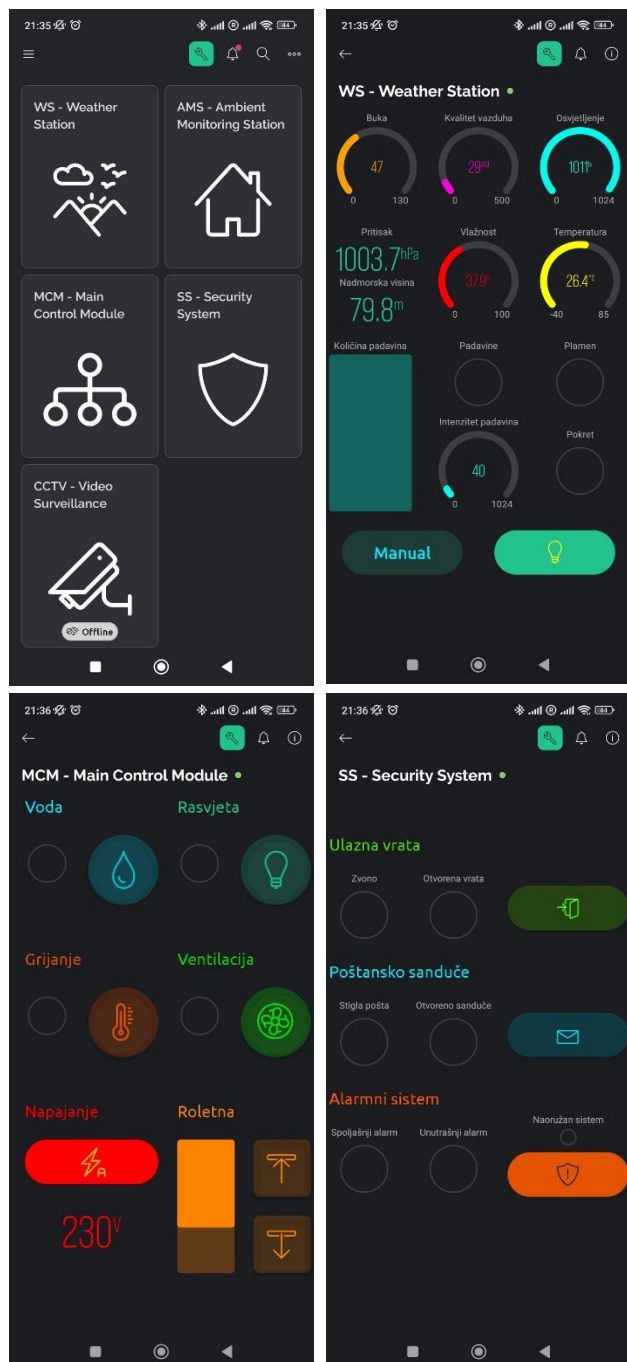
Softversko rješenje je izrazito značajno za postizanje precizne i efikasne kontrole nad hardverom, i ono predstavlja ključni skup instrukcija i algoritama koji omogućavaju mikrokontroleru da obavlja željene funkcije i implementira logiku odlučivanja. Za razvoj softvera kontrolera korišteno je Arduino IDE okruženje, dok se za kreiranje aplikacije rješenje pronašlo u Blynk platformi.

4.1. Arduino IDE

Korištenje Arduino IDE okruženja značajno pojednostavljuje programiranje, naročito za početnike, pružajući korisnicima širok spektar funkcija i biblioteka koje su prilagođene za Arduino platformu. Na raspolaganju je i izrazito velika i susretljiva otvorena Arduino zajednica sa mnogobrojnim forumima, koji značajno mogu da utiču na brzinu rješavanja problema tokom programiranja. Softver je organizovan u modularne blokove, radi lakšeg održavanja i proširivosti koda.

4.2. Blynk

Blynk predstavlja inovativnu platformu za brzu izradu Internet of Things (IoT) projekata i stvaranje povezanih uređaja bez potrebe za velikim znanjem o programiranju ili kompleksnim hardverskim konfiguracijama. Riječ je o alatu koji omogućava korisnicima da jednostavno povežu svoje uređaje sa internetom i pruža im se mogućnost za praćenje, upravljanje i automatizaciju uređaja putem mobilne aplikacije (Android, iOS ili web). Projekti se povezuju sa Blynk Cloud-om obezbjeđujući tako daljinsku kontrolu i praćenje parametara sistema iz bilo kog dijela svijeta. Za svaki od ranije opisanih modula kreirana je zasebna aplikacija (slika 12).



Slika 12. Primjer izgleda Android aplikacije: gore lijevo – prva strana (izbornik), gore desno – spoljašnja jedinica, dole lijevo – centralna upravljačka jedinica, dole desno – kontrola pristupa

5. ZAKLJUČAK

U samom uvodu naglašeno je da je sistem zamišljen kao modularan i proširiv, tako da je konačan završetak ovakvog projekta jako teško definisati. Faza koja je upravo završena može se smatrati zadovoljavajućom jer su obuhvaćene i implementirane sve funkcionalnosti i uslovi koji su inicijalno postavljeni, dok je sistem prilagođen i postavljen tako da korisniku bude blizak, intuitivan i jednostavan za korištenje. Apsolutno je jasno da sistem kao ovakav još uvijek ne može da predstavlja kompletan Smart Home System, iz razloga što je tek potrebno implementirati automatizaciju, odnosno obezbijediti autonomiju sistemu u zavisnosti od potreba korisnika. Tek nakon toga sistem može da dobije istinski epitet „pametne kuće“. Jedan od važnijih idućih koraka jeste povezivanje sistema na WAN mrežu, kako bi istom moglo da se pristupi sa bilo koje udaljenosti, s obzirom na to da je korisnik trenutno ograničen samo na lokalnu mrežu (LAN).

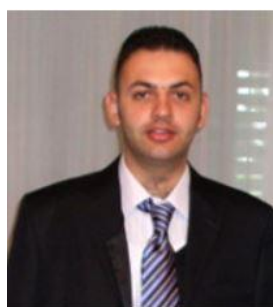
6. LITERATURA

- [1] Dragan Antić, Šta je Internet of Things, dostupno na: <https://samoobrazovanje.rs/sta-je-internet-of-things-internet-inteligentnih-uredjaja/>, [22. mart 2019.]
- [2] Sourav Gupta, Bi-Directional Logic Level converter using MOSFET, dostupno na: <https://circuitdigest.com/tutorial/bi-directional-logic-level-controller-using-mosfet>, [14. avgust 2019.]
- [3] CrossRoads, 4x4 Keypad connected with 1 wire, dostupno na: <https://forum.arduino.cc/t/4x4-keypad-connected-with-1-wire/195931>, [novembar 2013.]

Kratka biografija:



Bijelić Srđan rođen je u Prijedoru (BiH) 1988. godine. Diplomirao je na Fakultetu tehničkih nauka u Novom Sadu iz oblasti mehatronike 2013. godine. Od 2014. godine je radio kao inženjer za telekomunikacione optičke mreže, a od 2022. godine radi kao projekt menadžer za projekte slabe struje. Kontakt: srdjan.d.bijelic@gmail.com



Vladimir Rajs rođen je 1982. godine u Apatinu. Diplomirao je 2007. a doktorirao 2015. godine na Fakultetu tehničkih nauka u Novom Sadu. Od 2016. godine zaposlen je kao docent, a od 2020. kao vanredni profesor na Departmanu za energetiku, elektroniku i telekomunikacije FTN-a. Oblasti interesovanja su mu elektronika i primijenjena elektronika.

U realizaciji Zbornika radova Fakulteta tehničkih nauka u toku 2023. godine učestvovali su sledeći recenzenti:

Aco Antić	Dunja Vrbaški	Marinko Maslarić	Nemanja Tasić
Aleksandar	Đorđe Vukelić	Marko Marković	Nenad Simeunović
Anđelković	Goran Jeftenić	Marko Todorov	Nikola Vojnović
Aleksandar Kovačević	Goran Savić	Marko Vekić	Platon Sovilj
Aleksandar	Goran Sladić	Maša Bukurov	Predrag Teodorović
Kupusinac	Goran Stojanović	Mijodrag Milošević	Radivoje Dinulović
Aleksandar Selakov	Goran Tepić	Milan Delić	Radomir Kojić
Aleksandra Radulović	Gordan Stojić	Milan Gavrić	Romana Bošković-
Aleksandra Pešterac	Gordana Ostojić	Milan Marinković	Živanović
Ana Nešić Tomašević	Gordana	Milan Mirković	Sandra Dedijer
Andraš Anderla	Milosavljević	Milan Rapajić	Saša Medić
Andrija Rašeta	Igor Dejanović	Milan Rackov	Slavica Mitrović
Atila Zelić	Igor Maraš	Milan Segedinac	Senka Bajić
Bojan Batinić	Igor Peško	Milan Trivunić	Slađana Milićević
Bojan Matić	Ilija Bašićević	Milan Vidaković	Slobodan Morača
Bojan Tepavčević	Iva Šiđanin	Milena Krklješ	Slobodan Šupić
Bojan Jovanović	Ivan Mezei	Milica Vračarić	Slobodan Tabaković
Boris Agarski	Ivan Prokić	Milica Miličić	Srđan Popov
Boris Stojić	Ivana Mihajlović	Milica Kisić	Srđan Vukmirović
Branko Milosavljević	Ivana Vasiljević	Miloš Simić	Stevan Gostojić
Damir Đaković	Ivana Katić	Miloš Šešlija	Stevan Milisavljević
Danijela Ćirić	Ivana Maraš	Milovan Lazarević	Stevan Stankovski
Danijela Gračanin	Ivana Miškeljin	Milja Simeunović	Suzana Draganić
Danijela Lalić	Jelena Atanacković	Miljana Prica	Svetlana Bačkalić
Darko Čapko	Jeličić	Miodrag Milutinov	Svetlana Nikoličić
Darko Stefanović	Jelena Borocki	Miodrag Žigić	Tamara Škorić
Dejan Ecet	Jelena Ivetić	Miroslav Dramićanin	Teodora Vučković
Dejan Reljić	Jelena Radonić	Miroslav Zarić	Vesna Stojaković
Dejan Movrin	Jelena Slivka	Mirko Raković	Višnja Žugić
Dejan Ubavin	Jelena Spajić	Miro Govedarica	Vladimir Đaković
Dejana Nedučin	Kalman Babković	Miroslav Kljajić	Vladimir Mučenski
Dragan Adamović	Lazar Kovačević	Miroslav Zarić	Vojin Ilić
Dragan Dinu	Lidija Krstanović	Mladen Tomić	Vuk Bogdanović
Dragan Ivanović	Ljiljana Popović	Mladen Radišić	Vuk Vranjkovic
Dragan Ivetić	Ljubica Duđak	Nataša Milosavljević	Zoran Brujić
Dragan Jovanović	Ljubo Nedović	Nebojša Brkljač	Zoran Čepić
Dragan Pejić	Ljubomir Budinski	Nebojša Radović	Zoran Jeličić
Dragan Ružić	Magdolna Pal	Nebojša Ralević	Željen Trpovski
Dragana	Maja Turk Sekulić	Neda Milić Keresteš	Željko Kanović
Konstantinović	Maja Petrović	Nemanja Kašiković	
Dragoljub Šević	Marija Silađi	Nemanja Sremčev	
Drago Žarković			