



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



ЗБОРНИК РАДОВА ФАКУЛТЕТА ТЕХНИЧКИХ НАУКА

Едиција: Техничке науке - зборници

Година: XXXVIII

Број: 3/2023

Нови Сад

Едиција: „Техничке науке – Зборници“

Година: XXXVIII

Свеска: 3

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 Балош
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 трећа 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 24.10.2022. до 04.11.2022. год., а који се промовишу 22.03.2023. 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 три свеске.

У овој свесци, са редним бројем 3. објављени су радови из области:

- машинства,
- електротехнике и рачунарства,
- анимација у инжењерству и
- чистих енергетских технологија.

У свесци са редним бројем 4. објављени су радови из области:

- грађевинарства,
- саобраћаја,
- графичког инжењерства и дизајна,
- архитектуре,
- сценске архитектуре и дизајна и
- биомедицинског инжењерства.

У свесци са редним бројем 5. објављени су радови из области:

- инжењерског менаџмента,
- инжењерства заштите на раду и заштите животне средине,
- математике у техници,
- инжењерства третмана и заштите вода (TEMPUS),
- управљања ризиком од катастрофалних догађаја и пожара и
- инжењерства информационих система.

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. Nikola Андријашевић, Сања Бојић, УНАПРЕЂЕЊЕ ПРОИЗВОДНОГ СИСТЕМА ПРИМЕНОМ LEAN МЕТОДОЛОГИЈЕ	311-314
2. Ilija Damjanac, PODELA SISTEMA ZA KLIMATIZACIJU PROJEKAT MAŠINSKIH INSTALACIJA – VRF SISTEM ..	315-318
3. Perica Radovanović, UPOREDNA ANALIZA SISTEMA GREJANJA STAMBENIH ZGRADA U SRBIJI I ZEMLJAMA EVROPSKE UNIJE	319-322

Radovi iz oblasti: Elektrotehnika i računarstvo

1. Nevena Pantić, Bane Popadić, IZBOR POGONSKOG PODSISTEMA ELEKTRIČNOG VOZILA SA NAGLASKOM NA MAŠINE SPECIJALNE KONSTRUKCIJE	323-326
2. Aleksandra Višnjić, Karolina Kasaš-Lažetić, NUMERIČKO MODELOVANJE RASPODELE ELEKTRIČNOG POLJA I POTENCIJALA U IZOLACIONOM SLOJU VN KABLOVA U PRISUSTVU PUKOTINA	327-330
3. Tamara Milovac, Dejan Jerkan, PRORAČUN FLUKSEVA SINHRONIH MAŠINA SA STALNIM MAGNETIMA	331-334
4. Marko Mijatović, PREDIKCIJA OCENE APLIKACIJE NA OSNOVU KOMENTARA	335-338
5. Aleksandra Milijević, ANOTACIJA I DETEKCIJA NEGACIJE U KOMENTARIMA PROIZVODA	339-342
6. Petar Nikolić, UTVRĐIVANJE RASA PASA I MAČAKA UPOTREBOM KONVOLUCIONIH NEURONSKIH MREŽA	343-346
7. Stefan Pajkanović, Marko Vekić, ANALIZA POREMEĆAJA KVALITETA ELEKTRIČNE ENERGIJE KROZ SIMULACIJU I MERENJA .	347-350
8. Дејан Јовић, Марко Векић, УПРАВЉАЊЕ УСПОСТАВЉАЈУЋИМ ПРЕТВАРАЧИМА ЕНЕРГЕТСКЕ ЕЛЕКТРОНИКЕ У МИКРОМРЕЖИ	351-354
9. Dragiša Simeunović, OSObine I UPOTREBLJIVOST Gin Gonic RADNOG OKVIRA	355-357

	STRANA
10. Stanko Jevtić, IMPLEMENTACIJA PERFORMANTNOG SISTEMA POMOĆU MIKROSERVISA	358-361
11. Minja Lepar, MODELOVANJE TEMA U TEKSTU NA OSNOVU NASLOVA DOKUMENATA	362-365
12. Александар Комазец, Предраг Теодоровић, РАЗВИЈАЊЕ МУЛТИПЛАТФОРМСКЕ ВИДЕО ИГРЕ СА ПОДРШКОМ ЗА ИГРАЊЕ ПРЕКО МРЕЖЕ	366-369
13. Катарина Нинковић, СОФТВЕРСКО РЕШЕЊЕ ИНТЕРНЕТ ПРОДАВНИЦЕ	370-373
14. Milena Vujičić, RAZVOJ WEB APLIKACIJE „Sistem za prihvatanje rezultata ispita i statistika predmeta“	374-377
15. Igor Đeviki, PROJEKTOVANJE I RAZVOJ PULSNOG OKSIMETRA	378-381
16. Тамара Говорчин, ИНТЕГРАЦИЈА РЕЛАЦИОНЕ И ГРАФОВСКЕ БАЗЕ ПОДАТАКА ЗА ПОДРШКУ РАДА ИНФОРМАЦИОНОГ СИСТЕМА ДРУШТВЕНЕ МРЕЖЕ	382-386
17. Dušan Milunović, KATEGORIZACIJA I ANALIZA SENTIMENTA DOKUMENATA UPOTREBOM VIŠEJEZIČNOG TRANSFORMER MODELA SA VIŠE IZLAZA	387-390
18. Јелена Достих, КОМПАРАТИВНА АНАЛИЗА ПРИМЕНЕ ТРАДИЦИОНАЛНИХ МЕТОДА И ГРАФ НЕУРОНСКИХ МРЕЖА У ОБЛАСТИ СИСТЕМА ЗА ПРЕПОРУКУ	391-394
19. Ivan Činčurak, GENERISANJE TEKSTUALNOG OPISA SLIKE POMOĆU MAŠINSKOG UČENJA	395-398
20. Tanja Drčelić, CHATBOT NAMENJEN ZA POMOĆ STUDENTIMA PRILIKOM UPISA NA FAKULTET	399-402
21. Митар Перовић, АНАЛИЗА ПОДАТАКА СА ПЛАТФОРМЕ JOBERTY.RS ПOMOЋУ ТЕХНИКА МАШИНСКОГ УЧЕЊА	403-406
22. Tanja Vukmirović, RAZVOJ VS CODE EKSTENZIJE ZA VIZUELIZACIJU I MANIPULACIJU GRAPHQL ŠEMOM	407-410
23. Nikolina Šarenac, HTML VIZUELIZACIJA GRAPHQL SCHEMA-E U VS CODE EDITORU	411-414
24. Nikolina Stamenić, OBRADA I VIZUELIZACIJA PODATAKA UZ OSLONAC NA AWS	415-418
25. Olivera Kojić, Neven Kovački, VIŠE-PERIODNA REKONFIGURACIJA DISTRIBUTIVNE MREŽE PRIMENOM DINAMIČKOG PROGRAMIRANJA	419-422
26. Katarina Tomić, JEDNOSTEPENI I DVOSTEPENI TREKERI VIŠE OBJEKATA	423-426
27. Стефан Новаковић, РЕАЛИЗАЦИЈА РЕПРОДУКОВАЊА СЛИКЕ НА LED ДИСПЛЕЈУ СА ПАСИВНОМ МАТРИЦОМ	427-430

Radovi iz oblasti: Animacije u inženjerstvu

1. Aleksandra Bobić, UPOREDNI PRISTUP GENERISANJU GEOMETRIJE OKRUŽENJA VIDEO IGARA PRIMENOM FOTOGRAFIJE I DIGITALNOG VAJANJA	431-434
--	---------

Radovi iz oblasti: Čiste energetske tehnologije

1. Zlata Marijanović,
ENERGETSKA SANACIJA U CILJU POVEĆANJA ENERGETSKE EFIKASNOSTI NA PRIMERU
STAMBENOG OBJEKTA U INDIJI 435-438

УНАПРЕЂЕЊЕ ПРОИЗВОДНОГ СИСТЕМА ПРИМЕНОМ LEAN МЕТОДОЛОГИЈЕ**IMPROVEMENT OF PRODUCTION SYSTEM USING LEAN METHODOLOGY**Никола Андријашевић, Сања Бојић, *Факултет техничких наука, Нови Сад***Област – МАШИНСТВО**

Кратак садржај – У раду је описана примена LEAN методологије и њена примена у једном производном систему. Унапређење је спроведено у компанији Alutec d.o.o. из Србобрана. Представљене су теоријске основе LEAN методологије, и неки од алата који се користе у производним системима. У фокусу рада је опис производног процеса хладњака за светлосне групе данашњих аутомобила. Ови производи имају широку примену у данашњој ауто индустрији. Такође, у фокусу су и примењени LEAN алати од којих су: 5S Канбан, Андон светла, Мапирање тока вредности.

Кључне речи: LEAN методологија, LEAN алати – 5S, Канбан, Мапирање тока вредности.

Abstract – The paper describes the application of LEAN methodology and its application in one production system. The improvement was carried out in the company Alutec d.o.o. from Srbobran. The theoretical foundations of LEAN methodology and some of the tools used in production systems are presented. The focus of the paper is the description of the production process of radiators for the light groups of today's cars. These products are widely used in today's auto industry. Also, the focus is on applied LEAN tools, including: 5S Kanban, Andon lights, Value Stream Mapping.

Keywords: LEAN methodology, LEAN tools – 5S, Kanban, Mapping the value stream.

1. УВОД**1.1. Увод у предмет истраживања**

Упоредо са повећањем захтева за обраду материјала у металопрерађивачкој индустрији, развијају се нове методе и технологије које су тачније, продуктивније и економичне. Данас се за обраду плочастих материјала сечењем поред конвенционалних процеса обраде резањем све више употребљавају неконвенционални процеси обраде. Повећањем прецизности и тачности неконвенционалних обрадних система, смањује се потреба за вишеструким накнадним обрадама, уз истовремено постизање задовољавајуће продуктивности и економичности.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је била др Сања Бојић, ванр. проф.

Тема овог рада је оријентисана на упоређивање карактеристика три обрадна система за обраду материјала која користи различите технологије у посматраном производном систему „Alutec d.o.o.“ Србобран.

1.2. Краћи осврт на предузеће „Alutec d.o.o.“ и анализа производног програма

Од 2020. године Alutec d.o.o. проналази своје место у аутомобилској индустрији и у Србији. Великим константним улагањем и технолошким напредовањем предузеће се позиционирало међу веома траженим младим компанијама које теже да постану добављачи, односно део великих мултинационалних компанија у аутомобилској индустрији.

У нашем продајном програму, са лагером од преко 150.000 делова месечно који се производе у великим серијама од 1.000 до 200.000 производа који броје на стотине различитих производа од алуминијума. Ови производи своје место проналазе на деловима аутомобила, теретног програма као и мотора односно двоточкаша.

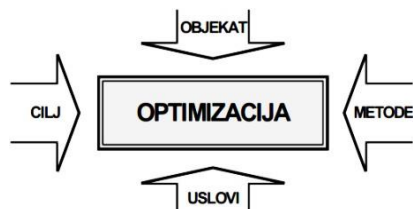
1.3. Потребе за оптимизацијом производног система

Свако предузеће уколоко жели да опстане и напредује на глобалном тржишту мора да прати промене и захтеве тржишта и да их истражује. Потреба за опстанком производње је неизбежна уз примену LEAN система од самог менаџмента па све до основне примене у производњи. Путем праћења тржишта предузеће идентификује један процес који може бити од велике користи, или пак створити одређене проблеме. У зависности од промена, менаџмент мора да прилагоди своје стратегије и организације. Уколико је предузеће инфериорно у праћењу промена, може се десити да изгуби корак са главним конкурентима јер може доћи до појаве нове технологије, па ће нови конкуренти на ефикаснији начин задовољити потребе тржишта. Циљеве пословања предузећа утврђене у процесу планирања неопходно је операционализовати у виду задатака које треба обавити, затим времена и трошкова њиховог извршавања, као и ресурса који су потребни за остваривање планираних циљева где LEAN методологија проналази своје место поред многобројних аспеката самог предузећа. Поред операционализације ових задатака потребно је урадити и анализу, односно оптимизацију са применом LEAN методологије.

1.4. Основни појмови оптимизације

У основне појмове оптимизације спадају (слика 1):

- Циљеви,
- Објекти,
- Методе,
- Услови.



Слика 1. Основни појмови оптимизације

Објекти оптимизације су многобројни и разноврсни. Конкретно у машинској техници објекти оптимизације могу бити процеси (обрадни, технолошки, производни...), технички системи (уређаји, машине, постројења, средства транспорта...), производи, делатност како што су инжењерска и људска (пројектовање, конструисање, истраживање, управљање, организовање...).

Оптимизација у машинству представља једну од кључних етапа у процесу формирања оптималног пројекта новог система или дефинисања решења и услова функционисања датог система. Овај процес чине четири основне етапе:

- Пројектовање структуре система,
- Постављање модела система,
- Оптимизација параметара модела система,
- Анализа добијених решења.

2. LEAN МЕТОДОЛОГИЈА И ОСНОВНИ ПОЈМОВИ

LEAN се може представити као начин размишљања о томе како организовати посао. Основни принцип LEAN-а је да посао треба да делује као одговор на захтев купцу. LEAN се дефинише и као системски приступ идентификацији и елиминацији расипања кроз континуирано побољшање. LEAN филозофија користи алате и технике за креирање и промену културе у циљу спровођења добре праксе, побољшања процеса који омогућава смањење расипања, побољшање тока, више се фокусира на потребе купаца и подржава поглед на исти процес.

Историја LEAN-а долази са истока односно Азије. Почетак развоја LEAN-а растао је заједно са развојем фабрике Тојота чији је оснивач Сакичи Тојода који се и назива „оцем јапанске индустрије“. Поред оснивача важну улогу у развоју имали су и његов син Кичиро Тојода као и трећи члан Тојота фамилије Енци Тојода који одлази у посету Фордове фабрике. Ова посета имала је кључни значај за развој Тојота фабрике. Огромну улогу инжењера у фабрици након послератног периода Другог светског рата имао је Таичи Оно који је осмислио систем где је за процес

потребан мањи број људи који ће опслуживати већи број различитих процеса.

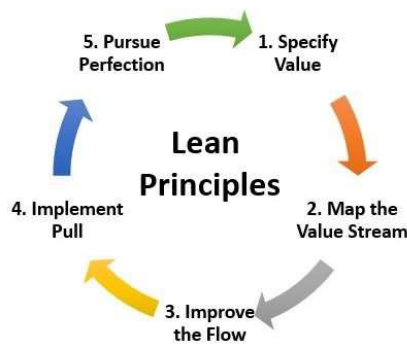
Систем пословања по Тојоти издваја осам LEAN губитака који су препознатљиви под називом „DOWNTIME“ губитци.



Слика 2. Осам LEAN губитака (DOWNTIME – Дефекти, Прекомерна производња, Чекање, Неискоришћени таленти, Транспорт, Инвентар, Покрет, Додатна обрада)

2.1. Основни принципи LEAN-а

Тојота је дефинисала пет основних принципа који омогућавају компанијама широм света да правилно искористи своје ресурсе и смањи време између почетка креирања производа па до завршетка уз помоћ LEAN алата и метода, а све то у циљу који је у фокусу да купац буде задовољан.



Слика 3. Пет основних LEAN принципа

1. Дефинисање вредности (*Specify Value*)
2. Успостављање тока настајањем вредности (*Map the Value Stream*)
3. Успостављање непрекидног тока (*Improve the Flow*)
4. Успостављање повратног система (*Implement Pull*)
5. Унапређивање (*Pursue Perfection*)

2.2. Основне методе и LEAN metodologije

Како би LEAN методологија била имплементирана унутар производње или организационе јединице неопходно је спровести низ метода које се планирају, спроводе и провере уз помоћ LEAN алата.

LEAN методе и алати који се сматрају за неке од основних добоко су међусобно повезани. Имају за циљ само једно, а то је елиминација и проналажење губитака у једном или више процеса био он сложен или прост.

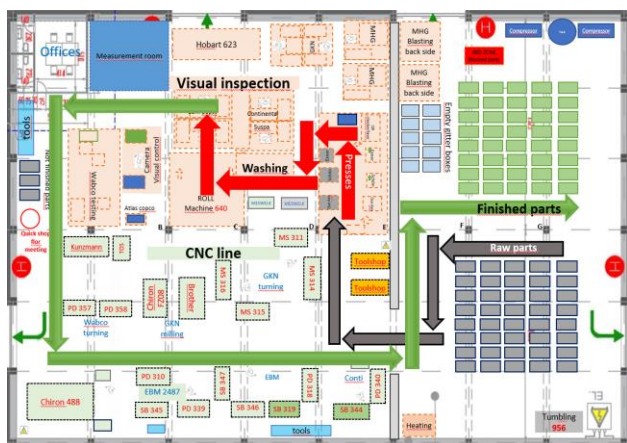
- 5S
- Just in Time
- Jidoka
- Kaizen
- Kanban
- Gemba

- Hoshin Kanri
- Total Productive Maintenance (TPM)
- Visual Management

3. АНАЛИЗА МОГУЋНОСТИ ПРИМЕНЕ LEAN МЕТОДА И АЛАТА У ПРОИЗВОДНОМ СИСТЕМУ Alutec d.o.o.

3.1. Шема постојећег стања производње

Из приказаног распореда радних јединица можемо видети да је производна јединица подељена на два дела и то где 2/3 су производни део са делом по хоризонталној линији. Испод хоризонталне линије зелено обележеним пољима налази се CNC линија, док горе од хоризонталне линије налази се визуелна контрола, процес пресовања, линија за монтажу, линија за машинско прање делова као и канцеларије у којима се налази и одељење контроле квалитета. 1/3 хале десно од зида налази се улазно-излазна зона, односно улазак и складиштење сирових производа и отпремања готових производа из производног дела.



Слика 4. Распоред радних јединица и ток материјала

3.2. Поставка производног система

Производња унутар производног система предузећа конципирана је на великосеријској производњи делова за ауто индустрију. Један од производних процеса који ћемо узети у анализу је процес пресовања хладњака за аутомобилску расвету. Ови производи су се развили заједно са развојем светских група у аутомобилима које данас виђамо свакодневно на улици. Такозвана LED (*Light Emitting Diode*) светла имају потребу за хладњацима односно делом склопа LED расвете који има за улогу да одведе топлоту са електро плоче на којој се и налазе светлеће диоде. Процес полази од сировина алуминијума AL EN AW 1050A, који долази од добављача. Затим, сировина се убацује у машину за мешање у којој се додаје цинков прах. Након мешања (тумбања) сировине, такви сирови комади се допремају до машине за пресовање коју опслужује оператер. Поставком сировине у алату и хладном деформацијом добија се првобитни изглед готовог производа који има даљи процес дораде на следећем алату где се одвија процес опсецања вишка материјала. Даљом дорадом производа, односно

процесом понављања пресовања у другом алату остварује се пробијање отвора. Следећи важан фактор формираног производа је машинско прање и припрема делова за визуелну контролу. Након опраних производа оператер визуелне контроле тражи грешке по познатом каталогу грешака настале процесом пресовања сировине. Након прегледаних производа следи паковање и отпрема у магацин одакле ће производ бити транспортован до крајњег купца.

4. ИМПЛЕМЕНТАЦИЈА LEAN-а У ПРОИЗВОДНОМ ПРОЦЕСУ

Међу постојећим поменутих алатима издвојили су се неки које су успешно нашли примену у постизању ефикасности како у економском смислу тако у техничком. Сва побољшања урађена доле наведеним алатима допринела су огромној уштеди и бољој организацији у раду. Алати који су примењени и који могу бити примењенији у предузећу су само један део LEAN методологије.

5S (*Five S*)

Овај алат LEAN-а представља његову основу, односно темељ за добар приступ приликом одржавања радног места чистим, уредним, организованим и безбедним радом. Имплементација овог алата је најлакша. Довољна је креативност која се претвара у одличне резултате који су видљиви и тренутни од тренутка имплементације. Назив 5S описује „5“ начела овог алата која почињу са словом „S“. Неколико примера 5S алата која су примењена у производњи компаније.

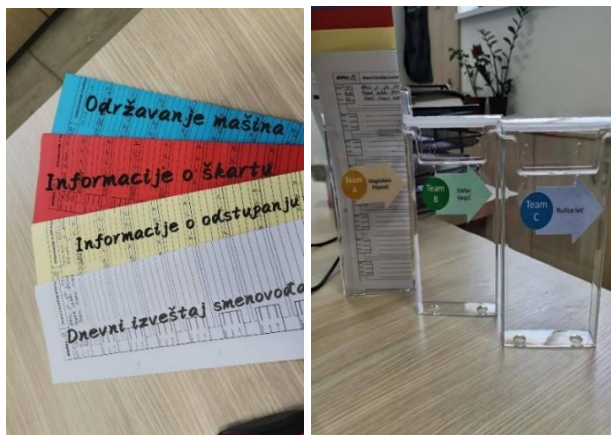


Слика 5. Заменски алати и магацин алата

Канбан (*Канбан*)

Канбан преведено на јапански значи табеларно планирање са картицама или знаковима. Приликом имплементације овог система планирања иницијално треба почети од једноставних корака, тј. одабрати делове процеса где је лако успоставити однос „купац – добављач“. Свака канбан картица даје информације за део процеса за који је она намењена. У производном систему можемо видети пример

табеларног планирања као и успостављен систем картица за пренос информација сменовођа. Са овим системом добили смо на времену приликом примопредаје смена, јер постоје три смене у којима се ради. Из разлога поштовања туђег времена уведен је систем преноса информација преко Канбан А4 документа који је успешно прихваћен. Овај систем има неколико добрих страна од којих су: Пренос добрих, прецизних, опширнијих информација, скраћено време преузимања смене, бољи преглед да ли се јавио квар на машини, бољи увид у аларме квалитета, брзо информисање који део производње је у застоју, могућност бржег доношења одлука за прерасподелу оператера (радника).



Слика 6. Канбан картице за пренос информација

5. ЗАКЉУЧАК

У фокусу пословања фирме мора бити купац јер он диктира темпо производње, односно производња се тако планира да испоштује све захтеве купца, посебно у данашње време када је све веома осцилаторно што је у огромној мери наметнула ауто индустрија широм света. Српска привреда је тек осетила ауто индустрију у последњих 15 година а посебно у последњих 5 година где су многе стране компаније донеле нове системе и побољшања из којих би многи могли да науче и подигну своја предузећа на бољи ниво пословања као њихови добављачи кроз које би исте компаније вршиле аудит и контроле добављача. Мала и средња предузећа могу да се опораве и крену са имплементирањем и инвестирањем у своје системе пословања као и у LEAN алате.

Постоје разне помоћи и субвенције за нову опрему од државе чији би циљ био да се тежи стабилизацији економске ситуације и уштеде.

Веома је важно напоменути да су спроведена унапређења извршили и запослени, својим идејама у оквиру свог радног времена, тако да су скоро све промене које су поменуте у обрађеном примеру организационог карактера, што значи да нису изискивале додатне трошкове и велика улагања.

Морамо имати на уму да примена LEAN концепта није лака, јер захтева, пре свега, инвестиције у виду образовања и стварања, односно подизања свести и културе људи о његовом значају и могућем ефекту на њихов рад, али сасвим сигурно доноси вишеструко корисне резултате.

6. ЛИТЕРАТУРА

- [1] Лазић, М.: Технологија обраде метала резањем, Машински факултет, Крагујевац, 2002.
- [2] Јанковић, П., Радовановић, М.: Савремени поступци обраде – пут ка еколошким технологијама, 1st International Conference Ecological safety in Post-Modern Environment, 26-27. јуна, Бања Лука,
- [3] Маринковић, Г.: Специфичности и карактеристике производног програма у прехранбеној процесној производњи, Факултет техничких наука, Чачак, 2012.
- [4] Бојић С., LEAN алати, презентација са предавања, Факултет техничких наука, Нови Сад, 2020. [5] Тодић, В., Пенезић, Н., Лукић, Д., Милошевић, М.: Технолошка логистика И предузетништво, ФТН издаваштво, Нови Сад, 2011.
- [5] Тодић, В., Пенезић, Н., Лукић, Д., Милошевић, М.: Технолошка логистика И предузетништво, ФТН издаваштво, Нови Сад, 2011. [6] Бекер И., Морача С., Лазаревић М., Шевић Д., Тешић З., Рикаловић А. и др.: LEAN Систем. Нови Сад: Факултет техничких наука, 2020.

Кратка биографија:



Никола Андријашевић рођен у Врбасу 1991. године. Мастер рад на Факултету техничких наука у Новом Саду, из области Логистика – Унапређење производног система пременом LEAN методологије одбранио је 2022. год.



Сања Бојић рођена је у Карловцу 1981. године. Докторирала је на Факултету техничких наука у Новом Саду 2013. године, а од 2019. године изабрана је у звање ванредног професора. Области истраживања су логистика, складишни системи и симулације токова материјала.

PODELA SISTEMA ZA KLIMATIZACIJU PROJEKAT MAŠINSKIH INSTALACIJA – VRF SISTEM

DIVISION OF THE AIR CONDITIONING SYSTEM MECHANICAL INSTALLATION PROJECT - VRF SYSTEM

Ilija Damjanac, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj – Rad se sastoji iz dva dela, podela sistema za klimatizaciju i projekat mašinskih instalacija – VRF sistem. Sistemi klimatizacije su ilustrovani pomoću pojednostavljenih šema u cilja lakšeg razumevanja načina rada. Takođe, bilo je riječi o određenim prednostima i nedostacima sistema. Projekat mašinskih instalacija, grejanje, klimatizacija i ventilacije prodajnog salona sa magacinom gotovih proizvoda, spratnosti P na kat. Parceli broj 4720/1 K.O. Kać. sistemom VRF, naspram koga je izračunata godišnja potrošnja toplote za grejanje.

Ključne reči: VRF sistem, grejanje, klimatizacija i ventilacija

Abstract – The paper consists of two parts, the variants of the air conditioning system and the project of mechanical installations - VRF system. Air conditioning systems are illustrated using simplified schematics to make it easier to understand how they work. Also, there were presented certain advantages and disadvantages of the system. Project of mechanical installations, heating, air conditioning and ventilation of a sales hall with a warehouse of finished products, floor space P on the floor. Plot number 4720/1 K.O. Say. by the VRF system, based on which annual heat consumption for heating was calculated.

Keywords: VRF system, heating, air conditioning and ventilation

1. UVOD

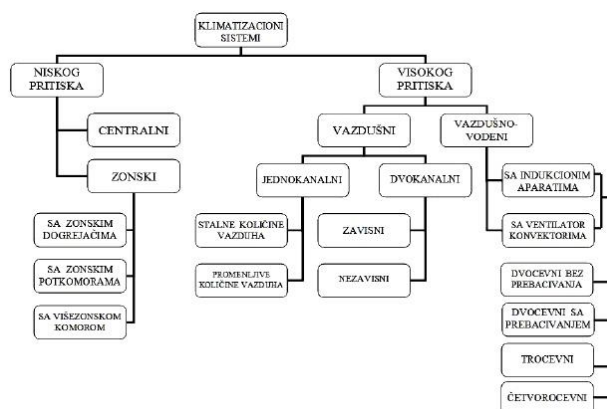
Sistemi klimatizacije ilustrovani su pomoću pojednostavljenih šema u cilja lakšeg razumevanja načina rada. Takođe, bilo je riječi o određenim prednostima i nedostacima sistema. Za projekat je rađen proračun toplotnih gubitaka i toplotnih dobitaka pomoću softvera Hanibal 5.0, a na osnovu koeficijenata prolaza toplote iz elaborata energetske efikasnosti, arhitektonskih podloga i šema stolarije. Toplotni gubici su računati prema standardu EN12831, dok su toplotni dobitci računati prema ASHRAE proceduri iz 1997. godine. Na osnovu sprovedenog proračuna gubitaka i dobitaka toplote za grejani i klimatizovani deo objekta usvojen je sistem za grejanje/klimatizaciju sa promenljivim protokom rashladnog fluida (variable refrigerant flow – VRF) koji se sastoji od jednog spoljašnjeg modula i više unutrašnjih jedinica.

NAPOMENA:

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

2. PODELA SISTEMA

Klasifikacija klimatizacionih sistema (slika 1.) može se izvršiti na različite načine, ali se podela najčešće zasniva na random fluidu pomoću koga se klimatizovani prostor greje ili hladi (vazduh ili vazduh i voda), kao i na brzinama strujanja vazduha kroz kanale (veće ili manja od 12 m/s). Kada se radi o većim brzinama, pad pritiska u sistemu je veliki, što podrazumeva i odgovarajući ventilator, pa su zato ovi sistemi visokog pritiska. Prema tome, razlikuju se sistemi niskog i visokog pritiska, koji su ili samo vazdušni ili tzv. vazdušno vodeni. Postoje i sistemi koji kao fluid za prenos toplote koriste isključivo vodu, koji se primenjuju u kombinaciji sa niskim od vazdušnih sistema. Po broju kanala za razvod vazduha, sistemi su jednokanalni ili dvokanalni, a pokoličini vazduha, sa stalnom i promenljivom količinom.



3. VAZDUŠNI SISTEMI

U grupu vazdušnih sistema spadaju sva postrojenja u kojima je prenos toplote na relaciji klimatizaciona komora-klimatizovana prostorija isključivo preko vazduha kao prenosioca toplote. Ovi sistemi se dele na jednokanalne i dvokanalne, od kojih obe vrste sistema mogu da rade sa konstantnom i promenljivom količinom vazduha, kao i sa malim odnosno velikim brzinama vazduha kroz kanale. Mogućnost ubacivanja vazduha u klimatizovan prostor u ovim sistemima su široke, pa se može postići uvek povoljna distribucija vazduha. Osim toga, postrojenja se lako i potpuno automatski prebacuju sa letnjeg na zimski režim, i obrnuto, što nije slučaj sa sistemima ostalih grupa.

U ovu grupu spadaju sledeći sistemi:

- centralni,
- zonski,
- jednokanalni visokog pritiska,

- dvokanalni nezavisnog dejstva,
- dvokanalni zavisnog dejstva.

4. VAZDUŠNO-VODENI SISTEMI SA INDUKCIJONIM APARATIMA

Za razliku od vazdušnih sistema, u kojima vazдушna struja kanalskom vezom spaja centralu i klimatizovani prostor, u ovoj grupi sistema veza se ostvaruje vodenom mrežom i vazdušnim kanalima, pa su radni fluidi voda i vazduh. Do ove kombinacije radnih fluida došlo se prilagođavanjem klimatizacionih postrojenja u modernim višespratnim zgradama sa velikim brojem prostorija i velikim udjelom prozorskih površina.

U ovu grupu spadaju sledeći sistemi:

- sa indukcionim aparatima,
- sa ventilator konvektorima
- dvocevni bez prebacivanja
- dvocevni sa prebacivanjem
- trocevni,
- četvoroccevni.

5. PRIMER PROJEKTOVANJA VRF SISTEMA

5.1 Tehnički opis

5.1.1 Uvodne napomene

Ovim radom obuhvaćene su instalacije grejanja, klimatizacije i ventilacije prodajnog salona sa magacinom gotovih proizvoda spratnosti P na kat. parceli broj 4720/1 K.O. Kač.

Rad je urađen na osnovu arhitektonskih podloga, podloga iz elaborata energetske efikasnosti, prema zahtevima investitora i prema važećim propisima i normativima za ovu vrstu instalacija

5.1.2 Instalacija grejanja i klimatizacije

Projektni uslovi za instalaciju grejanja su:

- Lokacija objekta: Kač (Novi Sad)
- Orjentacija objekta je uzeta prema situaciji
- Spoljni projektni uslovi (zima): $t_{sp} = -14,8 \text{ }^{\circ}\text{C}$, $\phi = 90 \%$ (Pravilnik o energetske efikasnosti zgrada)
- Koeficijenti prelaza toplote: preuzeti iz elaborate energetske efikasnosti
- Unutrašnje projektne temperature usvojene su prema nameni prostorija:
 - 1 Prodajni salon, $20 \text{ }^{\circ}\text{C}$
 - 2 Magacin gotovih proizvoda, $0 \text{ }^{\circ}\text{C}$ – ne greje se
 - 3 Hodnik, $20 \text{ }^{\circ}\text{C}$
 - 4 Kancelarija, $20 \text{ }^{\circ}\text{C}$
 - 5 Kancelarija, $20 \text{ }^{\circ}\text{C}$
 - 6 Kuhinja, $20 \text{ }^{\circ}\text{C}$
 - 7 Prostorija za magacionera, $20 \text{ }^{\circ}\text{C}$
 - 8 WC magacioneri, $15 \text{ }^{\circ}\text{C}$
 - 9 Prostorija za REK, $15 \text{ }^{\circ}\text{C}$ – ne greje se
 - 10 Garderoba, $20 \text{ }^{\circ}\text{C}$
 - 11 WC ženski, $15 \text{ }^{\circ}\text{C}$
 - 12 WC muški, $15 \text{ }^{\circ}\text{C}$

Projektni uslovi za instalaciju klimatizacije su:

- Spoljni projektni uslovi (leto): $t_{sp} = 34,0 \text{ }^{\circ}\text{C}$, $\phi = 33 \%$ (prema podacima iz ASHRAE 1997 za datu geografsku dužinu i širinu),
- Dobici toplote od osvetljenja i opreme preizetu iz elektro

dela I prema preporukama,

- Dobici od ljudi su: senzitivni: $75 \text{ W/}\checkmark$; latentni: $50 \text{ W/}\checkmark$.

- Unutrašnje projektne temperature usvojene su prema nameni prostorija :

- 1 Prodajni salon, $26 \text{ }^{\circ}\text{C}$
- 4 Kancelarija, $26 \text{ }^{\circ}\text{C}$
- 5 Kancelarija, $26 \text{ }^{\circ}\text{C}$
- 6 Kuhinja, $26 \text{ }^{\circ}\text{C}$
- 7 Prostorija za magacionera, $26 \text{ }^{\circ}\text{C}$
- 9 Prostorija za REK, $26 \text{ }^{\circ}\text{C}$
- Sve ne klimatizovane prostorije $30 \text{ }^{\circ}\text{C}$

Proračun toplotnih gubitaka i toplotnih dobitaka urađen je pomoću softvera Hanibal 5.0, a na osnovu koeficijenata prolaza toplote iz elaborata energetske efikasnosti, arhitektonskih podloga i šema stolarije. Toplotni gubici su računati prema standardu EN12831, dok su toplotni dobitci računati prema ASHRAE proceduri iz 1997. godine.

5.1.3 Instalacija ventilacije

U objektu prodajnog salona sa magacinom gotovih proizvoda spratnosti P predviđena je prirodna ventilacija prostorija preko spoljnih prozora, kao i veštačka ventilacija sanitarnih čvorova i garderobe u administrativnom delu objekta. U zoni spuštenog plafona u prostorijama koje se ventilišu postavlja se razvod od standardnih spiro cevi na koje se postavljaju aeroventili za izvlačenje vazduha. Izvlačenje se vrši preko kanalskih ventilatora, proizvođača "S&P" koji otpadni vazduh, preko spoljnih protivkišnih žaluzina izbacuju napolje. Predviđena su dva ventilatora, model VENT-150N za sanitarne čvorove i VENT-125N za garderobu. Na svim vratima prostorija gde postoji veštačko odsisavanje predviđene su tipske rešetke za prisiv vazduha u donjoj zoni.

5.1 Ulazni podaci za proračun

Spoljni zidovi	
Oznaka	k
[-]	[W/m ² /K]
SFZ	0,138

Tabela 1. Koeficijent spoljnog zida

Unutrašnje pregrade	
Oznaka	k
[-]	[W/m ² /K]
ZNP	0,136
P	0,243
UZ1	0,387
UZ2	0,305
MK	0,294

Tabela 2. Koeficijent unutrašnjih pregrada

Koeficijenate prelaza toplote smo dobili ubacivanjem slojeva omotača zgrade u program "URSA Građevinska fizika 2", koji sadrži katalog materijala, a ima i mogućnost da u okviru tog kataloga unesete svoj materijal koga nema u bazi. Na osnovu odabira materijala koji je dat u arhitektonskom projektu dobili smo pregled konstrukcija i njihovih koeficijenata prelaza toplote:

Spoljni zid - neventilisani, $U_{max} = 0,300 \text{ W/m}^2\text{K}$

Spoljni zid, $U = 0,138 \text{ W/m}_2\text{K}$, $T_i = 20,0 \text{ }^\circ\text{C}$

Prozor, $U_{\max} = 1,500 \text{ W/m}_2\text{K}$

PR3, $U = 1,460 \text{ W/m}_2\text{K}$, $T_i = 20,0 \text{ }^\circ\text{C}$

PR4, $U = 1,450 \text{ W/m}_2\text{K}$, $T_i = 20,0 \text{ }^\circ\text{C}$

Spoljna vrata od stakla, $U_{\max} = 1,600 \text{ W/m}_2\text{K}$

VR, $U = 1,490 \text{ W/m}_2\text{K}$, $T_i = 20,0 \text{ }^\circ\text{C}$

Izlog, $U_{\max} = 1,800 \text{ W/m}_2\text{K}$

PR1, $U = 1,440 \text{ W/m}_2\text{K}$, $T_i = 20,0 \text{ }^\circ\text{C}$

PR2, $U = 1,460 \text{ W/m}_2\text{K}$, $T_i = 20,0 \text{ }^\circ\text{C}$

Zid prema negrejanim prostorima, $U_{\max} = 0,400 \text{ W/m}_2\text{K}$

Zid prema negrejanom prostoru, $U = 0,136 \text{ W/m}_2\text{K}$, $T_i = 20,0 \text{ }^\circ\text{C}$

Meduspratna konstrukcija ispod negrejanog prostora, $U_{\max} = 0,300 \text{ W/m}_2\text{K}$

MK, $U = 0,294 \text{ W/m}_2\text{K}$, $T_i = 20,0 \text{ }^\circ\text{C}$

Pod na tlu, $U_{\max} = 0,300 \text{ W/m}_2\text{K}$

Pod na tlu, $U = 0,243 \text{ W/m}_2\text{K}$

Koeficijenti spoljašnjih zidova i unutrašnjih pregrada su ubačeni u softver Hanibal 5.0 i na osnovu koeficijenata, arhitektonskih podloga i šema stolarije. Dobijeni su toplotni gubici računati prema standardu EN12831, dok su toplotni dobici računati prema ASHRAE proceduri iz 1997. godine.

1 _ Prizemlje prodaja							
B r.	Naziv	P [m ²]	Visina [m]	Qt [W]	Qv [W]	Qrh [W]	Quk [W]
1	1. Prodajni salon	527,32	3,4	12451	10607	10546	33604

2 _ Prizemlje admin.							
B r.	Naziv	P [m ²]	Visina [m]	Qt [W]	Qv [W]	Qrh [W]	Quk [W]
2	3. Hodnik	11,29	3,05	403	205	226	834
3	4. Kancelarija	15,67	3,05	390	564	313	1267
4	5. Kancelarija	14,53	3,05	434	525	291	1250
5	6. Kuhinja	8,99	3,05	123	487	180	790
6	7. Prostorija za magac.	9,57	3,05	253	345	191	789
7	8. WC magacioneri	2,73	3,05	20	42	55	117
8	9. Prostorija za REK	2,35	3,05	ne greje se			
9	10. Garderoba	3,99	3,05	115	143	80	338
10	11. WC ženski	3,53	3,05	-14	54	71	111
11	12. WC muški	3,64	3,05	-7	57	73	123

UKUPNO							
UKUPNO	603,61		14168	13029	12026	39223	

Tabela 3. Prikaz gubitaka toplote po prostorijama

1 _ Prizemlje prodaja							
Br o j	Naziv	Mesec	Sa t	Tu n	Qos	Qla t	Quk
[-]	[-]	[-]	[-]	[C]	[W]	[W]	[W]
1	1. Prodajni salon	Juli	18	26	19427	3000	22427

2 _ Prizemlje admin.							
Br o j	Naziv	Mesec	Sa t	Tu n	Qos	Qla t	Quk
[-]	[-]	[-]	[-]	[C]	[W]	[W]	[W]
3	4. Kancelarija	Juli	17	26	2064	200	2264
4	5. Kancelarija	Juli	17	26	1929	100	2029
5	6. Kuhinja	Septem bar	12	26	724	200	924
6	7. Prostorija za magacionera	Septem bar	2	26	957	500	1457
8	9. Prostorija za REK	Septem bar	20	26	1768	0	1768

Tabela 4. Prikaz dobitaka toplote po prostorijama

6. METODE PRORAČUNA GODIŠNJE POTROŠNJE TOPLOTE- METODOM STEPEN DANA

Sam pojam STEPEN-DAN, koji je ključni element ove metode, predstavlja, na neki način, pokazatelj kretanja spoljne temperature vazduha u nekom mestu tokom perioda grejanja.

Godišnja potrebna toplota za grejanje, $Q_{H,nd}$ se prema SRPS EN ISO 13790, za sisteme koji rade bez prekida u zagrevanju, računa po formuli:

$$Q_{H,nd} = Q_{H,ht} - \eta_{H,gn} \cdot Q_{H,gn} \text{ [kWh/a]}$$

gde su:

$Q_{H,ht}$ – godišnja potreba toplote za nadoknadu gubitaka toplote [kWh/a]

$\eta_{H,gn}$ – faktor iskorišćenja dobitaka za period grejanja

$Q_{H,gn}$ – godišnja količina toplote koja protiče od unutrašnjih dobitaka toplote i dobitaka usled sunčevog zračenja [kWh/a]

Sprecifična godišnja potrebna toplote za grejanje, $Q_{H,an}$ predstavlja količnik godišnje potrebne toplote za grejanje i korisne površine zgrade:

$$Q_{H,an} = \frac{Q_{H,nd}}{A_f} \text{ [kWh/(m}^2\text{a)]}$$

gde je:

A_f – korisna površina zgrade [m²]

Godišnja potrebna toplota za nadoknadu gubitaka toplote obuhvata toplotu koja je potrebna za nadoknadu transmisijih Q_T i ventilacionih gubitaka toplote Q_v :

$$Q_{H,ht} = Q_T + Q_v \text{ [kWh/a]}$$

Godišnja količina toplote koja potiče od unutrašnjih dobitaka toplote i dobitaka usled sunčevog zračenja:

$$Q_{H,gn} = Q_{int} + Q_{sol} \text{ [kWh/a]}$$

gde su:

Q_{int} – godišnja količina toplote koja protiče od unutrašnjih dobitaka toplote [kWh/a]

Q_{sol} – godišnja količina toplote koja protiče od dobitaka usled Sunčevog zračenja [kWh/a]

Pa se godišnja potrebna toplota za grejanje može izraziti na sledeći način:

$Q_{H,nd} = (Q_T + Q_V) - \eta_{H,gn} \cdot (Q_{int} + Q_{sol})$ [kWh/a]
Godišnja potrebna toplota za nadoknadu gubitaka toplote računa se po formuli:

$$Q_{H,ht} = (H_T + H_V) \cdot 24 \cdot HDD \cdot 10^{-3} \text{ [kWh/a]}$$

gde su:

H_T – koeficijent transmisijonog gubitaka toplote [W/K]

H_V – koeficijent ventilacionog gubitaka toplote [W/K]

HDD – broj stepen dana za lokaciju zgrade (HDD – Heating Degree Days)

7. REZULTATI PRORAČUNA GODIŠNJE POTROŠNJE TOPLOTE

Koeficijent transmisijonog gubitka toplote H_T	407,01 W/K
Koeficijent ventilacionog gubitka toplote H_V	686,25 W/K
Godišnja potrebna energija za nadoknadu gubitaka toplote $Q_{H,ht}$	70292 kWh
Godišnja količina energije koja potiče od unutrašnjih dobitaka toplote $Q_{H,int}$	28251 kWh
Godišnja količina energije koja potiče od dobitaka usled sunčevog zračenja $Q_{H,sol}$	3204 kWh
Bezdimenzioni odnos toplotnog bilansa Y_H	0,447
Faktor redukcije za grejanje $a_{H,red}$	0,852
Faktor iskorišćenja dobitaka toplote za period grejanja $\eta_{H,gn}$	0,980
Godišnja potrebna energija za grejanje $Q_{H,nd}$	39151 kWh
Godišnja redukovana potrebna energija za grejanje $Q_{H,nd,red}$	33370 kWh
Specifična godišnja potrebna energija za grejanje $Q_{H,an}$	29,8 kWh
Energetski razred zgrade	B

Tabela 5. Prikaz godišnje potrebne energije za grejanje – godišnji proračun

Mesec	$Q_{H,ht}$	$Q_{sol,gl}$	$Q_{sol,c}$	$Q_{s,ol}$	Q_{li}	Q_{el}	Q_{int}	$Q_{H,gn}$	$Q_{H,nd}$
Okt	2817	328	39	36	7	709	164	235	271
Nov	1040	4	381	43	4	8	328	470	512
Dec	1481	2	296	33	9	5	328	474	507
Jan	1631	8	365	41	6	5	328	474	515
Feb	1277	5	490	59	54	132	328	460	515
Mar	1032	1	657	84	74	146	328	474	548
Apr	2845	341	47	38	8	709	164	235	273

Tabela 6. Prikaz godišnje potrebne energije za grejanje – mesečni proračun

8. ZAKLJUČAK

VRF uređaji sa promenljivim protokom rashladnog fluida su trenutno prepoznati kao najefikasniji KGH uređaji na tržištu kada je u pitanju ušteda električne energije, zbog čega je uzet u razmatranje ovog rada.

Pored toga što pružaju znatnu uštedu, VRF sistemi su ujedno i mnogo bolji za okolinu u poređenju sa klasičnim sistemima, imajući u vidu da stvaraju daleko manju emisiju CO₂ gasova. Pored toga, VRF sistemi smanjuju gubitke strujanja vazduha koji se pojavljuje u klasičnim ventilacionim sistemima zbog različitog koncepta prenosa energije.

VRF sistemi omogućavaju stalni rad uređaja, čak i u najzahtevnijim vremenskim uslovima. Pored toga, pametna kontrola uređaja omogućava licima ovlašćenim za održavanje ventilacionih sistema da sa udaljene tačke prate i podešavaju funkcije uz pomoć aplikacije za smart telefon. Relativno jednostavna ugradnja čini VRF sisteme idealnim kada su u pitanju visoke zgrade, kao i prodajni saloni sa magacinom, koji je razmatran u ovom radu.

9. LITERATURA

- [1] „Klimatizacija” Branislav Todorović
- [2] <https://dokumen.tips/documents/sistemi-klimatizacije.html> preuzeo 04.09.2022
- [3] <http://aliquantum.rs/wp-content/uploads/2011/12/05-Godisnja-potrosnja-energije-za-grejanje.pdf> preuzeo 06.09.2022
- [4] <https://www.toshiba-aircondition.com/sr/detaljnije-o-vrf-tehnologiji.html> preuzeo 08.09.2022
- [5] https://www.daikin.rs/sr_rs/about/daikin-innovations/variable-refrigerant-volume.html preuzeo

08.09.2022

- [6] <https://www.gradnja.rs/vrf-sistemi-klimatizacije-idealni-za-visoke-zgrade/> preuzeo 10.09.2022
- [7] <https://coolsistem.com/product/vrf-lg/> preuzeo 10.09.2022
- [8] <https://munjaklimatizacija.rs/blog/vrf-sistem-klimatizacije/> preuzeo 13.09.2022
- [9] URSA d.o.o. Beograd: Uputstvo za upotrebu programa-URSA GRADEVINSKA FIZIKA 2, Beograd.
- [10] Ministarstvo građevinarstva, saobraćaja i infrastrukture: Pravilnik o energetskej efikasnosti zgrada, („Sl. glasnik RS“, br.61/2011)
- [11] Miroslav Kljajić, Igor Mujan: Energetski pregledi, Novi Sad, 2018.
- [12] [PRAVILNIK O USLOVIMA SADRŽINI I NAČINU IZDAVANJA SERTIFIKATA O ENERGETSKIM SVOJSTVIMA ZGRADA.pdf](https://mgsi.gov.rs) (mgsi.gov.rs) preuzeo 17.09.2022

Kratka biografija:



Ilija Damjanac rođen je u Trebinju 1992. godine. Master rad na Fakultetu tehničkih nauka iz oblasti Energetskih tehnologija je odbranio 2022. godine.

Kontakt: ilijadamjanac13@gmail.com

**UPOREDNA ANALIZA SISTEMA GREJANJA STAMBENIH ZGRADA U SRBIJI I
ZEMLJAMA EVROPSKE UNIJE****COMPARATIVE ANALYSIS OF THE HEATING SYSTEM OF RESIDENTIAL
BUILDINGS IN SERBIA AND THE EUROPEAN UNION**

Perica Radovanović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj – Kroz ovaj rad izvršen je pregled stanja u sektoru zgradarstva sa aspekta potrošnje energije za grejanje. Kao i poređenje sa članicama EU. Prikazan je i pregled koji su energenti najviše zastupljeni za proizvodnju toplotne energije. Ovaj pregled nam daje priliku da sagledamo kakva je energetska efikasnost u sektoru zgradarstva, kao i da li ima prostora i mogućnosti za poboljšanje energetske efikasnosti.

Ključne reči: Energetska efikasnost, toplotna energija, energetski pasoši, sistemi grejanja

Abstract – Through this document, an overview of the situation in the building sector from the aspect of energy consumption for heating was carried out. As well as a comparison with EU members. An overview of which energy sources are most represented for the production of thermal energy is also presented. This review gives us the opportunity to see what energy efficiency is in the building sector, as well as whether there is room and opportunities for improving energy efficiency..

Keywords: Energy efficiency, heat energy, energy passports, heating systems

1. UVOD

Cene energenata su na istorijskom maksimumu, pa je pitanje koliko mi trošimo energije, to jest da li su nam stanovi i kuće energetski efikasni veoma aktuelno. Često je energetska efikasnost pogrešno interpretirana kao štednja energije, što neizbežno dovodi do smanjenja kvaliteta života.

Međutim, biti energetski efikasan znači da se energija upotrebljava na racionalan način, bez odricanja. Prema podacima Ministarstva građevinarstva, saobraćaja i infrastrukture, procenjuje se da preko 90% domaćinstava i stambenih zgrada u Srbiji nije izgrađeno u skladu sa važećim standardima u oblasti energetske efikasnosti.

Pored cene energenata, energetska efikasnost je jako bitna kada govorimo o očuvanju životne sredine. Sagorevanje fosilnih goriva utiče na klimatske promene i na pojavu ekstremnih vremenskih uslova.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Miroslav Kljajić, vanr. prof.

**2. PREGLED ZAKONODAVNIH OKRIVA U
OBLASTI ENERGETSKE EFIKASNOSTI**

Proces energetske sertifikacije zgrada u Srbiji započeo je 30. septembra 2012. godine, kada je primena pravilnika iz oblasti energetske efikasnosti zgrada postala obavezna. U Srbiji je EPBD implementirana kroz Zakon o planiranju i izgradnji. Na osnovu člana 201 Zakona o planiranju i izgradnji, doneti su Pravilnici kojima se bliže propisuju procedure za unapređenje energetske efikasnosti zgrada:

- Pravilnik o uslovima, sadržini i načinu izdavanja sertifikata o energetskim svojstvima zgrada, „Službeni glasnik RS“ broj 69/2012, 44/2018 – dr. zakon i 111/2022;
- Pravilnik o energetske efikasnosti zgrada, „Službeni glasnik RS“ broj 61/2011.

Pravilnikom o energetske efikasnosti zgrada se bliže propisuju zahtevana energetska svojstva, kroz definisanje metodologije proračuna termičkih svojstava zgrada, kao i propisivanjem zahteva za nove i postojeće zgrade.

Pravilnik o uslovima, sadržini i načinu izdavanja sertifikata o energetskim svojstvima zgrada uređuje proces energetske sertifikacije zgrada, način izdavanja i sadržaj sertifikata, i definiše energetske razrede za stambene i nestambene zgrade, i to nove i postojeće.

2.1 Elaborat energetske efikasnosti

Elaborat energetske efikasnosti je zvanični dokument kojim se regulišu energetske karakteristike zgrada. On pruža uvid u potrošnju energije za zadovoljenje životnih potreba, potreba za boravak i rad ljudi u predmetnom objektu.

Elaborat EE je elaborat koji se izrađuje kao obavezan deo projektne dokumentacije za dobijanje građevinske dozvole pri izgradnji novog objekta, ili odobrenja za izvođenje radova pri rekonstrukciji, adaptaciji ili sanaciji

2.2 Energetski pasoš

Energetski pasoš je sertifikat o energetskim svojstvima zgrade čiji je sadržaj i izgled usklađen sa Pravilnik o uslovima sadržini i načinu izdavanja sertifikata o energetskih svojstvima zgrada, a izdaje ga ovlašćena organizacija. Energetski pasoš je dokaz kvaliteta zgrade i može značajno uticati na cenu nekretnine.

Energetski razred zgrade se određuje na godišnjoj energiji potrebnoj za grejanje. Postoji 8 energetskih razreda od A+

do G. Primarni indikator za određivanje energetske razreda zgrade jeste teoretska/izračunata finalna energija za grejanje. Finalna energija za grejanje, kao specifična vrednost, uzeta je kao indikator za određivanje energetske razreda (u kWh/m^2 godišnje). Energetski razred A+ predstavlja najkvalitetniju zgradu, sa minimalnom potrebnom energijom za grejanje $\leq 15 kWh/m^2a$. Energetski razred G predstavlja zgrade sa veoma lošim karakteristikama i najvećom potrebnom energijom za grejanje $> 250 kWh/m^2a$.

2.3 Zakon o energetske efikasnosti i racionalnoj upotrebi energije

Ovim zakonom uređuju se uslovi i način efikasnog korišćenja energije i energenata; politika efikasnog korišćenja energije; sistem energetske menadžmenta; mere politike energetske efikasnosti: korišćenje energije u zgradama, kod energetske delatnosti i krajnjih kupaca, za energetske objekte i energetske usluge; energetske označavanje i zahtevi u pogledu eko-dizajna; finansiranje, podsticajne i druge mere u ovoj oblasti; osnivanje Uprave za finansiranje i podsticanje energetske efikasnosti kao i druga pitanja od značaja za prava i obaveze fizičkih i pravnih lica u vezi sa efikasnim korišćenjem energije. Ovim Zakonom Srbija se usklađuje sa novim direktivama EU u ovoj oblasti.

Jedna od novina koja je propisana zakonom o energetske efikasnosti i racionalnoj upotrebi energije jeste sistem energetske menadžmenta. Ovaj sistem predstavlja organizovano upravljanje energijom na način da obveznici sistema energetske menadžmenta izvršavaju zakonom propisane obaveze.

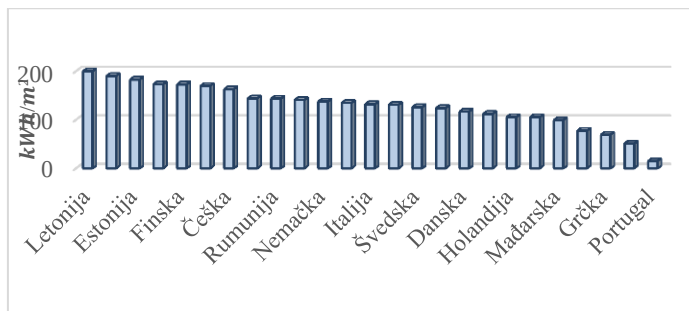
2.3 Pravilnik o kontroli sistema grejanja

Pravilnik o kontroli sistema za grejanje zgrada je donet osnovu člana 39. Zakona o energetske efikasnosti i racionalnoj upotrebi energije koji kaže da je korisnik sistema za grejanje, nazivne toplotne snage 70 kW i više u obavezi da obezbedi redovnu kontrolu tog sistema. Kontrolu rada sistema za grejanje obavlja ovlašćeno pravno lice ili preduzetnik u cilju procene stanja dostupnih delova sistema za grejanje, procene efikasnosti i toplotne snage sistema. Ovlašćeno lice nakon pregleda predlaže mere radi unapređenja energetske efikasnosti sistema.

3. POTROŠNJA TOPLOTNE ENERGIJE U STAMBENIM ZGRADAMA U SRBIJI I EU

4.1 Potrošnja energije u stambenim zgradama u članicama EU

Evropska unija dugo vremena prepoznaje da je energetske učinak u zgradama jako važan za ublažavanje posledica klimatskih promena. Zbog toga su formirane Direktive o energetske performansama zgrada, ili EPBD, i Direktiva o energetske efikasnosti, ili EED. Potrošnja energije u zgradama članica EU opada od 2008. godine, zbog napora učinjenih u stambenom sektoru. Potrošnja energije za grejanje u članicama EU, prikazan je u grafikonu 1.



Grafikon 1. Potrošnja energije za grejanje u članicama EU

Prosečna godišnja specifična potrošnja energije za grejanje je oko $124,03 kWh/m^2$ u EU, dok je najveća potrošnja energije u Letoniji i iznosi oko $199 kWh/m^2$.

4.2 Određivanje prosečne specifične potrebne energije za grejanje u Srbiji

Stambene zgrade u Srbiji su prema periodu izgradnje podeljene u osam etapa, za period od pre 1919. do 2022. godine. Podela stambenih zgrada u Srbiji prema periodu izgradnje prikazana je u tabeli 1.

Tabela 1. Podela stambenih zgrada u Srbiji

TIP	Period izgradnje	TIP	Period izgradnje
A	pre 1919	E	1971-1980
B	1919-1945	F	1981-1990
C	1946-1960	G	1991-2011
D	1961-1970	H	2012-2022

Kako bi se što preciznije odredila prosečna specifična potrošnja energije za grejanje stambenih zgrada potrebno je znati udeo površine tipa stambenih zgrada u ukupnoj površini stambenih zgrada u Srbiji. Podaci o površinama stambenih zgrada za tipove od A do G su prikazani u tabeli 2.

Tabela 2. Površine stambenih zgrada i udeli stambenih zgrada u ukupnoj površini stambenih zgrada za sve tipove

TIP	Površina stambenih zgrada sa jednim stanom [m^2]	Udeo površine tipa stambenih zgrada u ukupnoj površini [%]	Površina stambenih zgrada sa dva ili više stanova [m^2]	Udeo površine tipa stambenih zgrada u ukupnoj površini [%]
A	10 454 677	3,43	629 293	0,20
B	14 931 257	4,9	3 229 310	1,06
C	20 748 383	6,81	5 838 856	1,91
D	28 939 506	9,5	15 930 173	5,23
E	39 943 255	13,11	33 230 105	10,91
F	36 452 544	11,97	31 020 628	10,18
G	24 579 216	8,07	23 760 517	7,80
H	2 634 190	0,86	12 158 669	3,99

Ukupna površina: 304 480 579

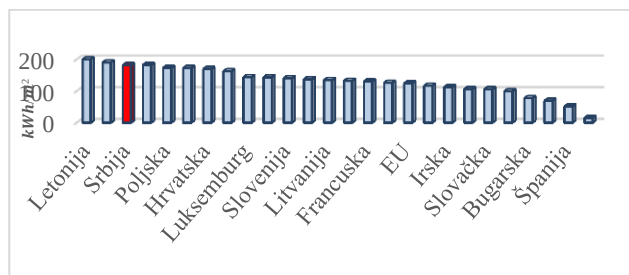
Podaci o potrošnji energije prikupljeni su iz energetske pasoša stambenih zgrada u Srbiji za sve tipove zgrada. Za svaki tip zgrade se na osnovu prikupljenih podataka iz energetske pasoša računa prosečna specifična potrebna energija za grejanje. Energetski pasoši su preuzeti sa digitalnog registra energetske pasoša CREP. Centralni registar energetske pasoša CREP je deo nacionalnog date management sistema (DMS), pušten je u rad u maju 2014. godine i njime upravlja Ministarstvo građevinarstva, saobraćaja i infrastrukture i Ministarstvo rudarstva u energetike. Prosečna specifična energija za grejanje za sve tipove prikazana je u tabeli 3.

Tabela 3. Prosečna specifična potrebna energija za grejanje

TIP	Prosečna specifična potrebna energija za grejanje za stambene zgrade sa jednim stanom [kWh/m^2]	Prosečna specifična potrebna energija za grejanje za stambene zgrade sa dva ili više stanova [kWh/m^2]
A	322,9	180
B	287	184,6
C	257	136,7
D	253,4	192,8
E	230,2	145
F	182,6	136,28
G	110,1	85
H	59,5	38,2

Na osnovu udela površine tipa stambenih zgrada u ukupnoj površini stambenih zgrada i prosečne specifične potrebne energije za grejanje za sve tipove može se izračunati prosečna specifična potrebna energija za grejanje za stambene zgrade u Srbiji i ona iznosi $181,6 kWh/m^2$.

Na osnovu dobijenih rezultata koji su prikazani u grafikonu 2, može se reći da se Srbija u poređenju sa članicama EU nalazi u samom vrhu po potrošnji energija za grejanje. Potrošnja energije za grejanje u Srbiji približna je potrošnji energije za grejanje u Belgiji, Estoniji, Poljskoj, Finskoj, ali je ta potrošnja daleko veće od prosečne potrošnje u EU.



Grafikon 2. Potrošnja energije za grejanje u Srbiji i u EU

4. CENA GREJANJA U SRBIJI I EU

4.1 Sistemi grejanja

Postoje razni sistemi grejanja, a svaki od njih ima svoje prednosti i mane. Bez obzira o kom sistemu grejanja da se radi on se sastoji od generatora toplote i opreme za distribuciju i predaju toplote. Sistemi grejanja se mogu podeliti u 2 kategorije:

1. lokalni sistemi grejanja;
2. centralni sistemi grejanja.

Dok se centralni sistemi grejanja mogu podeliti na etažno i na daljinsko grejanje.

Lokalni sistemi grejanja su oni kod kojih se toplotni izvor nalazi u samoj prostoriji koju je potrebno zagrevati. U grejnom uređaju se odvija proces sagorevanja goriva ili pretvaranja nekog drugog oblika energije u toplotu, koja

se preko površine peći prenosi na okolni prostor vazduhom ili direktno zračenjem. Od tehničke izrade peći, racionalnog izbora goriva i načina održavanja zavisi iskorišćenje toplote. Koeficijent iskorišćenja se kreće od 20% kod kamina pa do 85% kod nekih peći na gas.

Prema Republičkom zavodu za statistiku preko 50% domaćinstava u Srbiji koristi lokalno grejanje. Oko 75% domaćinstava koristi peći na drva, nakon peći na drva najviše su zastupljeni uređaji za grejanje koji koriste električnu energiju.

Etažno grejanje predstavlja proizvodnju toplotne energije u centralnoj jedinici sistema, koja se potom sprovodi do grejnih tela, a da se pri tome centralni kotao nalazi u sklopu stambene jedinice. Prednost etažnog grejanja u odnosu na daljinsko grejanje je ta što korisnik ima potpunu kontrolu kada će grejanje biti uključeno, to jest ne zavisi od toplane, a mane su održavanje sistema kao i to da je potreban prostor za kotao i ogrev.

Daljinsko grejanje predstavlja centralizovani sistem grejanja većeg broja potrošača. Parni ili vrelovodni kotlovi na čvrsto, tečno ili gasovito gorivo se obično koriste kao izvor toplote, dok se za transport toplote koristi posebna cevna mreža – toplovod. Toplovod mora biti dobro termički izolovan kako bi se gubici toplote od izvora do potrošača sveli na minimum.

4.2 Cene toplotne energije u Srbiji i u EU

Jako bitan faktor pri odabiru sistema grejanja i energenta je cena, to jeste koliko će nas na godišnjem nivou koštati grejanje. U nastavku će biti prikazane cene grejanja na različite energente kako u Srbiji tako i u EU.

Vrste energenata (energije) koje se koriste za grejanje kao i udeli energenata (energije) u ukupnoj potrošnji prikazani su u tabelama 4 i 5.

Tabela 4. Udeli energenata (energije) u ukupnoj potrošnji energenata (energije) za grejanje u Srbiji

Vrsta energenata (energije)	Ugalj	Biomasa (drvo)	Mazut i ulje za loženje	Prirodni gas	Električna energija	Obnovljivi izvori energije	Daljinsko grejanje
Udeo energenata (energije) u ukupnoj potrošnji [%]	11,4	53,1	0,9	9,6	6,2	0,6	18,3

Tabela 4. Udeli energenata (energije) u ukupnoj potrošnji energenata (energije) za grejanje u EU

Vrsta energenata (energije)	Ugalj	Biomasa (drvo)	Mazut i ulje za loženje	Prirodni gas	Električna energija	Obnovljivi izvori energije	Daljinsko grejanje
Udeo energenata (energije) u ukupnoj potrošnji [%]	4,2	18,5	15,6	38	5,2	8,3	10,2

Cene energije za grejanje u Srbiji i u EU prikazane su u tabeli 3.

Sada kada su nam poznate cene i udeli goriva u potrošnji energije za grejanje možemo odrediti prosečnu cenu energije za grejanja u EU i Srbiji, i koja iznosi $0,048 €/kWh$ za Srbiju i $0,101 €/kWh$ za EU.

Na osnovu prosečne cene energije za grejanje i prosečne specifične potrošnje energije za grejanje može se na sledeći način odrediti i prosečna cena grejanja po m^2 u EU i Srbiji.

$$C_{EU} = 0,101 \cdot 124,03 = 12,4 \text{ €/m}^2$$

$$C_{SRB} = 0,048 \cdot 181,6 = 8,7 \text{ €/m}^2$$

Tabela 3. Cene toplotne energije u Srbiji i EU

Vrsta energenata (energije)	Cene u EU €/kWh	Cene u Srbiji €/kWh
Električna energija	0,2369	0,0811
Prirodni gas	0,0782	0,0342
Daljinsko grejanje	0,065	0,06
Obnovljivi izvori energije	0,0592	0,02
Ugalj	0,11	0,061
Drva	0,067	0,04
Ulje za loženje	0,2	0,11

I ako građani Srbije u proseku potroše više energije kako bi zagrejali jedan m² stambenog prostora od građana EU, ali ipak građane Srbije manje košta kako bi zagrejali jedan m² stambenog prostora zbog niske cene toplotne energije.

5. ZAKLJUČAK

Stambeni objekti predstavljaju velike potrošače energije u EU, a pogotovu u Srbiji. Po potrošnji energije za grejanje Srbija se nalazi u samom vrhu u poređenju sa članicama EU. Prosečna potrošnja energije za grejanje u Srbiji je oko 181 kWh/m², dok je prosečna potrošnja energije za grejanje u EU oko 120 kWh/m². Pored toga što razbacujemo energiju loše stanje zgrada i kuća u Srbiji doprinosi i zagađenju vazduha. To nam govori činjenica da pojedini gradovi u Srbiji, tokom zimskog perioda kada traje grejna sezona imaju najlošiji kvalitet vazduha u Evropi. Jedan od razloga zbog čega udišemo loš vazduh je taj što se veliki deo domaćinstva i dalje greje na čvrsto gorivo, prvenstveno uglja i drva.

Jedna od ključnih barijera za povećanje energetske efikasnosti su nerealno niske cene energenata, to jest cena toplotne energije. Građani Srbije godišnje manje plaćaju grejanje i ako potroše više energije od građana EU. Usled niske cene energije korisnici nemaju ekonomski interes da ulažu u projekte povećanja energetske efikasnosti. Pored finansijske, barijere imaju izraženu i socijalnu dimenziju. Energija u Republici Srbiji nije roba. Znatno deo brige o socijalnom statusu stanovništva odvija se preko cene energije.

Srbija je donošenjem Zakona o planiranju i izgradnji kao i Zakona o energetskej efikasnosti i racionalnoj upotrebi energije, napravila prve korake ka povećanju energetske efikasnosti. Potrebno je da se ti zakoni dobro implementiraju u praksi i rezultati će vremenom biti приметni.

6. LITERATURA

[1] Službeni glasnik RS br. 61/2011: Pravilnik o energetskej efikasnosti zgrada

[2] Službeni glasnik RS br. 69/2012, 44/2018 – dr. zakon i 111/2022: Pravilnik o uslovima, sadržini i načinu izdavanja o energetskim svojstvima zgrada

[3] Službeni glasnik RS br. 40/2021: Zakon o energetskej efikasnosti i racionalnoj upotrebi energije

[4] Milica Jovanović Popović, Dušan Ignjatović, Ana Radivojević, Aleksandar Rajčić, LJiljana Đukanović, Nataša Čuković Ignjatović, Miloš Nedić, Nacionalna tipologija stambenih zgrada Srbije, Arhitektonski fakultet Univerziteta u Beogradu, 2013.

[5] Maja Todorović, Energetska efikasnost sistema grejanja i klimatizacije, Mašinski fakultet Beograd, 2013.

[6] The official web site of the european commission: https://ec.europa.eu/energy/eu-buildings-factsheets_en [datum pristupa 25.06.2022.].

[7] Републички завод за статистику: <https://www.stat.gov.rs/> [datum pristupa 27.06.2022.].

[8] Централни регистар енергетских пасоша: <https://crep.gov.rs/EnergetskiPasosi.aspx> [datum pristupa 14.08.2022.].

[10] The official web site of the european statistics – eurostat: <https://ec.europa.eu/eurostat/web/main/home> [datum pristupa 11.10.2022.].

Kratka biografija:



Perica Radovanović rođen je u Bijeljini 1996. god.

Master rad na Fakultetu tehničkih nauka iz oblasti Mašinstva – Energetika i procesna tehnika odbranio je 2022.god.
kontakt: radovanovic.perica@gmail.com

IZBOR POGONSKOG PODSISTEMA ELEKTRIČNOG VOZILA SA NAGLASKOM NA MAŠINE SPECIJALNE KONSTRUKCIJE

ELECTRIC VEHICLE PROPULSION SUBSYSTEM CHOICE WITH EMPHASIS ON MACHINES OF SPECIAL CONSTRUCTION

Nevena Pantić, Bane Popadić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U modernom dobu inovativna rešenja nude različite izvore energije koji mogu pokretati vozila. Još u 19. veku, naučnici su spoznali značaj električnih vozila, ali u pokušaju da ih tada distribuiraju na tržište nisu uspeali. Danas, sa napretkom tehnologije moguća je globalna upotreba električnih vozila. U ovom radu analizirana su električna vozila sa naglaskom na njihov električni pogonski podsystem. Kao idejno rešenje, odabrana je električna pogonska mašina specijalne konstrukcije, Brushless DC mašina koja bi mogla zadovoljiti potrebe vozila sa električnim pogonom.

Ključne reči: Električno vozilo, Električni podsystem, Hall senzor, Trapezna komutacija

Abstract – In the modern age, innovative solutions offer different energy sources that can drive vehicles. Back in the 19th century, scientists realized the importance of electric vehicles, but in an attempt to distribute them to the market, they failed. Today, with the advancement of technology, the global use of electric vehicles is possible. In this paper, electric vehicles are analyzed with an emphasis on their electric drive subsystem. As a conceptual solution, an electric drive machine of a special construction, a Brushless DC machine that could be used in a vehicle with an electric drive.

Keywords: Electric vehicle, Electric vehicle subsystem, Hall sensor, Six-Step commutation

1. UVOD

Inovativna tehnička rešenja mogu ponuditi različite izvore energije koji se koriste za pogon vozila. Danas, pored tečnih goriva koristi se i električna energija. Vozila koja koriste tu vrstu izvora energije nazivaju se električna vozila (eng. *Electric vehicle* – EV).

Iako vlada mišljenje da su EV tehnološko dostignuće novijeg doba, oni su zapravo stariji izum od vozila koja koriste motore sa unutrašnjim sagorevanjem (eng. *Internal combustion engine* – ICE).

Prvo EV konstruisao je Francuz Gustave Trouvé 1881. godine, to je bio tricikl pogonjen jednosmernim motorom koji je napajan energijom iz baterije.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio doc dr Bane Popadić.

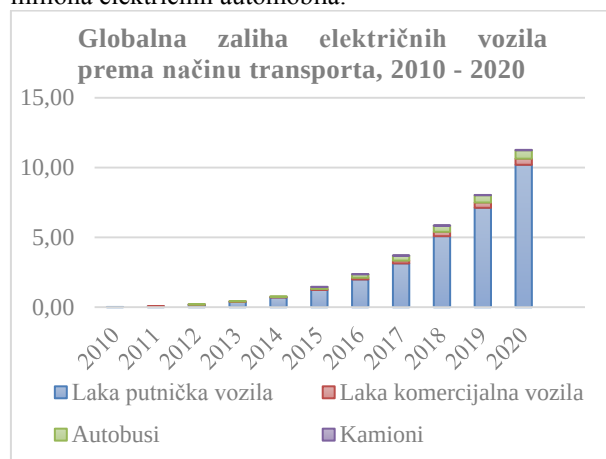
Mnogi naučnici su radili na sličnim projektima u cilju unapređenja nove tehnologije, ali na početku nisu uspeali privući pažnju javnosti jer brzina od 15 km/h i domet od 16 km, nisu impresionirali potencijalne kupce.

2. RAZVOJ ELEKTRIČNOG POGONA U VOZILIMA

Krajem 19. veka, masovna proizvodnja baterija, našla je primenu u EV. Razvojem modernijih saobraćajnica nastaje potreba za druskim vozilima boljih performansi. EV i dalje su bila preferirana u odnosu na svoje konkurente: vozila sa ICE motorom i vozila na parni pogon. Zbog nepouzdanosti i ručnog načina pokretanja vozila sa ICE motorom, takođe zbog niske efikasnosti i potrebe za osvetljenjem vozila na parni pogon, EV je bilo mnogo pouzdanije u odnosu na svoje konkurente.

Iako je razvoj EV krenuo u dobrom smeru, rad na poboljšanju tehnologije u pogonu sa ICE motorom, preokrenuo je uloge na tržištu. Tehnologija EV je stagnirala, dok je razvoj ICE vozila napredovao brzim tempom. Godine su prolazile bez većeg oživljavanja upotrebe EV.

Od početka 1990ih evolucija EV je ponovo u usponu. Kao prekretnica može se smatrati razvijanje ekološke svesti društva, koja je podstakla borbu za zdraviju životnu sredinu. Uloženi su veliki naponi i napravljeni značajni koraci u razvoju tehnologije, što je dovelo do ubrzanog rasta tržišta. Prema izveštaju (grafik 2.1.) Svetske agencija za energiju (IEA – *International Energy Agency*) krajem 2020. god. na svetskim putevima bilo je oko 10 miliona električnih automobila.



Grafik 2.1. – Globalna zaliha električnih vozila u svetu prema načinu transporta, 2010. – 2020. [1]

Evropa je po prvi put prestigla Kinu kao najveće svetsko tržište električnih automobila. Takođe, povećane su i globalne zalihe električnih autobusa i kamiona.

3. ARHITEKTURA ELEKTRIČNIH VOZILA

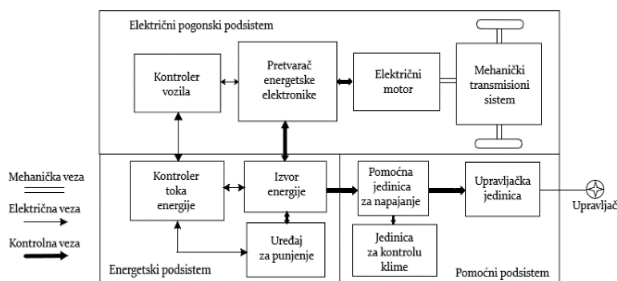
Osnovni principi u EV i u vozilu sa ICE motorom su slični, ali postoje neke bitne razlike. EV u svom pogonskom podsistemu koriste električni motor, koji može biti kontrolisan različitim tehnikama upravljanja. Za njegov rad potrebna je energija koja se obezbeđuje iz različitih izvora, kao što su hemijske baterije, gorive ćelije, ultrakondenzatori i zamajci. EV su mnogo fleksibilnija u odnosu na vozila sa ICE motorom, zbog odsustva složenih mehaničkih komponenti koje se koriste za pokretanje konvencionalnih vozila.

Prvobitno, konfiguracija EV bila je velikoj meri zasnovana na konfiguraciji konvencionalnih vozila uz specifične izmene koje su se odnosile na rezervoar za skladištenje energije i na pogonsku mašinu.

Konstrukcija prvobitnog EV imala je nedostatke koji su bili motiv za stvaranje originalnog dizajna koji će kompenzovati većinu problema i sa time povećati fleksibilnost električnog pogona. Velika masa vozila, smanjena fleksibilnost i performanse vozila, bili su uzrok postepenog nestajanja ove vrste električnog vozila.

Savremeni pogonski sklop električnog vozila ilustrovan je na slici 3.1. Pogonski sklop sastoji se od tri glavna podsistema:

- električni pogonski podsistem,
- energetska podsistem i
- pomoćni podsistem.



Slika 3.1. Savremeni pogonski sklop električnog vozila [2]

Električni pogonski podsistem obuhvata kontroler vozila, pretvarač energetske elektronike, električni motor i mehanički prenosni sistem.

Energetski podsistem uključuje izvor energije, jedinicu za kontrolu toka energije i uređaj za punjenje baterije. Pomoćni podsistem sadrži jedinicu za servo upravljanje, kontroler klime (temperature), takođe sadrži i pomoćnu jedinicu za napajanje. Na slici 3.1. strelicama je ukazano tok energije unutar EV.

Može se primetiti da između određenih jedinica tok energije može da bude u oba smera. Regenerativna energija kočenja proizvedena prilikom usporavanja vozila konvertuje se i skladišti u bateriji.

Većina baterija zajedno sa kondenzatorima i zamajcima, koji se koriste u savremenim pogonskim sklopovima električnih vozila, kompatibilni su da skladište povratnu energiju nastalu tokom regenerativnog kočenja.

4. POGONSKI PODSISTEM ELEKTRIČNOG VOZILA

Električni pogonski sistem je jezgro električnog vozila, a njegove glavne komponente su električni motor, uređaji energetske elektronike i kontrolni uređaji. Odabir električnog pogonskog sistema električnog vozila najvećim delom zavisi od tri faktora: željenih performansi, ograničenja vozila i izvora energije. Željene performanse određene su prema očekivanjima vozača, što uključuje maksimalnu brzinu, ubrzanje, kočenje, domet i slično. Zatim, odabir električnog pogona zavisi i od tipa vozila, njegove težine i nosivosti. Takođe, vrsta uređaja za skladištenje električne energije u vozilu važna je stavka pri kreiranju električnog pogona.

Jedan od važnijih zadataka jeste izbor tipa električne mašine u električnom pogonu, a na to utiču sledeći parametri:

- snaga motora,
- veliki polazni obrtni momenat,
- širok opseg brzina,
- efikasnost i pouzdanost,
- robusnost,
- jednostavno upravljanje,
- cena i
- jednostavna konstrukcija i dimenzije.

Za razliku od mašina koje se koriste u industriji, električne mašine u okviru pogonskog podsistema EV i HEV zahtevaju česta pokretanja i zaustavljanja, visoke stope ubrzanja/usporavanja, veoma širok opseg radnih brzina itd. Standardi određuju koja će se vrsta električne mašine koristiti, obično su to mašine koje imaju ili jednostavnu konstrukciju ili jednostavno upravljanje ili poboljšanu efikasnost.

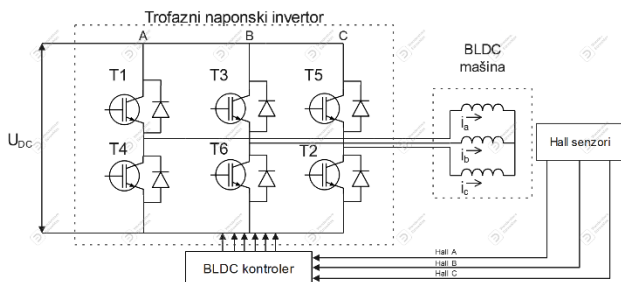
4.1 IZBOR MAŠINE SPECIJALNE KONSTRUKCIJE ZA POGONSKI PODSISTEM ELEKTRIČNOG VOZILA

Upotrebom energije permanentnih magneta za stvaranje pobude može se konstruisati mašina sa velikom gustinom snage, velikom brzinom i visokim koeficijentom efikasnosti. Prethodno je naglašeno da su navedene prednosti naročito važne u konstrukciji pogonskog podsistema EV i HEV. Najperspektivnija mašina specijalne konstrukcije iz familije mašina sa permanentnim magnetima je jednosmerna mašina bez četkica (BLDC – eng. *Brushless DC machine*). U literaturi može se naći pod nazivom elektronski komutovana mašina jednosmerne struje.

BLDC mašina slična je po konstrukciji i principu rada asinhronim i jednosmernim mašinama. Kao i ostale mašine, takođe i BLDC mašina ima stator i rotor. Za razliku od tradicionalnih jednosmernih mašina koje koriste mehanički komutator, BLDC mašina ostvaruje komutaciju bez mehaničkog kontakta upotrebom električnog prekidačkog kola. Da bi se kontrolisala brzina i smer rotacije koristi se senzor položaja rotora, koji šalje signale na ulaz BLDC kontrolera. Senzor položaja rotora, kontrolno kolo i energetski pretvarač snage deo su sistema BLDC mašine.

Konfiguracija pogonskog kola za upravljanje BLDC mašinom, čiji su statorski namotaji povezani u spregu Y,

prikazana je na slici 4.1. Naponski inverter ima 6 prekidača T1, T2, T3, T4, T5 i T6 koji su uključeni ili isključeni u zavisnosti od upravljačkih signala koje šalje BLDC kontroler. Na taj način se vrši komutacija statorskog polja, odnosno menja struja u statorskim namotajima i sa time se generiše obrtni momenat. Hall senzor je najčešće ugrađen u BLDC mašinu. Kada magnetni pol rotora prođe pored Hall senzora, on će dati signal visokog ili niskog nivoa, što ukazuje da severni magnetni pol ili južni magnetni pol prolaze pored senzora.

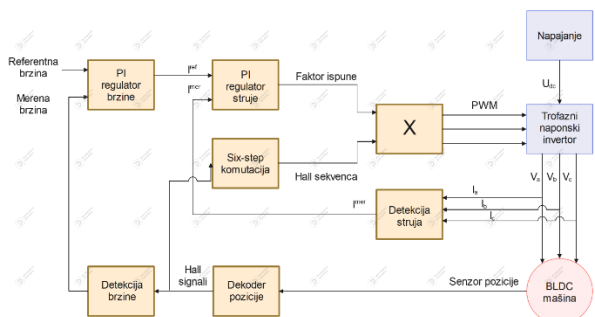


Slika 4.1. Šematski prikaz pogonskog kola za upravljanje BLDC mašinom

Komutacija se odvija sa promenom položaja magneta, tačnije kada se rotor obrne za električni ugao od 60°. Dakle, postoji krug podeljen u šest sektora u kojima se može naći rotor, a sa tim i šest potrebnih Hall kombinacija. Ova vrsta komutacije se naziva šest-koračna komutacija (eng. *six step commutation*) ili trapezna komutacija. Odlikuje je jednostavan algoritam, zbog toga je jeftin izbor, a može se primeniti u pogonima koji zahtevaju velike brzine i veliki obrtni moment.

5. SOFTVERSKA REALIZACIJA UPRAVLJANJA BLDC MAŠINOM

Softverska realizacija upravljanja BLDC mašinom je realizovana u MATLAB/SIMULINK radnom okruženju. Za potrebe eksperimentalnog istraživanja preuzet je oficijalni MATLAB model pod nazivom „Six-step commutation of BLDC motor using sensor feedback“ [3]. U primeru je implementirana tehnika šest-koračne ili trapezne komutacije za kontrolu brzine trofazne BLDC mašine. Na slici 5.1. data je šema upravljanja sa BLDC mašinom koja je regulisana po brzini.



Slika 5.1. Šema pogona upravljanja sa BLDC mašinom regulisanom po brzini

U radu je korišćena BLDC mašina kineskog proizvođača „JKONG MOTORS“, serijskog tipa „JK86BLS58e-X002“. Mašina je konstruisana sa ugrađenim Hall sondama, koje su pozicionirane u posebno izolovanom

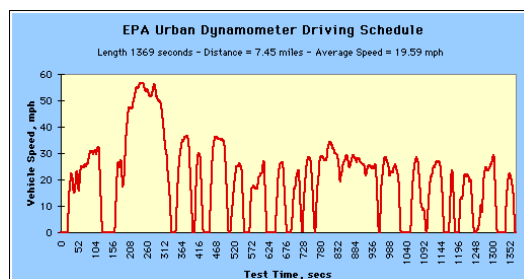
kućištu. U postupku eksperimentalnog rada određeni su neki parametri navedenog tipa mašine, a pojedini su preuzeti iz specifikacija proizvođača. Parametri BLDC mašine u MATLAB modelu su izmenjeni i prilagođeni.

5.1 TESTIRANJE POGONSKOG PODSISTEMA ELEKTRIČNOG VOZILA

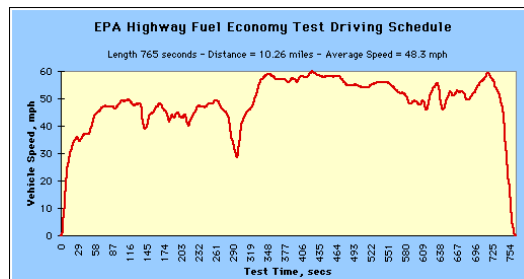
Za procenu performansi vozila koriste se standardizovani ciklusi vožnje preporučeni od strane agencije EPA (eng. *EPA - Environmental Protection Agency*) iz SAD.

Standardni ciklusi vožnje koriste se za ispitivanja kao što su: merenje potrošnje goriva, emisija zagađenja vazduha, analiza stanja baterije i potrošnja energije kod EV, takođe, koriste se za ispitivanje pojedinih komponenti vozila, pogonskog podsklopa, transmisije i slično.

Na slici 5.2. prikazan je karakterističan ciklus vožnje u gradskim uslovima, a na slici 5.3. je prikazan karakterističan ciklus vožnje na „otvorenom“ putu. Za testiranje pogonskog podsistema električnog vozila sa BLDC mašinom specijalne konstrukcije odabran je karakterističan ciklus vožnje u gradskim uslovima.



Slika 5.2. – Karakteristični ciklus vožnje u gradu [4]



Slika 5.3. – Karakteristični ciklus vožnje po „otvorenom“ putu [4]

U primeru je testiran dinamički odziv pogona u opsegu brzina koje su određene na osnovu ciklusa vožnje (prvih 140 s). Takođe, testiran je i odziv brzine sa promenom opterećenja. Shodno tome, potrebno je odrediti moment opterećenja mašine koji je uslovljen „profilom puta“.

Na dinamičke performanse vozila utiče rezultanta sila: sila otpora vazduha, otpor kotrljanja, pogonska (vučna) sila i sila otpora uspona. Kretanje vozila uslovljeno „kompleksnom spregom“ sila može se opisati sledećom jednačinom [2]:

$$m_v \frac{d}{dt} V_x = F_x - F_{aero} - F_{roll} - m_v \cdot g \cdot \sin \alpha \quad (5.1)$$

m_v – masa vozila,

V_x – brzina vozila,

F_x – pogonska (vučna) sila,

F_{aero} – sila otpora vazduha,

F_{roll} – sila otpora kotrljanja i

$m_v \cdot g \cdot \sin \alpha$ – sila otpora uspona (postoji ako je $\alpha \neq 0$).
Brzina vozila i izlazna mehanička snaga mogu se opisati sledećim jednačinama [2]:

$$V_x = r_w \cdot \frac{\omega_r}{g_{dr}} \cdot (1 - s_x) \quad (5.2)$$

$$P_x = T_w \cdot \frac{V_x}{r_w} = T_e \cdot \omega_r \cdot \eta_{dr} \cdot (1 - s_x) \quad (5.3)$$

r_w – poluprečnik točka,

ω_r – ugaona brzina obrtanja električnog motora,

s_x – prosečno proklizavanje i

g_{dr} – prenosni odnos mehaničkog podsistema.

T_w – pogonski otporni moment na točku,

T_e – mehanički moment opterećenja električnog motora i

η_{dr} – prosečna efikasnost izlaznog mehaničkog sklopa.

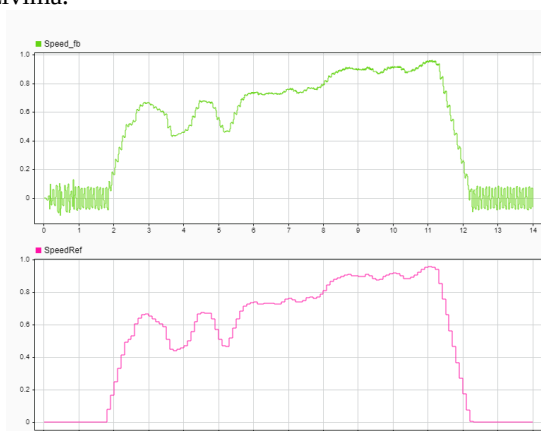
Moment opterećenja može se odrediti na osnovu poznavanja vrednosti pogonske sile. Ona se računa na osnovu prethodno napisane jednačine (5.1), a potom se moment opterećenja odredi na osnovu sledeće relacije [2]:

$$F_x = \frac{P_x}{V_x} = \frac{T_e \cdot g_{dr} \cdot \eta_{dr}}{r_w} \quad (5.4)$$

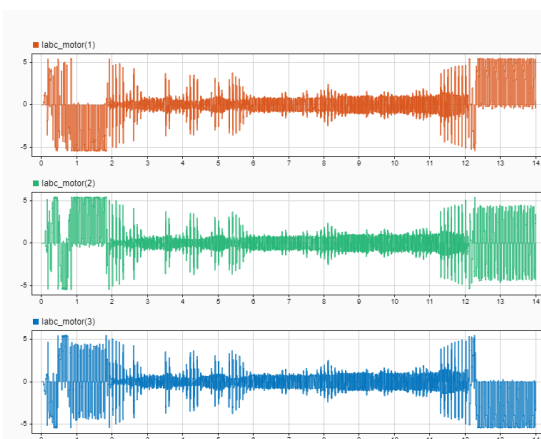
5.2 REZULTATI RAČUNARSKE SIMULACIJE

Rezultati računarske simulacije za slučaj upravljanja BLDC mašinom metodom trapezne komutacije u MATLAB/SIMULINK radnom okruženju prikazani su na slici 5.4.

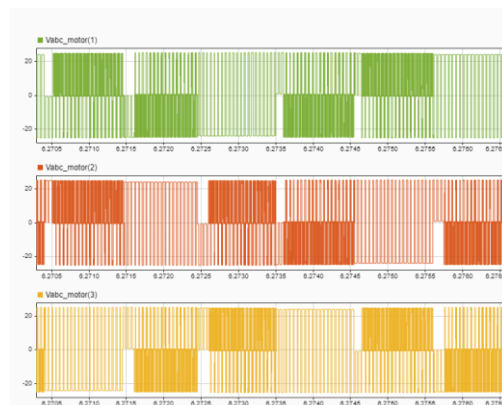
U simulaciji su posmatrane veličine od interesa, vršena je njihova analiza i upoređivane su sa predviđenim odzivima.



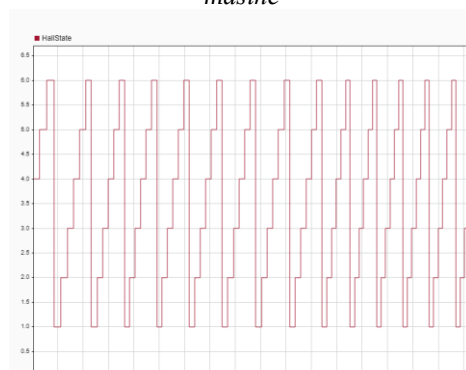
Slika 5.4. (a) Vremenski odziv brzine električne mašine u poređenju sa referentnom (zadatim) brzinom



Slika 5.4. (b) Vremenski odziv struja električne mašine



Slika 5.4. (c) Vremenska promena napona električnog mašine



Slika 5.4. (d) Hall signal za sinhronizaciju komutacije

6. ZAKLJUČAK

Na osnovu datih rezultata vremenskih odziva može se potvrditi ispravnost osmišljenog algoritma upravljanja, sposobnost brzog reagovanja na zadatu brzinu i obrtni moment opterećenja. Dobijeni rezultati računarske simulacije potvrđuju teorijska razmatranja koja su prethodno data u radu, te se može zaključiti da BLDC mašina specijalne konstrukcije može biti potencijalni izbor električnog pogonskog podsistema električnog vozila.

7. LITERATURA

- [1] "IEA Global EV stock by region, 2010 - 2020". URL: <https://www.iea.org/reports/global-ev-outlook-2021?mode=overview>
- [2] "Modern Electric Hybrid Electric and Fuel Cell Vehicle" – Mehrdad Ehsani
- [3] MATLAB model "Six-step commutation of BLDC motor using sensor feedback". URL: <https://www.mathworks.com/help/mcb/gs/six-step-commutation-blcdc-motor-using-position-sensor.html>
- [4] "EPA Dynamometer Drive Schedules". URL: <https://www.epa.gov/vehicle-and-fuel-emissions-testing/dynamometer-drive-schedules>

Kratka biografija:



Nevena Pantić rođena je 1997. godine u Vlasenici, BiH. Osnovne akademske studije je završila na Fakultetu tehničkih nauka 2016. godine, na smeru: Energetika, elektronika i telekomunikacije.

**NUMERIČKO MODELOVANJE RASPODELE ELEKTRIČNOG POLJA I
POTENCIJALA U IZOLACIONOM SLOJU VN KABLOVA U PRISUSTVU PUKOTINA**
**NUMERICAL MODELING OF ELECTRIC FIELD AND POTENTIAL DISTRIBUTION
IN WATER TREED INSULATION LAYER OF HV POWER CABLES**

Aleksandra Višnjić, Karolina Kasaš-Lažetić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu je izvršena analiza uticaja nepravilnosti u strukturi čvrstog dielektrika visokonaponskog kablova, u vidu šupljina, nečistoća, kao i pojave električnog i vodenog grananja (eng. treeing), na raspodelu električnog polja i električnog skalar potencijala unutar kablova. Svi proračuni su izvršeni pomoću COMSOL Multiphysics softverskog paketa, koji se bazira na metodi konačnih elemenata. Za potrebe analize, korišćen je model jednožilnog kablovskog voda 64/110 kV tip A2XS (FL)2Y (stara oznaka XHE 49-A). Dobijeni rezultati su pokazali da pojava vodenog grananja dovodi do značajnih promena u raspodeli električnog polja i električnog skalar potencijala unutar kablova. Uočeno je drastično povećanje električnog polja na vrhovima grana koje predstavlja preduslov za dalju degradaciju izolacije, što u krajnjoj instanci dovodi do proboja.

Ključne reči: VN kablovski vod, vektor jačine električnog polja, električni skalar potencijal, vodenog i električno grananje

Abstract – This paper analyses the impact of irregularities in a solid dielectric of the high-voltage cable in the form of voids and impurities. It also analyses how the electric and water treeing appearance affects the distribution of the electric field and electric scalar potential inside the cable. All calculations were performed applying the COMSOL Multiphysics software package based on finite element method. For analysis purposes, a model of a single-core cable line 64/110 kV type A2XS(FL)2Y (old identification XHE 49-A) was applied. The obtained results showed that the occurrence of water treeing leads to significant changes in the distribution of electric field and electric scalar potential inside the cable. A drastic increase in the electric field at the tips of the branches was observed, which is a prerequisite for further degradation of the insulation, which ultimately leads to breakdown.

Keywords: HV cable, electric field strength vector, electric scalar potential, water and electric treeing

1. UVOD

Savremeni razvoj industrije i privrede, uz osvajanje novih tehnologija, doveli su do naglog povećanja potrebe za

električnom energijom. Iz tog razloga došlo je do ekspanzije u konstrukciji svih delova elektroenergetskog sistema. Kablovi, kao jedan od osnovnih elemenata elektroenergetskog sistema, zauzimaju značajno mesto u distribuciji, a sve više i u prenosu električne energije.

U poslednjih nekoliko decenija je naročito zabeležen rastući trend primene visokonaponskih kablova, čime su otvorena razna pitanja u pogledu unapređenja i razvoja novih konstrukcijskih rešenja, materijala, kao i efikasnijih metoda za potrebe održavanja i monitoringa kablovskih vodova. Posebna pažnja prilikom projektovanja kablovskih vodova se pridaje dimezionisanju izolacionog sistema, kao elementa koji je izložen najvećem električnom naprezanju i koji dominantno određuje faktor pouzdanosti rada. Na nivou današnjeg tehnološkog razvoja, najveću primenu imaju kablovi sa sintetičkom izolacijom, pri čemu je umreženi polietilen (XLPE) potisnuo sve druge materijale. Međutim, pored dobrih električnih, mehaničkih i termičkih karakteristika, koje su ga izdvojile od ostalih materijala, glavni nedostatak izolacije od umreženog polietilena ogleda se u njegovoj osetljivosti na parcijalna pražnjenja i efekte električnog i vodenog grananja (eng. treeing) [1].

U osnovi, materijali koji se koriste za izradu izolacije, u svojoj strukturi nisu idealni, već sadrže brojne defekte u vidu uključaka, mikrošupljina i raznih drugih zaprljanja. Ova nehomogenost izolacije predstavlja rezultat karakteristika upotrebljenog materijala, konstrukcije kablova, primenjene tehnologije izrade, kao i mehaničkih naprezanja kojima je izolacija izložena tokom polaganja i eksploatacije kablova. Zbog svega navedenog, cilj ovog rada jeste da se ispita uticaj nepravilnosti u strukturi čvrstog dielektrika, u vidu šupljina, nečistoća, kao i pojave električnog i vodenog grananja, na raspodelu električnog polja i električnog skalar potencijala unutar kablovskog voda.

Svi proračuni u ovom radu su izvršeni pomoću COMSOL Multiphysics softverskog paketa koji se bazira na metodi konačnih elemenata. Ova metoda omogućava približno rešavanje parcijalnih diferencijalnih jednačina na osnovu zadatih početnih i graničnih uslova, a kao takva se pokazala pogodnom za proučavanje složenih elektromagnetskih pojava koje su opisane Maksvelovim jednačinama u diferencijalnom obliku.

2. TEORIJSKA OSNOVA

Elektromagnetska polja koja stvaraju kablovski vodovi su mrežne učestanosti (na ovim prostorima ona iznosi 50 Hz) i svrstavaju se u vremenski sporopromenljiva polja. Na osnovu toga, analize magnetskih i električnih efekata

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila vanr. prof. dr Karolina Kasaš-Lažetić.

jednog te istog elektromagnetskog polja mogu se sprovesti zasebno, nezavisnim posmatranjem vremenski sporopromenljivog magnetskog polja i vremenski sporopromenljivog električnog polja [2].

2.1 Osnove proračuna

Za opisivanje električnog polja koristi se vektor jačine električnog polja $\vec{E}$, koji može da se odredi pomoću električnog skalar potencijala V i magnetskog vektor potencijala $\vec{A}$. Stoga, relacija kojom se definiše vektor jačine električnog polja glasi:

$$\vec{E} = -\nabla V - \frac{\partial \vec{A}}{\partial t}. \quad (1)$$

S obzirom da razmatrani kabl ne sadrži feromagnetske delove, usled linearnosti sistema i prostoperiodične pobude u provodniku kabla, analizu problema je moguće postaviti i izvršiti u kompleksnom domenu. U tom slučaju, kompleksni električni skalar potencijal $\underline{V}$ se može odrediti rešavanjem sledeće kompleksne diferencijalne jednačine, poznatije kao Poasonova diferencijalna jednačina:

$$\Delta \underline{V} = -\frac{\rho}{\epsilon}. \quad (2)$$

Kako se predmet analize ovog rada bazira na modelu koji uzima u obzir specifičnu provodnost sredine, uz zanemarivanje zapreminskog naelektrisanja unutar izolacije i efekata povezanih sa indukovanim električnim poljem, kompleksni električni skalar potencijal se može odrediti na osnovu Laplasove kompleksne diferencijalne jednačine:

$$\Delta \underline{V} = 0. \quad (3)$$

Prvi sabirak u izrazu za ukupan vektor jačine električnog polja (1) predstavlja njegovu konzervativnu komponentu, koja je u kompleksnom domenu, za posmatrani slučaj radijalna:

$$\vec{E}_{\text{konz}} = -\nabla V = E_r \vec{r}_r. \quad (4)$$

Drugi sabirak opisuje indukovanu komponentu, koja predstavlja vrtložno polje, a za posmatrani model kablovskog voda je u pravcu z koordinate:

$$\vec{E}_i = -j\omega \vec{A} = E_{iz} \vec{z} = -j\omega \underline{A}_z \vec{z}. \quad (5)$$

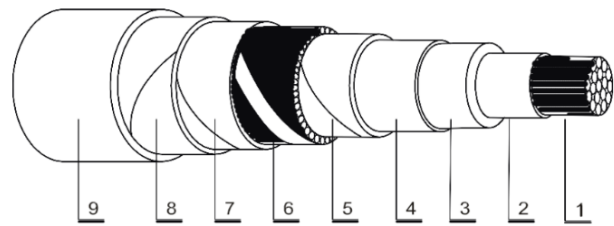
Unutar provodnih slojeva kablovskog voda dominantna je indukovana komponenta vektora jačine električnog polja, koja izaziva pojavu vrtložnih struja, kao i manje ili više izražen površinski efekat. Sa druge strane, indukovanu komponentu vektora jačine električnog polja je moguće zanemariti izvan provodnih delova kabla, pri čemu se u razmatranje uzima samo radijalna, konzervativna komponenta vektora jačine električnog polja.

3. MODEL

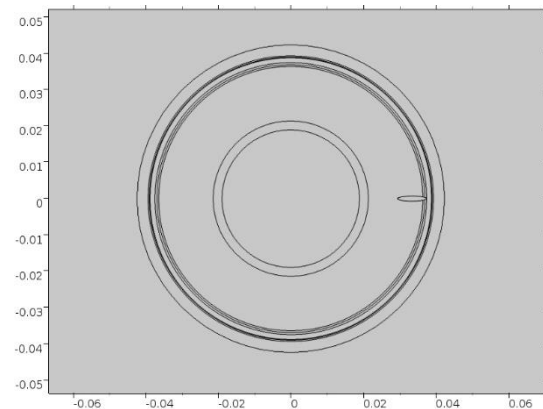
Za potrebe analize raspodele električnog polja i potencijala u ovom radu je korišćen model jednožilnog kablovskog voda 64/110 kV tip A2XS(FL)2Y, čija je stara oznaka XHE 49-A, sa provodnikom načinjenog od aluminijuma poprečnog preseka 1000 mm² i električnom zaštitom od bakra poprečnog preseka 95 mm², koji se po pravilu koristi u prenosnoj mreži Srbije nazivnog napona ≥ 110 kV (slika 1) [3].

Poprečni presek modelovanog kablovskog voda sa vodenim grananjem na ekranu izolacije, prikazan je na

slici 2, dok su njegove osnovne konstruktivne i elektromagnetske karakteristike date u tabeli 1.



Slika 1. Skica konstrukcije jednožilnog kabla 64/110 kV tip A2XS(FL)2Y



Slika 1. Poprečni presek modelovanog kablovskog voda u COMSOL-u

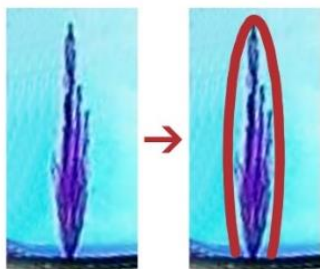
Tabela 1. Konstruktivne i elektromagnetske karakteristike modelovanog kabla (bez pukotina)

Br.	Konstruktivni elementi	Materijal	Prečnik [mm]	ϵ_r	σ [S/m]
1	Provodnik	Aluminijum	37,9	1	$37,74 \cdot 10^6$
2	Ekran provodnika	Trostruko ekstrudovani poluprovodni umreženi polietilen	42,4	2,3	10^{-15}
3	Izolacija	Umreženi polietilen	72,8	2,3	10^{-15}
4	Ekran izolacije	Trostruko ekstrudovani poluprovodni umreženi polietilen	73,8	2,3	10^{-15}
5	Separator	Poluprovodna bubreća traka	74,9	2,5	10^{-15}
6	Električna zaštita	Bakar	77,6	1	$59,98 \cdot 10^6$
7	Separator	Poluprovodna bubreća traka	78,0	2,5	10^{-15}
8	Laminirani plašt	Aluminijum	78,7	1	$37,74 \cdot 10^6$
9	Spoljašnji plašt	Polietilen visoke gustine	84,7	2,4	10^{-15}

Posebna pažnja u ovom radu posvećena je pojavi vodenog grananja, koja se ogleda kroz degradaciju sintetičke izolacije usled jednovremenog dejstva vode i električnog polja. Prema mestu nastanka pukotine u izolaciji, izvršena je osnovna podela na vodenog grananje koje se javlja na ekranu provodnika ili izolacije, kao i na ono koje se razvija iz nepravilnosti u unutrašnjim slojevima izolacije [4].

Kako je praksa pokazala, najopasnijom vrstom vodenog grananja se smatra slučaj njegovog nastanka na ekranu

izolacije. U skladu sa tim, osnovni cilj ovog rada je usmeren na određivanje uticaja ove vrste vodenog grananja na raspodelu električnog polja i potencijala unutar kabla. Za potrebe modelovanja korišćena je uprošćena geometrija prikazana na slici 3.



Slika 2. Pojednostavljeni oblik vodenog grananja na ekranu izolacije

Za potrebe definisanja relativne permitivnosti vodenog grananja iskorišćen je rezultat studije [5] koja je pokazala da je maksimalna vrednost relativne permitivnosti oblasti vodenog grananja tri puta veća od relativne permitivnosti izolacije. Sa druge strane, za potrebe definisanja specifične provodnosti vodenog grananja iskorišćen je rezultat studije [6] kojom je pokazano da je maksimalna vrednost specifične provodnosti regije vodenog grananja 10^{10} puta veća od specifične provodnosti izolacije. U ovom radu, za potrebe modelovanja vodenog grananja, usvojeno je linearno smanjenje relativne permitivnosti i specifične provodnosti, od vrha vodenog grananja do njegovog korena uz separatorski sloj, koje su definisane na sledeći način:

$$\epsilon_r = (\epsilon_{\max} - \epsilon_{\min}) \cdot f_1(x) \cdot f_2(y) + \epsilon_{\min}, \quad (6)$$

$$\sigma = (\sigma_{\max} - \sigma_{\min}) \cdot f_1(x) \cdot f_2(y) + \sigma_{\min}, \quad (7)$$

gde su,

$\epsilon_{\min}$ – relativna permitivnost izolacije,

$\epsilon_{\max} = 3 \cdot \epsilon_{\min}$,

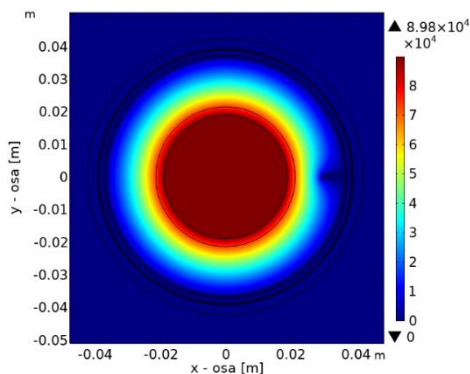
$\sigma_{\min}$ – specifična provodnost izolacije,

$\sigma_{\max} = 10^{10} \cdot \sigma_{\min}$,

$f_1(x), f_2(y)$ – linearne funkcije po x i y osi.

4. REZULTATI

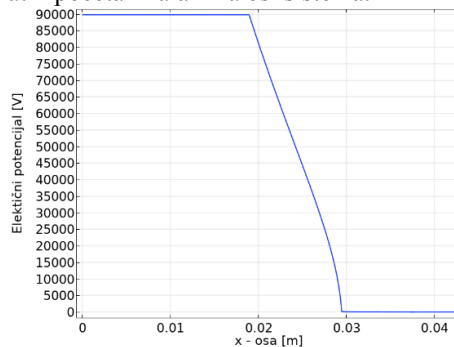
Na slici 4 je prikazana prostorna raspodela električnog skalar potencijala unutar kablovskog voda, na kojoj se jasno vidi izobličenje u okolini vodenog grananja.



Slika 3. Prostorna raspodela električnog skalar potencijala

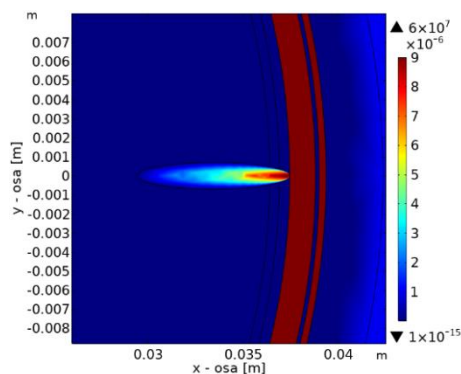
U svrhu analize promene električnog skalar potencijala u oblasti vodenog grananja, posmatrana je njegova promena

po x osi (slika 5), duž pravca $y = 0$, pri čemu se koordinatni početak nalazi na osi sistema.



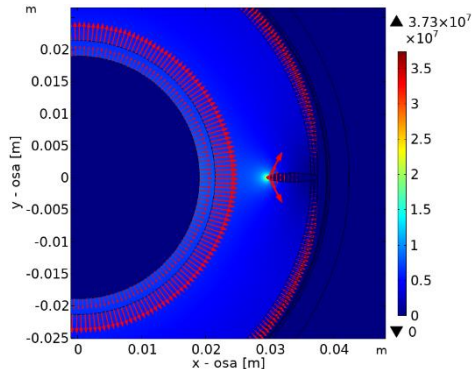
Slika 4. Promena električnog potencijala po x – osi

Kao što se može videti, unutar provodnih slojeva električni skalar potencijal ima konstantnu vrednost. Očekivano, najveća vrednost električnog skalar potencijala je unutar provodnika i ona iznosi $V_{\max} = 89,8$ kV. Sa slike 5 se vidi da je, unutar regije vodenog grananja (za $y = 0 \wedge \forall x \in [x_1, x_2]$, gde su $x_1 = 0,02985$ m i $x_2 = 0,03745$ m) električni skalar potencijal približno jednak nuli. Ovo je posledica povećane specifične provodnosti regije vodenog grananja, prikazane na slici 6, kao i njenog kontakta sa električnom zaštitom koja se nalazi na nultom potencijalu.



Slika 5. Prostorna raspodela specifične provodnosti regije vodenog grananja

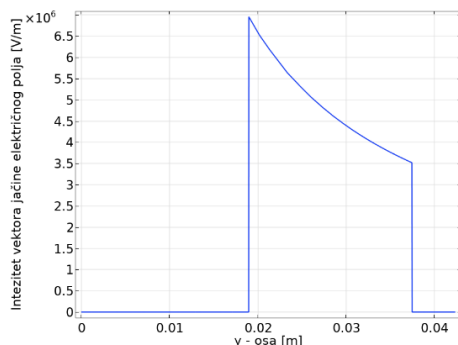
Na slici 7 je prikazana prostorna raspodela električnog polja, sa koje se može videti da linije vektora jačine električnog polja skreću pod pravim uglom ka oblasti vodenog grananja usled njene povećane specifične provodnosti u odnosu na izolaciju.



Slika 6. Prostorna raspodela električnog polja u okolini vodenog grananja

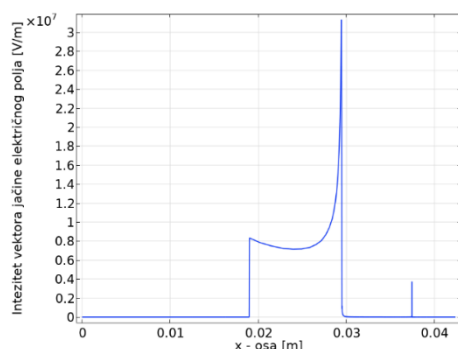
Promena inteziteta vektora jačine električnog po y – osi je prikazana na slici 8 i njom je predstavljena situacija bez

nepravilnosti u izolaciji. Kao što je očekivano, intezitet vektora jačine električnog polja unutar provodnih slojeva je približno jednak nuli. U izolatorskim slojevima, promena jačine električnog polja je obrnuto srazmerna rastojanju od ose sistema, pri čemu je intenzitet vektora jačine električnog polja najveći na unutrašnjem sloju izolacije uz provodnik, a najmanji na spoljašnjem sloju. U neposrednoj blizini provodnika, intezitet vektora jačine električnog polja se skokovito menja i dostiže maksimalnu vrednost, koja iznosi $E_{\max} = 6,98 \text{ MV/m}$. Analogno principu Faradejevog kaveza, električnom zaštitom je električno polje ograničeno samo na unutrašnjost kabla, što se takođe vidi na slici 8.



Slika 7. Promena intenziteta vektora jačine električnog polja po y – osi

U svrhu analize promene intenziteta vektora jačine električnog polja u oblasti vodenog grananja, posmatrana je njegova promena po x – osi (slika 9).



Slika 8. Promena intenziteta vektora jačine električnog polja po x – osi

Odmah može da se uoči značajna razlika u raspodeli intenziteta vektora jačine električnog polja u odnosu na pravac bez oblasti vodenog grananja (slika 8). Usled povećanja specifične provodnosti u oblasti vodenog grananja, intezitet vektora jačine električnog polja u toj oblasti je približno jednak nuli. Takođe, povećanje relativne permitivnosti vodenog grananja doprinosi dodatnom smanjenju električnog polja unutar posmatrane oblasti, pošto je jačina električnog polja obrnuto srazmerna relativnoj permitivnosti sredine.

Ipak, najbitnije zapažanje odnosi se na drastično povećanje električnog polja na vrhu vodenog grananja, koje dostiže vrednost $E_{\max} = 37,3 \text{ MV/m}$ i nešto manji skok na korenu vodenog grananja uz separator. Ovo se objašnjava činjenicom da je električno polje u okolini provodnih tela obrnuto srazmerno poluprečniku zakrivljenosti površine tela - vrh vodenog grananja ponaša se kao šiljasti deo provodnika.

5. ZAKLJUČAK

Za uspostavljanje pouzdanog i sigurnog napajanja električnom energijom, od suštinske važnosti je stanje, kao i redovno i pravilno održavanje svih elemenata elektroenergetskog sistema. Održavanje i monitoring kablovskih vodova predstavlja skup složenih metoda merenja i ispitivanja u cilju održavanja kablovskih vodova u tehnički ispravnom stanju. Posebnu pažnju iziskuje analiza stanja izolacionog sistema. U ovom radu je pokazano da pojava vodenog grananja na ekranu izolacije dovodi do značajnih promena u raspodeli električnog polja i električnog skalar potencijala unutar kabla. Najdrastičnije povećanje električnog polja je uočeno na njegovom vrhu, gde dostiže vrednost od $37,3 \text{ MV/m}$. Kako dielektrična čvrstina umreženog polietilena iznosi oko 21 MV/m , očigledno je da će pomenuto povećanje električnog polja na vrhovima grana prouzrokovati degradaciju izolacije, a samim tim i brže širenje vodenog grananja. Pokazano je, takođe, da i pored pojave vodenog grananja, maksimalna vrednost električnog skalar potencijala ostaje nepromenjena, pa samim tim, nema ni devijacije napona koja bi ukazala na postojanje vodenog grananja.

6. LITERATURA

- [1] Stojan V. Nikolajević, *Kablovska tehnika*, JP Službeni list SRJ, Beograd, 2007.
- [2] K. Kasaš-Lažetić, D. Herceg, G. Mijatović, D. Kljajić, M. Prša, *Electromagnetic Field Produced by Single HV Cable*, INDEL, 2018, Banja Luka, pp. 1-5.
- [3] IS-EMS 200:2019 *Osnovni tehnički zahtevi za izbor i montažu energetskih kablova i kablovskog pribora u prenosnoj mreži*
- [4] E. F. Steennis, *Water treeing – the behaviour of water trees in extruded cable insulation*, 1989.
- [5] T. Toyoda, S. Mukai, Y. Ohki, Y. Li, T. Maeno, *Estimation of conductivity and permittivity of water trees in PE from space charge distribution measurements*, IEEE Transaction on Dielectrics and Electrical Insulation, 2001, pp. 111-116
- [6] M. Acedo, I. Radu, F. Frutos, J.C. Filippini, P. Nottingher, *Water treeing in underground power cables: modelling of the trees and calculation of the electric field perturbation*, Journal of Electrostatics, 2001, pp. 267-294



Aleksandra Višnjic, rođena je 1995. godine u Štutgartu (Nemačka). Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnika i računarstvo – Energetska elektronika i električne mašine je odbranila 2022. godine.

Kontakt:

aleksandra.g.visnjic@gmail.com



Karolina Kasaš-Lažetić rođena je u Novom Sadu. Doktorirala je na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva.

Oblast interesovanja je teorijska i primenjena elektromagnetika.

PRORAČUN FLUKSEVA SINHRONIH MAŠINA SA STALNIM MAGNETIMA

CALCULATION OF FLUX LINKAGES OF PERMANENT MAGNET SYNCHRONOUS MACHINES

Tamara Milovac, Dejan Jerkan, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu su date osnove metode konačnih elemenata, koja je primenjena za izračunavanje fluksnih obuhvata sinhronih mašina sa stalnim magnetima (SMSM). Na primeru sinhronog motora sa utisnutim stalnim magnetima, izvršen je proračun fluksnih obuhvata, a zatim su prikazani i analizirani njihovi talasni oblici usled pobuđenosti statorskkih namotaja, kao i onih koji su posledica pobuđenosti mašine stalnim magnetima.

Ključne reči: SMSM, FEA, fluksni obuhvat, zasićenje.

Abstract – The paper presents the basics of the finite element method, which was applied to calculate the flux linkages of synchronous machines with permanent magnets (PMSM). On the example of a synchronous motor with interior permanent magnets, the calculation of flux linkage were performed, and then waveform of flux linkage caused by the permanent magnet and waveform of flux linkage caused by armature stator currents were shown.

Keywords: PMSM, FEA, flux linkage, saturation.

1. UVOD

Sinhronne mašine su rotacione električne mašine koje karakteriše obrtno magnetno polje u vazдушnom zazoru, čija sinhrona brzina obrtanja zavisi od frekvencije struja statorskkih namotaja. Pojam sinhrona proizilazi iz činjenice da je mehanička brzina obrtanja rotora jednaka brzini obrtnog polja.

Kako bi se ostvarila elektromehanička konverzija uz sinhronu brzinu obrtanja rotora, potrebno je obezbediti da rotorsko magnetno kolo ima sopstveni izvor pobuđivanja. Pobuđivanje se može realizovati pomoću zasebnog pobudnog namotaja kroz koji se uspostavlja jednosmerna struja, ili postavljanjem stalnih magneta na rotoru. Tema ovog rada je proračun flukseva sinhronih mašina sa stalnim magnetima kao izvorom pobude, tako da će na dalje biti reči samo o ovoj vrsti mašina.

Sinhronne mašine sa stalnim magnetima imaju visoku efikasnost, visoku stabilnost obrtnog momenta, malu inerciju i jednostavno upravljanje. Ovaj tip mašina najčešće služi za pogone koji rade u širokom opsegu brzina. Brzine se kreću u opsegu od nekoliko stotina, pa čak i do preko deset hiljada obrtaja u minuti.

Neke od najčešćih primena SMSM su: robotski aktuatori, alatne mašine, kućni aparati i električna vozila.

NAPOMENA:

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

U radu je izvršen proračun fluksnih obuhvata SMSM, zatim su prikazani i analizirani njihovi talasni oblici po d i q osi, usled pobuđenosti statorskkih namotaja, kao i onih koji su posledica pobuđenosti mašine stalnim magnetima. Za izvršene analize korišćena je softverska implementacija metode konačnih elemenata.

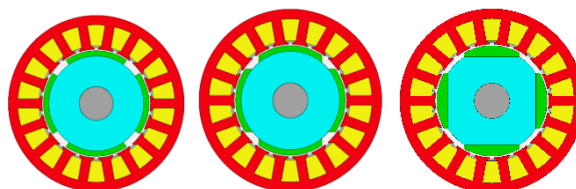
2. TIPIČNE KONSTRUKCIJE SMSM

U zavisnosti od načina postavljanja stalnih magneta na rotoru, SMSM se mogu podeliti na sinhronne mašine sa površinski postavljenim i na one sa magnetima utisnutim u rotor.

Kod sinhronih mašina sa površinski postavljenim magnetima, magneti su pričvršćeni na obod rotora. Različite magnetne provodnosti u dve električno ortogonalne ose dovode do magnetne asimetrije. Te dve ose se nazivaju podužna d osa i poprečna q osa. Zato što nema izraženog efekta isturenosti polova podužna i poprečna induktivnost, L_d i L_q su približno jednake.

Prednost konstrukcije sinhronne mašine sa površinski postavljenim magnetima je to što im je jednostavna montaža, a nedostatak je to što je magnet više izložen demagnetisućem polju i centrifugalnim silama. Takođe, ovakve mašine nisu pogodne za pogone kod kojih se zahtevaju velike brzine obrtanja.

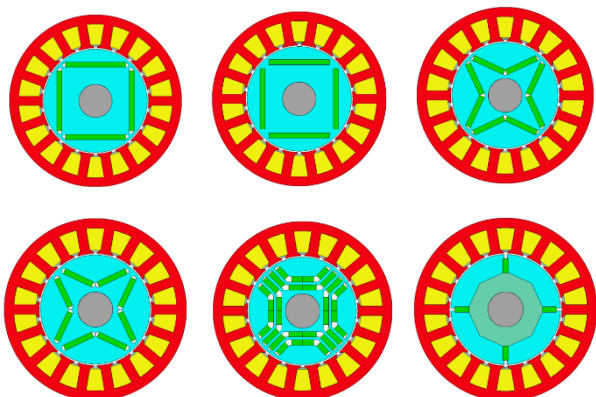
Na slici 1 su prikazani primeri topologije sinhronih motora sa površinski postavljenim stalnim magnetima na rotoru. Svaku od topologija odlikuje različit oblik stalnih magneta, što utiče na način njihovog fiksiranja za rotor, odnosno cenu njihove izrade.



Slika 1. Primeri topologija sa površinski postavljenim magnetima

Kod sinhronih mašina sa utisnutim stalnim magnetima magneti su dobro oklopljeni i zaštićeni, zbog čega su pogodni kod pogona koji zahtevaju veliki raspon brzina. Rotor ovako konstruisanih SMSM ima magnetnu anizotropiju zbog kompleksne geometrije. Za razliku od sinhronih mašina sa površinski postavljenim magnetima, ove mašine zbog manjeg vazдушnog zazora imaju izraženiju reakciju armature.

Na slici 2 prikazane su neke od topologija mašina sa utisnutim stalnim magnetima na rotoru. Za izradu rotora je potrebna znatno manja količina gvožđa u d osi, zbog čega je induktivnost L_d mnogo manja od induktivnosti L_q . Takođe, postoji zavisnost induktivnosti statora od ugla rotora što dovodi do pojave reluktantnog momenta, koji predstavlja dodatnu komponentu koja se superponira na osnovni momenat, omogućavajući ovakvom tipu mašina veću gustinu momenta po jedinici mase.



Slika 2. Primeri topologija sa utisnutim stalnim magnetima

Uloga statorskih namotaja je da u vazдушnom zazoru mašine osigura magnetno polje. Topologija statora kod oba navedena tipa mašina podrazumeva dve osnovne vrste namotaja: koncentrični namotaj i raspodeljeni namotaj.

Koncentrični namotaj je manji, jednostavniji i ima manje gubitke u bakru namotaja. Kod koncentričnog namotaja provodnici su smešteni oko istaknutog pola u jedan veliki žleb. Ovakav namotaj daje maksimalni iznos polja koje nije sinusnog karaktera usled čega se javljaju viši harmonici.

Kod raspodeljenog namotaja provodnici su raspoređeni u više žlebova ispod jednog pola. Imaju bolju sinusnu raspodelu polja u vazдушnom zazoru usled čega imaju manji uticaj viših harmonika. Za raspodeljeni namotaj potrebna je veća količina bakra, zbog čega su gubici u bakru veći nego kod koncentričnog namotaja.

3. MODELOVANJE ZASIĆENE SINHRONE MAŠINE SA UTISNUTIM STALNIM MAGNETIMA

Matematičko modelovanje je od izuzetnog značaja za izučavanje različitih pojava i procesa koji se odvijaju, kako u ustaljenom režimu rada, tako i tokom prelaznih procesa u električnim mašinama, u ovom poglavlju se pristupa izvođenju jednačina bitnih za dobijanje fluksnih obuhvata po d i q osi. Takođe, će biti reči i o metodi konačnih elemenata, kao i o metodi zamrznutih permeabilnosti koja omogućava razdvajanje vrednosti fluksnih obuhvata koji potiču od dejstva stalnih magneta i dejstva armaturnih struja.

Metoda konačnih elemenata je grafo-numerička metoda koja predstavlja izuzetno moćno sredstvo za izračunavanje fenomena u elektromagnetnom polju koje se uspostavlja u sredinama složene geometrije. Rotacione električne mašine upravo predstavljaju takve uređaje, pa je primena ove metode u analizi njihovog ponašanja sve

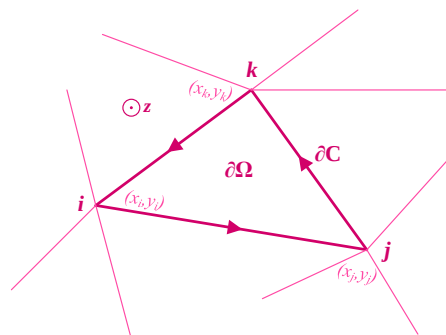
zastupljenija. Postoje različite implementacije ove metode, u zavisnosti od problema koji se želi istražiti, a najzastupljenija je ona koja se temelji na rešavanju parcijalnih diferencijalnih jednačina magnetskog vektor potencijala $\mathbf{A}$.

U zavisnosti od nivoa detaljnosti koji se želi postići, postoje dvodimenzionalne i trodimenzionalne implementacije ove metode. Iskustvo pokazuje da je za širok spektar problema najčešće dovoljno formulisati dvodimenzionalni problem metode konačnih elemenata, naročito za probleme koji nemaju izraženu varijaciju geometrije duž treće dimenzije (u slučaju rotacionih mašina treća dimenzija predstavlja osnu dužinu, kojom se definiše dužina paketa limova). U tom slučaju domen od interesa predstavlja planarnu površ poprečnog preseka magnetnog kola mašine, a konačni elementi na koje se poprečni presek izdeljuje postaju trougaoni.

Dvodimenzionalna formulacija metode konačnih elemenata u kojoj je planarna površ poprečnog preseka mašine postavljena u xy ravni prerasta u parcijalnu diferencijalnu jednačinu u kojoj magnetski vektor potencijal $\mathbf{A}$ ima samo z komponentu. Relacija koja se može definisati na nivou pojedinačnog konačnog elementa tada glasi:

$$\frac{1}{\mu} \left(\frac{\partial^2 A_z}{\partial x^2} + \frac{\partial^2 A_z}{\partial y^2} \right) = -J_{0izv} + \sigma \left(\frac{\partial A_z}{\partial t} + \text{grad}_z V \right) \quad (1)$$

U relaciji (1) $\mu \left[\frac{H}{m} \right]$ predstavlja permeabilnost, a $\sigma \left[\frac{S}{m} \right]$ predstavlja provodnost sredine kojoj konačni element pripada. Član $J_{0izv} \left[\frac{A}{m^2} \right]$ predstavlja gustinu struje strujnih izvora (ukoliko takvih izvora ima u samom konačnom elementu), $V[V]$ je električni skalar potencijal, dok $A_z \left[\frac{W_b}{m} \right]$ predstavlja pomenutu z komponentu magnetskog vektor potencijala. Na slici 3 je prikazan detalj jednog konačnog elementa trougaonog oblika.



Slika 3. Detalj konačnog elementa trougaonog oblika

Relacija 1 se nizom matematičkih transformacija prevodi u integralnu jednačinu u kojoj se oblik rešenja forsira unapred definisanim funkcijama oblika trougaonih segmenata. Za slučaj primene metode u magnetno linearnim sredinama sistem jednačina koje je potrebno formulisati za sve konačne elemente se može značajno brže rešiti nego u slučaju nelinearnih sredina, kada i sama permeabilnost koja u jednačinama figuriše zavisi od intenziteta polja u čijem rešavanju učestvuje kao promenljivi parametar. Otuda se metoda konačnih elemenata u primeni rešavanja elektromagnetnih polja u električnim mašinama smatra daleko najzahtevnijom metodom sa aspekta računarskih resursa, ali i utroška

vremena, budući da se praktično sve rotacione mašine izrađuju od feromagnetskih materijala, koji su izuzetno magnetno nelinearni.

Magnetna polja stalnih magneta i armaturnih struja deluju u nelinearnoj sredini i međusobno utiču jedno na drugo. Razdvajanje vrednosti fluksnih obuhvata koji potiču od dejstva stalnih magneta i dejstva armaturnih struja moguće je uz primenu metode zamrznutih permeabilnosti na osnovu koje se vrše tri FEM simulacije. Rezultat ovih simulacija omogućuje izračunavanje parametara u jednačinama u svim radnim tačkama (i_d, i_q) sa velikom tačnošću.

Kako bi se razdvojio uticaj pobudnih struja i stalnih magneta na ukupan fluksni obuhvat statorskih namotaja, potrebno je primeniti sledeći postupak. Najpre se izvrši simulacija na kompletnom modelu mašine, koja je istovremeno pobuđena i stalnim magnetima, ali i odgovarajućim statorskim strujama. Rezultati ovakve simulacije služe isključivo kako bi se dobila odgovarajuća slika zasićenosti u mašini, odnosno lokalne vrednosti permeabilnosti u svakom od konačnih elemenata. Zatim se izvrši „zamrzavanje“ ovih vrednosti permeabilnosti, tako da se suštinski dobije mašina sa linearnim feromagnetskim materijalom, ali takvim da je on izuzetno nehomogen, jer mu vrednost permeabilnosti na lokalnom nivou varira.

Na ovako pripremljenom modelu zamrznutih permeabilnosti se zatim vrše simulacije gde se mašina pobuđuje isključivo statorskim strujama ili stalnim magnetima. Tako se postiže efikasno razdvajanje njihovih doprinosa ukupnim fluksnim obuhvatima, što se zatim može iskoristiti za proračun induktivnosti.

Algoritmi zasnovani na primeni metode konačnih elemenata omogućuju numerički proračun magnetnog vektor potencijala $\vec{A}$ u svim tačkama mreže konačnih elemenata na koju je izdelfen poprečni presek mašine. Na osnovu tih proračuna moguće je rekonstruisati magnetno polje u mašini i izračunati veličine od interesa.

Kao rezultat numeričkih proračuna dobijaju se fluksni obuhvati u svakoj od faza trofaznog statorskog namotaja. Njih je potrebno transformisati u dq domen da bi se dobili fluksni obuhvati po d i q osama potrebni za izračunavanje L_d i L_q .

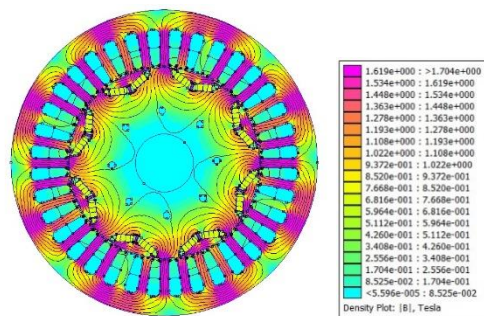
$$\psi = \psi_d + j\psi_q$$

$$= \frac{2}{3}(\psi_a + a\psi_b + a^2\psi_c), \quad a = e^{j\frac{2\pi}{3}} \quad (2)$$

$$\psi_d = \frac{2}{3}\left(\psi_a - \frac{1}{2}\psi_b - \frac{1}{2}\psi_c\right) \quad (3)$$

$$\psi_q = \frac{1}{\sqrt{3}}(\psi_b - \psi_c) \quad (4)$$

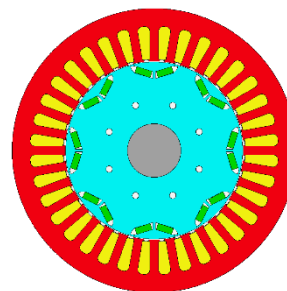
Nad modelom motora, kreiranim u programskom paketu *FEMM 4.2.* su zatim izvršene magnetostatičke simulacije. Kao odziv ovih simulacija se dobijaju vrednosti vektora magnetske indukcije u svakom od konačnih elemenata, koje se zatim koriste za proračun fluksnih obuhvata. Na slici 4 prikazan je odziv magnetostatičke simulacije.



Slika 4. Rezultati simulacije

4. PRAKTIČAN PRIMER

Za potrebe istraživanja i proračuna fluksnih obuhvata primenom metode konačnih elemenata i principa zamrznutih permeabilnosti u ovom radu je analizirana IPMSM čiji je poprečni presek prikazan na slici 5. Reč je o osmopolnoj sinhronoj mašini sa utisnutim stalnim magnetima u jednostrukom „V“ rasporedu. Statorski namotaj je izveden kao trofazni dvoslojni, dok je mašina predviđena za napajanje iz pretvarača koji na svom ulazu dobija nazivni jednosmerni napon od 48 V. Osnovni podaci o mašini dati su u tabeli 1.



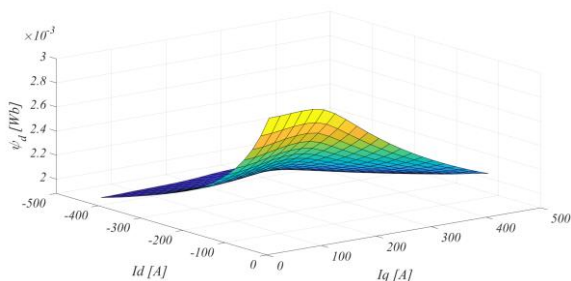
Slika 5. Izgled sinhronne mašine sa utisnutim stalnim magnetima na rotoru

Tabela 1. Osnovni podaci o razmatranoj mašini

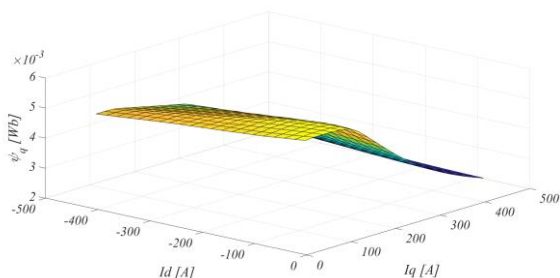
Osnovne karakteristike motora ACX-3544-01		
Karakteristika	Jedinica	Brojna vrednost
DC napon međukola	[V]	48
Maksimalna struja	[A]	510
Maksimalna brzina	$\frac{o}{min}$	6000
Kontrolna strategija	/	MTPA
Temperatura namotaja	[°C]	100
Temperatura za magnete	[°C]	100
Temperatura limova	[°C]	100

U nastavku će biti prikazani isključivo rezultati simulacija na modelima sa zamrznutim permeabilnostima, dok bi se fluksni obuhvati potpuno pobuđene mašine lako mogli dobiti sabiranjem, te su stoga opravdano izostavljeni.

Na osnovu relacije (3) se iz vrednosti fluksnih obuhvata po fazama a , b i c konačno može dobiti i fluksni obuhvat po d osi, a kao posledica pobuđenosti statorskih namotaja. Talasni oblik ovog fluksa u zavisnosti od struja po d i po q osi je prikazan na slici 6, dok je na slici 7 prikazan talasni oblik fluksnog obuhvata po q osi dobijen na osnovu relacije (4).



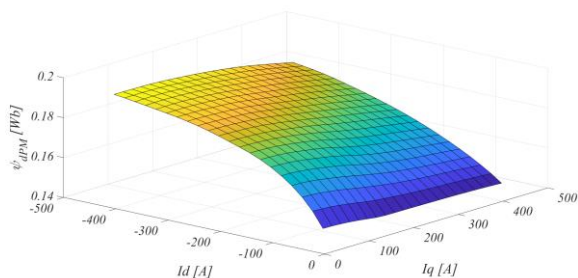
Slika 6 . Zavisnost fluksnog obuhvata u d osi od komponenta statorskih struja u d i q osama



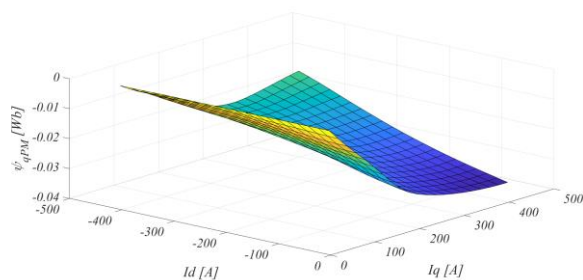
Slika 7 . Zavisnost fluksnog obuhvata u q osi od komponenta statorskih struja u d i q osama

Na osnovu slika 6 i 7 može se videti da zbog efekta zasićenja linearan porast vrednosti struja ne dovodi do linearne promene u vrednostima fluksnog obuhvata.

Efekat zasićenja magnetnog kola se još upečatljivije može prikazati ukoliko bi se izračunali i predstavili talasni oblici fluksnih obuhvata isključivo kao posledica pobuđenosti mašine stalnim magnetima. Po osnovnoj, linearnoj teoriji modelovanja električnih mašina, ovi fluksevi bi morali biti konstatni. Međutim, to u praksi svakako nije slučaj, a korišćena metoda zamrznutih permeabilnosti omogućava da se istraži promena fluksa stalnih magneta u zavisnosti od nivoa zasićenja mašine izazvanog združenim delovanjem svih pobuda.



Slika 8 . Zavisnost fluksnog obuhvata koji potiče od dejstva stalnih magneta u d osi od komponenta statorskih struja u d i q osama



Slika 9. Zavisnost fluksnog obuhvata koji potiče od dejstva stalnih magneta u q osi od komponenta statorskih struja u d i q osama

Na slikama 8. i 9. su prikazani talasni oblici fluksnih obuhvata po d i q osi u zavisnosti od komponenta statorskih struja po d i q osi, a kao posledica pobuđenosti mašine stalnim magnetima.

5. ZAKLJUČAK

U radu su prikazani odzivi fluksnih obuhvata u d i q osi, a kao posledica pobuđenosti statorskih namotaja i fluksnih obuhvata koji potiču od dejstva stalnih magneta. Primetan je uticaj nelinearnosti feromagnetskog materijala u oblastima sa strujama statorskih namotaja većih intenziteta, gde se i očekuje najveće zasićenje. Naročito je ovaj efekat zavisao od intenziteta struja po d osi. Dobijeni fluksni obuhvati se dalje mogu koristiti za unapređenje kontrolnih strategija upravljanja pogonima sa ovakvim tipom pogonskih mašina.

6. LITERATURA

- [1] N. Bianchi and S. Bolognani, „Magnetic Models of Saturated Interior Permanent Magnet Motors based on Finite Element Analysis“, Conf. Rec. IEEE Ind. Appl. Soc. Annu. Meeting, pp 27-34, 1998.
- [2] D. Jerkan, M. Gecić, D. Marčetić „IPMSM Inductances Calculation Using FEA“, Conf. Rec. INDEL 2014, pp 134-138, 2014.
- [3] Venco Ćorluka, „Modeliranje sinkronog motora s unutarnjim permanentnim magnetima i koncentriranim namotom“, doktorski rad, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, 2017.
- [4] Besplatni, nekomercijalni softver FEMM 4.2 dostupan na veb stranici www.femm.info.
- [5] S. J. Salon, Finite Element Analysis of Electrical Machines, Springer US, 1995, pp. 1–16.

Kratka biografija:

Tamara Milovac rođena je u Novom Sadu, 1997. godine. Diplomirala je na Fakultetu tehničkih nauka u Novom Sadu 2021. godine. Master rad iz oblasti Elektrotehnike i računarstva – Energetska elektronika i električne mašine odbranila je 2022. godine.

Dejan Jerkan je docent na Fakultetu tehničkih nauka u Novom Sadu, na Katedri za Energetsku elektroniku i pretvarače. Oblast interesovanja su mu modelovanje i dijagnostika električnih mašina, kao i metoda konačnih elemenata.

PREDIKCIJA OCENE APLIKACIJE NA OSNOVU KOMENTARA**PREDICTING APPLICATION RATINGS BASED ON USER REVIEWS**Marko Mijatović, *Fakultet tehničkih nauka, Novi Sad***Oblast – SOFTVERSKO INŽENJERSTVO I INFORMACIONE TEHNOLOGIJE**

Kratak sadržaj – U ovom radu predstavljena je specifikacija, implementacija i evaluacija sistema za predikciju ocene na osnovu tekstualnog komentara. Upoređena su dva pristupa – rekurentne neuronske mreže i modeli ansambla.

Ključne reči: mašinsko učenje, rekurentne neuronske mreže, modeli ansambla

Abstract – This paper presents a specification, implementation and evaluation of a system that predicts the rating of application based on textual comments. Two approaches were compared - recurrent neural networks and ensemble models.

Keywords: machine learning, recurrent neural networks, ensemble learning

1. UVOD

Recenzije korisnika sadrže informacije koje su korisne za analitičare i dizajnere aplikacija, kao što su zahtevi korisnika, greške, i iskustva prilikom korišćenja aplikacija. Kao povratne informacije, recenzije korisnika govore prodavcima kako da poboljšaju kvalitet softverskog proizvoda. Potencijalni korisnici aplikacija najčešće pregledaju recenzije pre nego što ih instaliraju. Dakle loše recenzije, negativno utiču na percipirani kvalitet aplikacije, kao i na popularnost, a samim tim i zaradu.

Problem kojim se rad bavi definisan je na sledeći način: na ulazu se nalazi recenzija korisnika u tekstualnoj formi, dok izlaz treba da bude prediktovana ocena (broj 1-5).

Predikcija ocene na osnovu tekstualne recenzije može se posmatrati i kao regresioni, ali i kao klasifikacioni problem. Izučavajući literaturu, i radove koji se bave rešavanjem sličnog problema, razmatrani su različiti pristupi i upoređivani su na osnovu postignutih rezultata.

U radu "Predicting star ratings based on annotated reviews of mobile apps.", Monet i Dagmar su upotreбили sentiment analizu u kombinaciji sa modelima linearne regresije [1]. Ovaj rad je istakao koliko je sentiment analiza od velikog značaja za rešavanje problema. Iako je sentiment analiza vrlo moćna tehnika u obradi prirodnog jezika, njena primena na nepotpune skupove podataka vrlo verovatno će dati lošije rezultate.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kovačević, red. prof.

Kao evaluaciona metrika za performanse primenjenih regresionih modela upotrebljena je RMSE (eng. Root Mean Square Error), što je vrlo dobar pokazatelj odstupanja prediktovanih vrednosti od pravih u problemu koji se rešava. Naime, drugi metodi evaluacije kao što su F1 mera i tačnost (eng. accuracy), koji se primenjuju kod klasifikacionih modela isto tretiraju i velike i male greške u predikciji modela. Nije isto da li je model prediktovao 2 za ocenu umesto 1, ili je prediktovao 5 umesto 1. Metrika evaluacije koja je upotrebljena u ovom radu, verodostojnije prikazuje odstupanje predikcija od stvarnih vrednosti, nego što to prikazuju metrike evaluacije korištene u drugim pristupima koji su razmatrani u nastavku.

S druge strane, Umer i dr. su u studiji "Predicting numeric ratings for Google apps using text features and ensemble learning." koristili modele ansambla [2]. Pretprocesiranje podataka u ovom radu kvalitetno je odrađeno, na interesantan način (metod je predstavljen u samom radu) uklonjeni su autlajeri (eng. outliers), koji značajno mogu da naruše performanse primenjenih algoritama. Kao i u prethodnoj studiji upotrebljena je sentiment analiza kako bi se unapredile performanse pri predikciji. Modeli ansambla koji su korišteni su Random Forest, i GBM (eng. Gradient Boost Machine). Postignuta je tačnost 70 %, što se može smatrati vrlo dobrim rezultatom. Jedan od pristupa rešavanja problema koji će se koristiti u ovom radu su modeli ansambla.

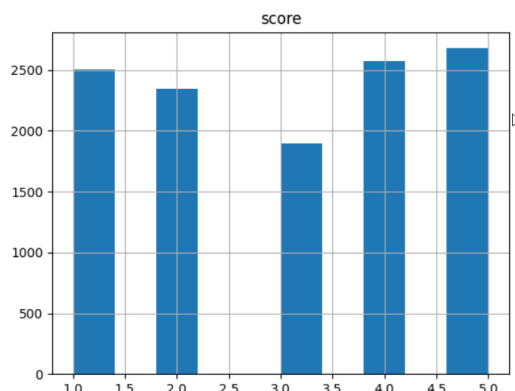
Gezici, Bahar i dr. su u radu "Neural sentiment analysis of user reviews to predict user ratings." predstavili model rekurentne neuronske mreže koji je pokazao najbolje rezultate [3]. Postignuta je tačnost 87 %, što je značajno bolje u odnosu na rezultate pristupa gde su korišteni modeli ansambla. Najverovatnije je da je za ovako dobre rezultate zadužen ogroman skup podataka koji je korišten. Metod upotrebljen u ovom radu (rekurentne neuronske mreže) vrlo je pogodan za prirodu problema koji se rešava, te iz tog razloga postignuti su bolji rezultati nego u drugim studijama koje su razmatrane. Stoga, primarni pristup rešavanju problema koji je upotrebljen u ovom radu jesu rekurentne neuronske mreže.

Kako je skup podataka koji se koristi znatno manji nego skup korišten u gore navedenoj studiji, kvalitetno pretprocesiranje podataka je vrlo važno kako bi se postigli dobri rezultati.

2. METOD

Najpre je opisan skup podataka koji se koristio za treniranje i testiranje obučanih modela mašinskog

učenja [4]. U skupu je 12496 recenzija. Ciljna varijabla je ocena, i posmatraće se balansiranost skupa podataka u odnosu na ciljnu varijablu. Na slici 1 prikazana je raspodela ciljne varijable – ocena korisnika.



Slika 1. Histogram ocena

Na osnovu slike 1, može se zaključiti da su podaci dobro balansirani. Raspodela ciljne varijable bi bila suprotnost u odnosu na Gausovu krivu. Prosečna ocena (3) je najmanje zastupljena, zatim imamo ocene (2) i (4), i najviše su zastupljene ocene (1) i (5). To donekle i ima smisla, jer najčešće recenzije ostavljaju ili oni korisnici koji su nezadovoljni aplikacijama, ili oni koji su oduševljeni.

2.1. Podela na trening/test skup

Skup podataka podeljen je na trening i test skup u razmeri 4:1. Za ovakvu raspodelu odlučeno je zbog toga što je skup podataka umerene veličine. Prilikom podele, očuvana je raspodela ciljne varijable u trening i test skup. Za optimizovanje hiperparametara korištena je unakrsna validacija tako da nije bilo potrebe za posebnim izdvajanjem validacionog skupa.

2.2. Uklanjanje autlajera

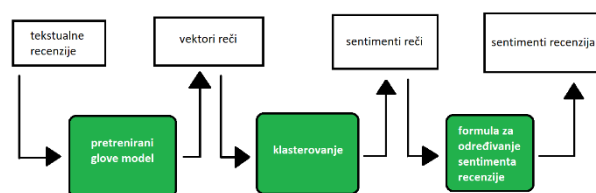
Recenzije koje se nalaze u korištenom skupu podataka većinski su napisane na engleskom jeziku, mada ima i onih koje su napisane na španskom, poljskom itd. Upotrebom *langdetect* biblioteke u programskom jeziku *Python*, pronađene su sve recenzije koje nisu napisane na engleskom jeziku i zatim su eliminisane. Autlajere je moguće ukloniti i vizualizacijom podataka, međutim u problemu koji se rešava tumačenje vizualizacije nema smisla. Autlajeri se moraju detektovati na neki drugi način. Ukoliko se sentiment recenzije ne poklapa sa ocenom koja je data (pozitivan sentiment i niska ocena, ili negativan sentiment i visoka ocena), onda takvu torqu smatramo autlajerom. Dakle, potrebno je da se uoče sve recenzije kod kojih se sentiment ne poklapa sa ocenom. Nažalost, u korištenom skupu podataka ne postoji obeležje za sentiment recenzije, tako da treba uraditi predikciju sentimenta nenadgledanim učenjem. Sentiment analiza nenadgledanim učenjem predstavlja složen zadatak. Rafal Wojcik predstavio je svoje rešenje problema, koje je prilagođeno u ovom radu. U nastavku se opisuje ova metoda.

Prvi korak bio je da se reči iz recenzija prevedu u vektore realnih brojeva, na taj način da semantički slične reči imaju slične vrednosti vektora. Ova tehnika obrade prirodnih jezika poznata je kao *word embedding*. Dva

dobro poznata *word embedding* algoritma su *word2vec* i *glove*. Kako je korpus reči svih recenzija u skupu podataka previše mali da bi se *word embedding* algoritam trenirao “od nule”, upotrebljen je pretrenirani *Glove* model [10]. Model je izabran jer je bio treniran na podacima sličnim skupu podataka koji se koristio.

Drugi korak sentiment analize bio je da se klasteruju reči u dve grupe - sa pozitivnim i negativnim sentimentom. Svako reči pridružen je *sentiment score*, broj koji je označavao izraženost pozitivnog odnosno negativnog sentimenta. *Sentiment score* se računao tako što se delio broj klastera kome reč pripada (+1 za pozitivan, -1 za klaster sa negativnim sentimentom) sa distancom od centroida klastera. Na taj način formiran je rečnik sentimenta recenzija - svako reči iz recenzija pridružen je sentiment score.

Krajnji korak bio je da se odredi sentiment recenzije na osnovu sentiment rečnika dobijenog u prethodnom koraku. Svaka reč u recenziji pretvorena je u vektor brojeva upotrebom TF-IDF (eng. *term frequency-inverse document frequency*) algoritma, slika 2. Sentiment recenzije računao se kao skalarni proizvod dva vektora – *tfidf* težina reči u recenziji i vektora sentiment score-ova za svaku reč u recenziji. Znak sentimenta označavao je da li je u pitanju pozitivan ili negativan sentiment. Korisno je bilo što je za svaku recenziju dobijena numerička vrednost koja je predstavljala izraženost sentimenta.



Slika 2. Koraci sentiment analize

2.3. Pretprocesiranje teksta

Koraci u obradi teksta:

- uklonjeni su znakovi interpunkcije
- sva slova pretvorena su u mala (eng. *lowercase*)
- tokenizacija
- uklanjanje stop reči (eng. *stop words*)
- lematizacija umesto stemminga, kako reči ne bi izgubile značenje

Neophodno je pretvoriti tekstualne recenzije u numeričku formu. *Word embedding* tehnike koje su korištene su TF-IDF i *Glove*. Mana *tf-idf* algoritma jeste što procesira jednu reč u trenutku, ne i okolinu reči tj. kontekst. Modeli ansambla trenirani su na recenzijama koje su pretprocesirane pomoću *tf-idf* algoritma, dok se kod rekurentne neuronske mreže u *embedding* sloju koristio pretrenirani *Glove* model.

2.4. Klasifikacija upotrebom modela ansambla

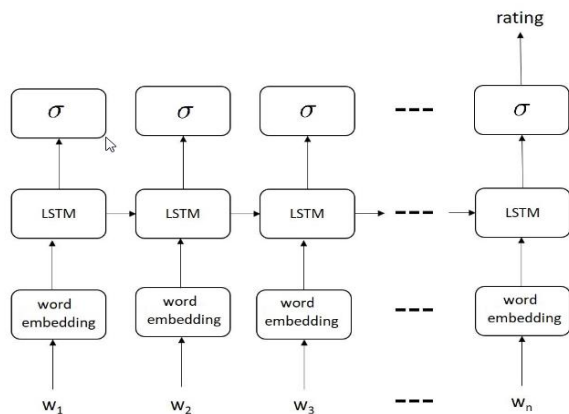
Problem rešavan u ovom radu može se posmatrati kao regresioni ali i klasifikacioni problem. U slučaju klasifikacije, u pitanju je multikategorijska klasifikacija. Primena binarnih klasifikatora moguća je i za probleme multikategorijske klasifikacije. Ideja ovog pristupa je da se obuču više modela, i da se njihovim glasanjem formira konačna predikcija.

Ansambl modeli na ulazu očekuju podatke u numeričkoj formi, pa je potrebno pretvoriti recenzije u numeričke podatke. U tu svrhu upotrebljen je TF-IDF algoritam, pripadnik *bag of words* grupe algoritama. Mane ovog algoritma su što se procenjuje jedna reč u datom trenutku (ne uzima se u obzir kontekst tj. okolina reči), a ne uzima u obzir ni semantičku analizu recenzija.

U poglavlju gde se opisuje uklanjanje autlajera, objašnjen je postupak sentiment analize nenadgledanim učenjem. Sentimenti dobijeni ovim postupkom uzeti su u obzir prilikom obučavanja ansambl modela. Upotrebljeni su bagging i boosting algoritmi: Random Forest, Ada Boost i Gradient boosting.

2.5. Rekurentna neuronska mreža

U kontekstu rešavanja problema u ovom radu, rekurentna neuronska mreža je idealan pristup, jer omogućuje procesiranje sekvenci. U nastavku je predstavljena struktura LSTM mreže koja je upotrebljena za rešavanje problema.



Slika 3. Arhitektura mreže, preuzeto iz rada [3]

Na slici 3 prikazana je arhitektura LSTM mreže. Na ulazu u model se očekuje sledeći format – reči predstavljene u vektorskom prostoru malih dimenzija koji definiše semantičke i sintaktičke osobine reči. Dakle, LSTM model na ulazu ima pretprocesirane reči recenzije, dok je izlaz sentiment recenzije. Za datu recenziju x , sa n reči $\{w_1, w_2, \dots, w_n\}$, najpre se svaka reč mapira na *word embedding* upotrebom pretreniranog Glove modela. Na izlaz svake LSTM jedinice primenjuje se sigmoid funkcija. Predikcija ocene je izlaz poslednje sekvence, pošto je ovakva arhitektura preslikavanje mnogo na jedan.

Rekurentna neuronska mreža trenirana je minimizacijom funkcije negativne verodostojnosti (eng. negative likelihood).

2.6. Hiperparametri modela

Za algoritam **Gradient boosting** hiperparametri su podešeni na sledeći način:

- $n_estimators = 50$, Predstavlja broj slabih klasifikatora, unakrsnom validacijom je utvrđena vrednost.
- $learning_rate = 1$, Brzina učenja, defaultna vrednost, nije promenjena jer se uobičajeno koristi.
- $max_depth = 15$, Dubina stabla odlučivanja, hiperparametar koji kontroliše overfitting, jer veće dubine dozvo-

ljavaju modelu da se previše prilagodi trening podacima, preporuka je da se parametar utvrdi unakrsnom validacijom, što je učinjeno.

Specifikacija hiperparametara za algoritam **Ada boost**:

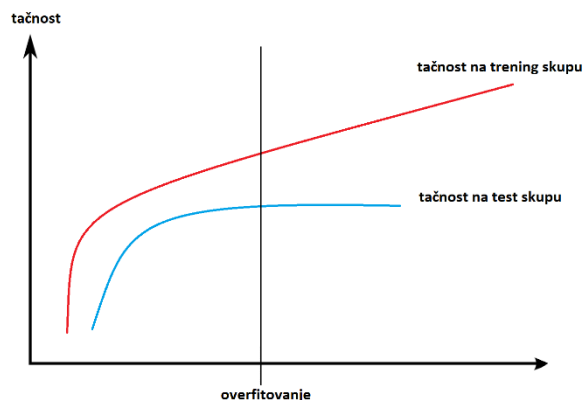
- $n_estimators = 80$, Predstavlja broj slabih klasifikatora, unakrsnom validacijom je utvrđena vrednost.
- $learning_rate = 1$, Brzina učenja, defaultna vrednost, nije promenjena jer se uobičajeno koristi.

Specifikacija hiperparametara za algoritam **Random Forest**:

- $n_estimators = 100$, Predstavlja broj slabih klasifikatora, unakrsnom validacijom je utvrđena vrednost.
- $min_samples_split = 10$, Minimalan broj uzoraka potreban da se razdvoji čvor stabla odlučivanja, određen unakrsnom validacijom.
- $min_samples_leaf = 4$, Minimalan broj uzoraka na svakom listu stabla odlučivanja, utvrđen unakrsnom validacijom.
- $max_depth = 30$, Dubina stabla odlučivanja, hiperparametar koji kontroliše overfitting, jer veće dubine dozvoljavaju modelu da se previše prilagodi trening podacima, preporuka je da se parametar utvrdi unakrsnom validacijom, što je i urađeno.
- $bootstrap = true$, Hiperparametar koji govori kako će se popunjavati stabla odlučivanja, ako je postavljen na false, čitav skup podataka će se koristiti u svakom stablu odlučivanja, upotrebljena je defaultna vrednost, jer se uobičajeno koristi.

Specifikacija hiperparametara **rekurentne neuronske mreže**:

- $epochs = 30$, Predstavlja broj epoha treniranja neuronske mreže.
- $batch_size = 4$, Kontroliše koliko često se ažuriraju težine u neuronskoj mreži. Optimizuje se tako što se za fiksiran broj epoha isprobavaju različite vrednosti. Odabir ovog parametra zavisi dosta i od hardverskih kapaciteta mašine na kojoj se mreža obučava..



Slika 4. Kriterijum za obustavljanje obučavanja mreže

Rekurentna neuronska mreža trenirana je na računaru sa sledećim specifikacijama: procesor *Intel i7-10750H*, grafička kartica *NVIDIA GeForce GTX 1650*, 16 GB RAM memorije.

3. REZULTATI I DISKUSIJA

U tabeli 1, prikazani su rezultati primenjenog modela rekurentne neuronske mreže. Ovaj pristup dao je najbolje rezultate, što opravdava činjenica da ima *state-of-the-art* status u oblasti rešavanja problema.

Preciznost	Odziv	F1 mera	Tačnost
77.81	64.32	61.54	82.03

Tabela 1. Performanse modela rekurentne neuronske mreže

Algoritmi	Ada boost	Gradient boost	Random Forest
F1 mera	54.32	55.16	57.88
tačnost	67.78	68.89	73.44

Tabela 2. Poređenje rezultata primenjenih modela ansambla

Na osnovu tabele 2, vidi se da je od korištenih modela ansambla najbolje rezultate postigao algoritam Random Forest. Priroda problema koji se rešava je po svojoj složenosti pogodnija za *bagging* algoritme. Zbog toga su očekivani ovakvi rezultati, da je najbolje performanse postigao predstavnik *bagging* algoritama (Random Forest).

Dakle od svih upotrebljenih modela najbolje se pokazao model rekurentne neuronske mreže. Pretpostavlja se da bi rezultati bili još uspešniji, da se koristio veći skup podataka za obučavanje mreže. Ono što bi takođe moglo da unapredi performanse je *word embedding*, konkretno mogao bi se upotrebiti pretrenirani model koji je obučavan nad većim skupom podataka.

4. ZAKLJUČAK

U ovom radu predstavljen je sistem predikcije ocena na osnovu tekstualnih recenzija korisnika. Značaj za rešavanje ovog problema jeste u proceni kvaliteta softvera, koji je bitan kako prodavcima, tako i korisnicima softvera. Problem je rešen na dva različita načina, sa upoređivanjem performansi ova dva pristupa.

Prvi pristup bila je klasifikacija pomoću modela ansambla uz sentiment analizu. Drugi pristup rešavanju problema bio je upotrebom rekurentne neuronske mreže.

Upoređujući performanse, zaključeno je da se rekurentna neuronska mreža pokazala kao bolji pristup u rešavanju problema. Pokazano je koliko je *word embedding* važan za oba pristupa i u kojoj meri utiče na performanse.

Takođe sentiment analiza koja se oslanja na *word embedding*, značajno pomaže u predikciji ocene na osnovu tekstualnih komentara. Glavna prednost u rešenju predstavljenom u ovom radu jeste kvalitetno preprocesiranje podataka. Sve nepravilnosti i šumovi koji bi mogli da naruše performanse primenjenih algoritama su uklonjeni. U procesu otklanjanja outlajera, pojavio se novi složen problem – sentiment analiza nenadgledanim učenjem. Tom problemu moglo bi se posvetiti i u daljim istraživanjima jer ima veliki značaj i primenu.

Problem koji se rešavao mogao bi se i generalizovati. Pored aplikacija na Google Play prodavnici, ljudi recenziraju i ocenjuju mnoge druge stvrari. Obrada prirodnog jezika vrlo je značajna oblast, i automatizovana procena zadovoljstva korisnika na osnovu njegovog komentara je takođe od velikog značaja.

5. LITERATURA

- [1] D. Monett and H. Stolte, "Predicting Star Ratings based on Annotated Reviews of Mobile Apps," Oct. 2016, pp. 421–428. doi: 10.15439/2016F141.
- [2] M. Umer, I. Ashraf, A. Mehmood, S. Ullah, and G. S. Choi, "Predicting numeric ratings for Google apps using text features and ensemble learning," *ETRI Journal*, vol. 43, no. 1, pp. 95–108, Feb. 2021, doi: 10.4218/etrij.2019-0443.
- [3] B. Gezici, N. Bolucu, A. Tarhan, and B. Can, "Neural Sentiment Analysis of User Reviews to Predict User Ratings," in *2019 4th International Conference on Computer Science and Engineering (UBMK)*, Samsun, Turkey, Sep. 2019, pp. 629–634. doi: 10.1109/UBMK.2019.8907234.
- [4] <https://www.kaggle.com/datasets/prakharrathi25/google-play-store-reviews>

Kratka biografija:



Marko Mijatović rođen je 25. oktobra 1998. godine u Subotici. Završio je Osnovnu školu „Kizur Istvan“ u Subotici 2013. godine odličnim uspehom, kao đak generacije. Gimnaziju „Svetozar Marković“ u Subotici završava 2017. godine. Diplomirao je na Fakultetu tehničkih nauka u Novom Sadu 2021. godine, na smeru Softversko inženjerstvo i informacione tehnologije.

ANOTACIJA I DETEKCIJA NEGACIJE U KOMENTARIMA PROIZVODA**ANNOTATION AND DETECTION OF NEGATION IN PRODUCT REVIEWS**

Aleksandra Milijević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – Rad je baziran na kreiranju sistema za anotaciju i detekciju negacije u komentarima razne vrste proizvoda na engleskom jeziku.

Ključne reči: Mašinsko učenje, Negativne reči, Opseg negacije

Abstract – This paper presents the system for the annotation and detection of negation in the reviews of various types of products in the English language.

Keywords: Machine learning, Negation cues, Scope of negation

1. UVOD

Poslednjih godina pojam negacije u prirodnom jeziku predstavlja sve veći objekat interesovanja. Detekcija negacije je važan deo u analizi sentimenata (eng. Sentiment Analysis) u oblasti Obrade prirodnog jezika (NLP) i smatra se jednim od najtežih problema za rešavanje.

Oblasti u kojima je detekcija negacije veoma bitna su klinički zapisi, komentari na društvenim mrežama, kupovina preko interneta, recenzije filmova. U slučaju prve oblasti zdravlje pacijenata zavisi od pravilno kreiranog sistema, dok u ostalim oblastima, samim korisnicima je veoma korisna informacija koji prizvod da kupe ili koji film da gledaju.

Vrste negacije koje postoje su neafiksne negacije sa jednom reči, afiksne negacije i negacije koje su iskazane pomoću više reči. U ovom radu su obrađene samo prve dve vrste.

Sistem kreiran u radu sastoji se iz sledećih delova: polu-automatske anotacije negativnih reči u rečenicama, detekcije negativnih reči, detekcije opsega negacije. U svakom od poslednja dva dela vrši se izbor atributa i klasifikacija uz pomoć algoritama mašinskog učenja. Za klasifikaciju negativnih reči koristi se Support Vector Machine (SVM), a za opseg negacije Conditional Random Fields (CRF) algoritam. Rezultati SVM algoritma evaluirani su pomoću kros-validacije i F1 mere, a CRF pomoću F1 mere.

2. PREGLED STANJA U OBLASTI

Postoji više pristupa koji mogu da se koriste za detekciju negacije. Najranija istraživanja kreirala su određena pravila pomoću kojih određuju negaciju u rečenicama (eng. Rule-based negation detection). Podaci koji su se koristili bili su najvećim delom iz medicinskog domena kao što su klinički izveštaji pacijenata. Rad [1] je jedan od prvih u kome je korišćen prethodno naveden pristup. Autori rada su razvili jednostavan algoritam regularnih izraza pod nazivom NegEx koji je postigao tačnost od 84.5%.

2007. godine u radu [2] kreiran je algoritam ConText koji predstavlja proširenje NegEx-a. Osim negacije vrši i određivanje da li su klinička stanja navedena u kliničkim izveštajima hipotetička, istorijska, ili ih je iskusio neko drugi a ne pacijent. 2008. godine [3] je objavio BioScope korpus medicinskih tekstova koji služe kao osnova za anotaciju korpusa negativnim rečima i opsega negacije prilikom detekcije negacije.

2009. objavljen je cTAKES algoritam koji nije uspevao da detektuje kompletne obrasce negacije. Iz tog razloga [4] je koristio parser zavisnosti za detekciju negacije čime su poboljšane performanse cTAKES sistema.

Autori rada [5] su 2012. godine objavili anotirani skup podataka sa negativnim rečima i opsegom negacije. Skup podataka sadrži poznate priče Konana Dojla i služi kao osnova za identifikovanje negacije u tekstovima.

U radu [6] određivalo se koje reči se negiraju ključnom reči negacije. Vršeno je poređenje više metoda za određivanje opsega negacije. Donesen je zaključak da se u najvećoj meri opseg negacije prostire u rasponu od 6 reči sa desne strane, dok sa leve stane u proseku od 3 reči.

U radu [7] korišćen je pristup mašinskog učenja za detekciju negacije. Koristili su CRF model koji je obučen na osnovu atributa ekstrahovanih pomoću parsera zavisnosti na BioScope i Product Reviews korpusu. U radu [8] opisan je prvi sofisticirani sistem za detekciju obima negacije prilikom analize sentimenata Twitter tekstova. [9] je predložio lingvistički model zasnovan na SVM i CRF algoritmu za detekciju negacije. SVM se koristi za klasifikaciju reči negacije na osnovu maksimalne margine, dok je detekcija opsega negacije vršena kao problem sekvencijalnog labelovanja korišćenjem CRF algoritma.

3. TEORIJSKI POJMOVI I DEFINICIJE**3.1. Definicija i primeri negacije**

Negacija je konverzija afirmativne rečenice koja pretvara prostu potvrdnu rečenicu u negativnu. Identifikuje se

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kovačević, red.prof.

pomoću znaka negacije, koji se sastoji od jednog ili više tokena koji signaliziraju negaciju.

Oni mogu biti od jedne reči, više reči, ili afiksni kao što je prefiks *im-* u reči *impossible* u primeru 3.1.5, ili sufiks *-less* u *endless*, a u nekim slučajevima reči koje povezuju dve ili više negativnih alternativa, kao što su *neither/nor*. Postoji opseg negacije, gde su reči samo unutar tog opsega pogođene negacijom, a ne cela rečenica. Postoji situacije u kojima reči negacije nemaju opseg (3.1.3), ili je opseg dvosmisle kao u primeru 3.1.1. U daljim primerima znak negacije je označen podebljanim slovima, a opseg uglastim zagradama.

- Jedna reč: *No, nor, never, not*

3.1.1 [Anna is] **not** [short and fat].

3.1.2 We have two dogs, **neither** [one can sit].

3.1.3 **No**, we did not go to the market.

- Više reči: *no longer, by no means, for nothing*

3.1.4 [There is] **no longer** [snow outside].

- Afiksi: *im-, un-, dis-, ir-, in-, -less*

3.1.5 [Boys are] **impossible** [to understand].

U ovom radu uzete su u obzir samo afiksne i neafiksne negacije sa jednom reči.

3.2. SVM algoritam

SVM je nadgledani algoritam za učenje koji se koristi za probleme klasifikacije i regresije. Može da rešava linearne

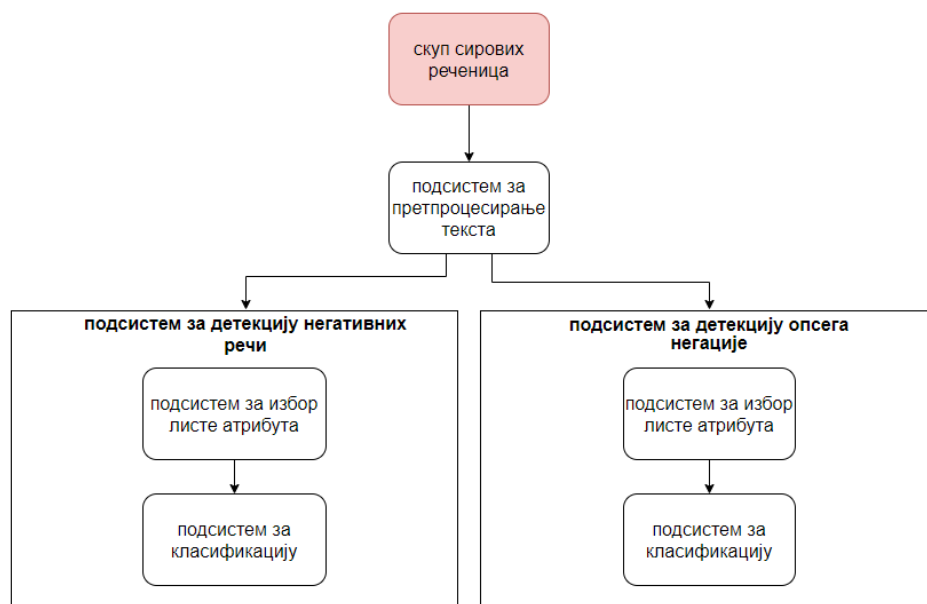
i nelinearne probleme i ideja je ta da algoritam kreira liniju ili hiperravan koja razdvaja podatke u klase. Bitan hiperparametar kod ovog algoritma je parametar regularizacije C i govori nam koliko pogrešne klasifikacije želimo da izbegnemo. Za velike vrednosti C javlja se tvrdi SVM i on ne dozvoljava greške u klasifikaciji (može izazvati overfitting). Dok za male vrednosti C javlja se meki SVM koji dozvoljava greške prilikom klasifikacije.

3.3. CRF algoritam

CRF je tipa diskriminativnog klasifikatora. Ulazni podaci su sekvencijalni i mora se uzeti u obzir prethodni kontekst kada se vrši predviđanje. Nedostatak je visoka računaska složenost faze obuke algoritma. CRF ne radi sa nepoznatim rečima, odnosno rečima koje nisu bile prisutne u uzorku podataka za obuku. Neke od prednosti: moguće je postići visok kvalitet označavanja ako se odaberu prave karakteristike; fleksibilan u pogledu izbora karakteristika itd. Jedni od bitnih hiperparametara su maksimalan broj iteracija, $c1$ i $c2$ koeficijenti za $L1$ i $L2$ regularizaciju respektivno.

4. METODOLOGIJA

Sistem se sastoji od tri podsistema: podsistema za pretprocesiranje teksta, podsistema za detekciju negativnih reči i podsistema za detekciju opsega negacije, slika 1.



Slika 1. Arhitektura sistema

4.1. Podsistem za predprocesiranje

U prvom podsistemu vršeno je predprocesiranje rečenica uz pomoć zbirke alata Stanza gde su primenjene tehnike tokenizacije i lematizacije. Svaka rečenica je podeljena na reči i za svaku reč čuvane su sledeći atributi: token, lema, POS tag (univerzalni deo govora - part-of-speech), id glave, deprel (odnos zavisnosti između posmatrane reči i njene sintetičke glave). Takođe je vršena anotacija negativnih reči gde je sačuvana informacija da li je reč negativna ili ne. Postoje dva leksikona: leksikon negativnih reči i leksikon afiksa na osnovu kojih je određen poslednje naveden atribut (tabele 1. i 2.).

Tabela 1. Leksikon negativnih reči

hardly	neednt	isnt	without
neither	wasnt	wouldnt	no
nobody	lack	never	nowhere
not	nor	nothing	cant
cannot	none	dont	doesnt
didnt	n't	hasnt	havnt
havent	hadnt	mightnt	shouldnt

Tabela 2. Leksikon negativnih prefiksa i sufiksa

un-	il-	anti-
in-	ir-	non-
im-	dis-	-less

4.2. Podsistemi za detekciju negativnih reči i opsega

Podsistemi za detekciju negativnih reči i opsega negacije dele se na dva dela: određivanje liste atributa, klasifikacija. Atributi izabrani kod prvog podsistema su: token, lema, POS tag, lema prethodne/naredne reči posmatranog tokena, POS tag prethodne/naredne reči posmatranog tokena, afiks (kod afiksne negacije), n-gram afikslnih negativnih reči (deo reči bez afiksa). Što se tiče podsistema za detekciju opsega negacije, atributi koji su korišćeni su isti kao i za prvi podsistem (bez atributa vezanih za afikse) sa dodatkom sintaksičkih atributa: veze zavisnosti (deprel), POS tag glave posmatranog tokena, usmerena putanja zavisnosti od glave negativne reči do svakog tokena (najkraći put od glave negativne reči do posmatranog tokena - korišćen Dijkstrin algoritam). U fazi klasifikacije, SVM je korišćen za detekciju negativnih reči, dok za detekciju opsega je upotrebljen CRF. Oba algoritma su pokazala dobre rezultate u ranijim radovima koji su obrađivali ovu temu.

5. EKSPERIMENTI I REZULTATI

U radu su vršena dva eksperimenta. Prvi se odnosi na detekciju negativnih reči, i testirane su razne kombinacije atributa i vrednosti hiperparametara da bi se dobio optimalan rezultat. Drugi eksperiment je vezan za detekciju opsega negacije i takođe su vršene razne kombinacije atributa i vrednosti hiperparametara. Ulaz u oba podsistema je rečnik reči gde je za svaku reč označeno kojoj rečenici pripada, i svaka sadrži svoje osobine dobijene iz faze preprocesiranja gde je bio upotrebljen Stanza paket, kao i da li je reč negativna ili ne.

5.1. Skup podataka

Skup podataka koji je korišćen u radu je SFU Review korpus koji se sastoji iz više manjih korpusa sa recenzijama raznih proizvoda. Za svaku kategoriju postoji 25 pozitivnih i 25 negativnih txt fajlova. Postoji i anotirana verzija ovog skupa podataka koja sadrži anotaciju negacije i spekulacije. Međutim za potrebe ovog rada vršena je zasebna anotacije. Skup rečenica je preuzet sa kaggle¹ sajta gde je isti ovaj korpus podeljen na pojedinačne rečenice i nalazi se u csv formatu. Zbog preobimnosti od 17 000 rečenica izdvojeno 2000.

5.2. Eksperiment 1 - detektovanje negativnih reči

U prvom eksperimentu korišćeni su atributi koji su navedeni u podsistemu za detekciju negativnih reči u poglavlju metodologija. Za obučavanje je korišćen SVM algoritam, gde je za kernel funkciju izabran RBF. Testiran je hiperparametar C, čije su vrednosti uzete u opsegu 0.01-0.1. Sistem je evaluiran pomoću kros-validacije i to je uzeta u obzir F1 mera. Bazni model je kreiran samo pomoću tokena i leme reči. Pokazano je da se najbolji rezultat dobija kada je vrednost C=0.1.

[1] <https://www.kaggle.com/datasets/ma7555/the-sfu-review-corpus-negation-annotated>

U narednoj tabeli biće prikazani rezultati dobijeni raznim kombinacijama atributa.

Tabela 3. Razne kombinacije atributa za SVM algoritam

token, lema, pos	91.2%
token, lema, pos, bigram leme prethodne i posmatrane reči, pos prethodne reči	90.8%
token, lema, pos, bigram leme naredne i posmatrane reči, pos naredne reči	90.8%
token, lema, pos, bigram lema prethodne/naredne reči i posmatrane reči, pos prethodne/naredne reči	91.3%
token, lema, pos, bigram lema prethodne/naredne reči i posmatrane reči	91.35%
token, lema, pos, bigram lema prethodne/naredne reči i posmatrane reči, afiks, karakter 5-gram	91.44%

5.3. Eksperiment 2 - detektovanje opsega negacije

U drugom eksperimentu korišćeni su atributi koji su navedeni u podsistemu za detekciju opsega negacije u poglavlju metodologija. Testirane su razne kombinacije ovih atributa. Vršeno je BIO labelovanje gde je svaka negativna reč označena sa B, reči koje upadaju u opseg negacije sa I, a sve ostale reči u rečenici sa O. Opseg negacije se proteže u rasponu od 6 reči sa desne strane negativne reči i 3 reči sa leve strane. Za obučavanje je korišćen CRF, a u okviru njega lbfgs algoritam za obučavanje. Hiperparametri čije su razne vrednosti testirane su maksimalan broj iteracija (u opsegu 50-250), c1 i c2 koeficijenti u opsegu 0.1-0.3. Skup podataka je podeljen na obučavajući i test skup i korišćena je random vrsta uzorkovanja. Rešenja su evaluirana pomoću makro F1 mere pošto je u pitanju višeklasna klasifikacija. Najbolja kombinacija hiperparametara je sledeća: max_iterations=100, c1=0.3, c2=0.1, gde F1 mera iznosi 78.46%. Kombinacija atributa se nalaze u narednoj tabeli.

Tabela 4. Razne kombinacije atributa za CRF algoritam

token, lema, pos	77.02%
token, lema, pos, pos glave	68.43%
token, lema, pos, pos glave, deprel	71.54%
token, lema, pos, deprel	75.86%
token, lema, pos, deprel, bigram lema naredne i posmatrane reči	77.26%
token, lema, pos, deprel, bigram lema prethodne i posmatrane reči	80.65%
token, lema, pos, deprel, bigram lema naredne/prethodne i posmatrane reči	82.34%
token, lema, pos, deprel, bigram lema naredne/prethodne i posmatrane reči, usmerena zavisna putanja	83.82%

6. DISKUSIJA

Rezultati dobijeni u evaluaciji sistema su veoma dobri kako za detekciju negativnih reči (F1=91.44%), tako i za detekciju opsega negacije (83.82%). U radu [10] za detekciju negativnih reči rezultat iznosi 93.49%, a za opseg 77.98%. Na prvi pogled sistem za detektovanje negativnih reči je bolji u pomenutom radu, dok je detekcija opsega bolja u ovom radu. Međutim, treba uzeti u obzir više stvari.

U radu [10] detekcija negacije vršena je na ConanDoyle skupu podataka koji sadrži 5520 rečenica, dok u ovom radu skup sadrži 2000 rečenica. Iako sam postupak testiranja nije bio u potpunosti identičan, izveden je sličan zaključak, a to je da kombinacija tokena, leme, POS taga, bigrama posmatrane reči sa prethodnom/narednom reči, afiksa i 5-grama bez afiksa afikslnih negativnih reči daje najbolje rezultate.

Što se tiče opsega negacije, atributi takođe nisu bili identični. U radu [10] su se, pored ovde korišćenih atributa, koristili i POS tag glave drugog reda (glava glave), graf zavisnosti od svake reči do negativne reči, tip negacije (jednorečna, višerečna, afikslna). Sa druge strane, u pomenutom radu skup podataka je već bio anotiran sa negacijom, dok je u ovom radu anotacija poluautomatska i vršena je zasebno. Za opseg negacije nije uzeto u obzir kada se afikslna negativna reč nalazi na kraju rečenice pa ceo deo rečenice pre negativne reči spada u opseg negacije.

7. ZAKLJUČAK

U ovom radu predstavljen je sistem za anotaciju i detekciju negacije u rečenicama. Motivacija je bila to što postoje razni tekstovi na internetu čija analiza je od velikog značaja. Detekcija negacije je ključna u toj analizi. Predstavljen sistem sastoji se od podistema za preprocesiranje teksta gde se vrši anotacija negativnih reči, podistema za detekciju negativnih reči i podistema za detekciju opsega negacije. Skup podataka nad kojim je testiran ovaj sistem sastoji se od 2000 rečenica iz SFU Review korpusa. Detekcija negativnih reči vršena je pomoću SVM algoritma, a opsega negacije pomoću CRF-a. Za evaluaciju SVM algoritma upotrebljena je krosvalidacija zasnovana na F1 meri i najbolji rezultat iznosi 91.44%. Za CRF korišćena je makro F1 mera i rezultat nad test skupom iznosi 83.82%.

U nastavku razvoja ovog sistema postoje razne stavke koje bi se mogle poboljšati. Pre svega, prilikom anotacije negativnih reči mogle bi se uzeti u obzir i negacije izražene sa više reči. Za određivanje opsega negacije treba ispitati i slučajeve kada je negativna afikslna reč na kraju rečenice. Što se tiče algoritama klasifikaciju, postoje i drugi atributi koji bi se mogli uzeti u obzir, kao na primer graf zavisnosti od svake reči do negativne reči. Algoritam koji je takođe veoma zastupljen u detekciji negacije jeste RNN neuronska mreža, pa bi se i ona mogla uključiti u sistem.

8. LITERATURA

- [1] W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper, and B. G. Buchanan, "A simple algorithm for identifying negated findings and diseases in discharge summaries," *J. Biomed. Informat.*, vol. 34, no. 5, 2001.
- [2] P. D. Turney and M. L. Littman, "Unsupervised learning of semantic orientation from a hundred-billion-word corpus," 2002. Доступно: <https://arxiv.org/ftp/cs/papers/0212/0212012.pdf>
- [3] V. Vincze, G. Szarvas, R. Farkas, G. Móra, and J. Csirik, "The bioscope corpus: Biomedical texts annotated for uncertainty, negation and their scopes," *BMC Bioinf.*, vol. 9, no. 11, pp. 1–9, 2008.
- [4] S. Sohn, S. Wu, and C. G. Chute, "Dependency parser-based negation detection in clinical narratives," *AMIA Summits Transl. Sci. Proc.*, vol. 2012, Oct. 2012.
- [5] R. Morante and W. Daelemans, "Conandoyle-neg: Annotation of negation in conan doyle stories," in *Proc. 8th Int. Conf. Lang. Resour. Eval., Istanbul, Turkey, 2012*.
- [6] A. Hogenboom, P. van Iterson, B. Heerschof, F. Frasinca, and U. Kaymak, "Determining negation scope and strength in sentiment analysis," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2011.
- [7] I. Councill, R. McDonald, and L. Velikovich, "What's great and what's not: Learning to classify the scope of negation for improved sentiment analysis," in *Proc. Workshop Negation Speculation Natural Lang. Process.*, 2010.
- [8] J. Reitan, J. Faret, B. Gambäck, and L. Bungum, "Negation scope detection for Twitter sentiment analysis," in *Proc. 6th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal.*, 2015.
- [9] M. Enger, E. Velldal, and L. Øvrelid, "An open-source tool for negation detection: A maximum-margin approach," in *Proc. Workshop Comput. Semantics Beyond Events Roles*, 2017.
- [10] M. Enger, "A portable toolkit for detecting negation," in *UiO:Department Of Informatics*, 2016, pp.5-6.

Kratka biografija:



Aleksandra Milijević rođena je u Beogradu 1998. god. Master rad na Fakultetu tehničkih nauka iz oblasti Računarstva i automatike odbranila je 2022.god.

kontakt: aleksandramilijevic98@gmail.com

UTVRĐIVANJE RASA PASA I MAČAKA UPOTREBOM KONVOLUCIONIH NEURONSKIH MREŽA**DOG AND CAT BREED DETERMINATION USING CONVOLUTIONAL NEURAL NETWORKS**

Petar Nikolić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – SOFTVERSKO INŽENJERSTVO I INFORMACIONE TEHNOLOGIJE

Kratak sadržaj – U radu su predložena dva pristupa rešavanju problema utvrđivanja rase pasa i mačaka na slikama. Prvi pristup predstavlja klasifikaciju u jednom koraku u kojem se utvrđuje rasa životinje bez prethodnog utvrđivanja njene vrste. U drugom pristupu, rasa se utvrđuje u dva koraka – u prvom se određuje vrsta životinje, dok se u drugom slika prosleđuje specijalizovanom klasifikatoru koji utvrđuje rasu u okviru prethodno utvrđene vrste. Implementacija oba pristupa zasniva se na upotrebi konvolucionih neuronskih mreža, kao i tehnike augmentacije podataka. Dodatno, kod oba pristupa ispitan je i uticaj primene tehnike aktivnog učenja.

Ključne reči: klasifikacija slika, konvolucione neuronske mreže, augmentacija podataka, aktivno učenje

Abstract – The paper proposes two approaches to solving the problem of determining the breed of dogs and cats in images. The first one is a one-step classification without previously determining the species of the animal. The second one consists of two steps - in the first one, the image is classified based on the species. In the second one, the image is passed to a specialized classifier which determines the breed within the previously determined species. Both solutions are based on the use of convolutional neural networks, as well as the data augmentation technique. Additionally, the impact of the application of the active learning technique was analyzed for both proposed solutions.

Keywords: image classification, convolutional neural networks, data augmentation, active learning

1. UVOD

Kućni ljubimci, pogotovo psi i mačke kao neki od najpopularnijih, imaju bitnu ulogu u životu ljudi, od razonode, preko emocionalne podrške, do pomoći u slučaju zdravstvenih problema. Mačkama i psima se posvećuje dosta vremena na internetu, te tako postoji i veliki broj aplikacija koje su im posvećene, kao što su *Dog Buddy* [1], *MeowTalk* [2] i *Relax My Cat* [3].

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kovačević, red. profesor.

Neretko se pak dešava da vlasnici nisu dovoljno upoznati sa rasama svojih pasa i mačaka, što može dovesti do nesporazuma među vlasnicima, kako uživo tako i *online*. Stoga bi postojanje načina za pouzdano i automatsko utvrđivanje rase ljubimaca na osnovu slike bilo značajno za njih.

Utvrdjivanje rase pasa i mačaka predstavlja problem višeklasne klasifikacije slika. Rešavanje ovog problema izazovno je iz više razloga. Jedan jesu velike razlike u izgledu koje mogu postojati u okviru jedne rase ali i između rase pasa i mačaka (boja, dužina krzna itd). Drugi problem se tiče kvaliteta samih slika, odnosno visoke varijabilnosti po pitanju rezolucije i dimenzija slike, te osvetljenosti i zamućenosti.

U ovom radu predloženo je rešenje problema utvrđivanja rase pasa i mačaka na primeru 3 rase pasa i 3 rase mačaka. Rešenje se zasniva na višeklasnoj klasifikaciji slika upotrebom konvolucione neuronske mreže sa samostalno razvijenom arhitekturom. Radi rešavanja prvog problema ispitan je utvrđivanje rase sa i bez prethodnog utvrđivanja vrste životinje. U cilju dobijanja robusnih modela kojima bi mogao biti rešen i problem kvaliteta slika, u rešenju je korišćena i tehnika augmentacije podataka primenom različitih transformacija slike. Dodatno, radi dobijanja manjeg trening skupa, a uz očuvanje ili unapređenje performansi rešenja, na modelima korišćenim za klasifikaciju ispitan je uticaj primene tehnike aktivnog učenja.

U drugom poglavlju dat je opis korišćenog skupa podataka. U trećem poglavlju opisane su metode i tehnike korišćene prilikom implementacije rešenja. U četvrtom poglavlju opisano je predloženo rešenje. U petom poglavlju prikazani su ostvareni rezultati primene predloženog rešenja i analizirane njegove greške. U šestom poglavlju nabrojani su najvažniji zaključci i navedeni mogući pravci daljeg razvoja.

2. SKUP PODATAKA

U cilju implementacije predloženog rešenja, iskorišćen je *The Oxford-IIIT Pet Dataset* [4] skup podataka sa ukupno 7,349 slika pasa i mačaka. Od tog broja, 4,978 slika za 25 rase pasa i 2,371 sliku za 12 rase mačaka. Za svaku od rase postoji po 200 slika. Dodatno, sve slike u skupu poseduju anotacije koje sadrže naziv rase životinje na slici, zatim koordinate pravougaonika koji sadrži glavu životinje, kao i metapodatke o svojstvima same slike, kao što su njene dimenzije i naziv.

Za potrebe razvoja i testiranja predloženog rešenja korišćen je podskup goreopisanog skupa koji se sastoji od slika po 3 rase pasa i mačaka. Konkretno, u pitanju su *Basset*, *Leonberger* i *Samoyed* u slučaju pasa i *Persian*, *Egyptian Mau* i *Siamese* u slučaju mačaka.

3. KORIŠĆENE METODE I TEHNIKE

Od tehnika mašinskog učenja prilikom implementacije predloženog rešenja korišćene su konvolucione neuronske mreže, augmentacija podataka i aktivno učenje. Pomoću konvolucionih neuronskih mreža kreirani su modeli za potrebe klasifikacije, augmentacija podataka je iskorišćena za podizanje kvaliteta trening i test skupova, dok je aktivno učenje upotrebljeno radi smanjenja obima trening skupa uz očuvanje ili unapređenje performansi modela.

3.1. Neuronske mreže

U cilju kreiranja modela koji obavljaju klasifikaciju slika pasa i mačaka, korišćene su konvolucione neuronske mreže sa samostalno razvijenom arhitekturom, koja sadrži 6 konvolucionih slojeva, koji koriste *ReLU* aktivacionu funkciju i ispraćeni su svaki sa po jednim *Max Pooling* i *Dropout* slojevima. Nakon konvolucionih, sledi jedan potpuno povezani sloj, praćen poslednjim slojem sa *Softmax* aktivacionom funkcijom.

3.2. Aktivno učenje

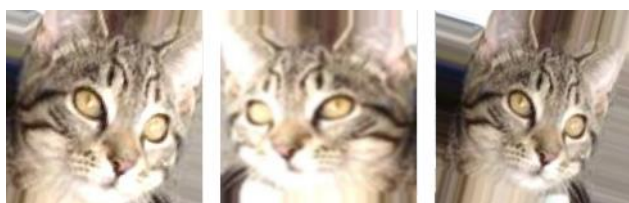
Aktivno učenje predstavlja tehniku polunadgledanog učenja koja se koristi u cilju smanjenja količine podataka potrebne za obučavanje modela mašinskog učenja, a sa očuvanjem performansi ili njihovim unapređenjem [5]. Ovo se postiže iterativnim povećavanjem trening skupa, gde se u svakoj iteraciji utvrđuju najinformativniji podaci, to jest podaci iz kojih bi model mogao najviše naučiti.

Glavni problem primene ove tehnike jeste odabir metrike za informativnost podataka, budući da postoji dosta različitih metrika, počev od nasumičnog odabira kao najjednostavnije, do metrika koje se zasnivaju na različitim teorijskim pretpostavkama.

Prilikom implementacije predloženog rešenja, ispitan je uticaj primene 4 različite metrike – *EP Measure*, *BvSB*, kao i *Bavg* i *Bmax* varijante *Bernoulli entropy* metrike.

Tehnika aktivnog učenja u ovom radu primenjena je na sledeći način:

1. iz celog trening skupa T izdvaja se pseudonasumično odabrani podskup P , koji sadrži 20% slika ukupnog skupa
2. iz podskupa P izdvajaju se trening i validacioni skupovi u odnosu 90-10 i obavlja se treniranje modela u $30 - 5 \times k$ epoha, gde broj k predstavlja trenutnu iteraciju aktivnog učenja



Slika 1. Primeri slika dobijenih augmentacijom podataka

3. primena treniranog modela na skup T i rangiranje dobijenih predikcija primenom metrike informativnosti
4. prebacivanje N najinformativnijih slika iz skupa T u skup P
5. ponavljanje koraka 2-4 u K iteracija

Hiperparametri goreopisanog algoritma su broj iteracija K , broj najinformativnijih podataka N , odnos veličina trening i validacionog skupa, kao i primenjena metrika informativnosti i njihove vrednosti su određene empirijski.

3.3. Augmentacija podataka

Radi dobijanja modela koji su otporniji na loš kvalitet slika, te kako bi se nadomestio mali broj slika po rasi u korišćenom skupu podataka, primenjena je tehnika augmentacije podataka u tri koraka:

1. Učitavanje slika, kao i koordinata pravougaonika koji sadrži glavu životinje iz anotacija radi kreiranja isečka originalne slike sa glavom životinje. Isečki su dimenzija 224×224 piksela.
2. Nad dobijenim isečcima primenjuje se jedna ili više transformacija, i to rotacija ulevo i udesno za do 15 stepeni, horizontalno okretanje, isecanje po obe ose do 10%, zumiranje u opsegu $0.95 - 1.25 \times$, promena osvetljenosti u opsegu od -30% do $+40\%$. Broj transformacija, kao i vrednosti u navedenim opsezima određuju se pseudonasumično za svaki isečak ponaosob.
3. Primena *Gaussian Blur* transformacije sa kernelom dimenzija 7×7 nad oko 30% pseudonasumično odabranih transformisanih isečaka.

Na slici 1 prikazani su primeri slika dobijenih goreopisanom postupkom. Nakon primene augmentacije podataka, dobijen je konačan obučavajući skup sa 6,264 slika, sa 1,044 slika po rasi.

4. OPIS PREDLOŽENOG REŠENJA

Tokom razvoja predloženog rešenja isprobana su dva pristupa klasifikaciji rase pasa i mačaka. Sledi opis oba pristupa.

4.1. Prvi pristup – klasifikacija u jednom koraku

Prvi pristup predstavlja utvrđivanje rase pasa i mačaka u jednom koraku, bez prethodnog utvrđivanja vrste životinje na slici. U tu svrhu korišćena je konvoluciona neuronska mreža sa samostalno razvijenom arhitekturom, koja je obučena da prepozna 3 rase pasa i 3 rase mačaka. Ovaj model će u nastavku nositi oznaku CNN-1.

Prednost ovog pristupa jeste lakoća njegove implementacije, budući da se oslanja na samo jednu neuronsku mrežu. Slabost pristupa leži u tome što dobijeni model može napraviti krupnu grešku ukoliko sliku psa klasifikuje kao sliku mačke i obrnuto.

4.2. Drugi pristup – klasifikacija u dva koraka

Drugi pristup predloženog rešenja (CNN-2S u nastavku) predstavlja utvrđivanje rase u dva koraka:

1. U prvom koraku, slika se klasifikuje po principu pas-mačka upotrebom konvolucione neuronske mreže sa samostalno razvijenom arhitekturom (u nastavku CNN-2). Mreža je trenirana na celom trening skupu, podeljenom u klase *dog* i *cat*.
2. U drugom koraku se slika na osnovu rezultata prvog koraka prosleđuje jednoj od dve specijalizovane konvolucione neuronske mreže sa samostalno razvijenom arhitekturom. Jedna od ovih mreža specijalizovana je za klasifikaciju po rasama pasa, a druga po rasama mačaka. Ove mreže su trenirane na odgovarajućim podskupovima celog trening skupa koji sadrže samo slike pasa, odnosno samo slike mačaka. Mreža koja obavlja klasifikaciju po rasama pasa u nastavku će nositi oznaku CNN-D, dok će mreža za klasifikaciju mačaka nositi oznaku CNN-C.

4.3. Ispitivanje metrika informativnosti

Uticaj primene tehnike aktivnog učenja ispitivan je kod oba pristupa, za sva 4 modela. Broj iteracija primene aktivnog učenja K je za sva četiri modela iznosio 4, dok je vrednost hiperparametra N određena eksperimentalno za svaku iteraciju primene za svaki od četiri korišćena modela ponaosob (tabela 1).

Tabela 1. Vrednost hiperparametra N po modelima i iteracijama

Iteracija	CNN-1	CNN-2	CNN-D	CNN-C
1	1,050	1,050	525	525
2	1,050	1,050	525	525
3	900	900	450	450
4	900	900	450	450

Radi evaluacije primene odabranih metrika, postupak ispitivanja metrika informativnosti je odrađen tri puta i uporedo su analizirane srednje vrednosti tačnosti (*accuracy*) i *loss*-a na validacionom skupu, i to po završetku druge iteracije primene aktivnog učenja. Srednje vrednosti su analizirane budući da se početni trening skup u prvom koraku postupka određuje pseudonasumično.

Druga iteracija je odabrana zbog činjenice da su sva četiri modela ostvarila jako dobre performanse već na toj iteraciji, a uz korišćenje samo oko 53% celog trening skupa, kao i da značajnijeg povećanja performansi u sledećim iteracijama nije bilo.

Tabela 2 sadrži prikaz najboljih metrika informativnosti po modelima, kao i ostvarene srednje vrednosti tačnosti i *loss*-a.

Tabela 2. Rezultati ispitivanja metrika informativnosti

Model	Metrika	Tačnost (%)	Gubitak
CNN-1	Bmax	99.6	0.0145
CNN-2	Bmax	99.2	0.0248
CNN-D	EP Measure	100	0.0029
CNN-C	BvSB	99.8	0.0136

5. REZULTATI I DISKUSIJA

Oba predložena pristupa evaluirana su pomoću $F1$ mere nad test skupom. U nastavku je dat opis korišćenog test

skupa i postupka evaluacije. Dodatno, navedena je i analiza grešaka korišćenih modela.

5.1. Test skup

Za potrebe testiranja performansi predloženog rešenja, korišćen je test skup od ukupno 420 slika, sa 60 slika po rasi. Skup je formiran odabirom po 7 slika za svaku rasu, čime je dobijen inicijalni skup sa 42 slike. Nad ovim skupom je zatim primenjen postupak augmentacije podataka identičan onom koji je primenjen za kreiranje konačnog trening skupa, a koji je opisan u trećem poglavlju.

Kreiranje ovakvog test skupa je bilo potrebno kako bi se mogle proveriti performanse dobijenih modela u slučaju potrebe za klasifikacijom slika niskog kvaliteta i različitih karakteristika, koje se mogu očekivati u realnim uslovima korišćenja.

5.2. Postupak testiranja i ostvareni rezultati

Obavljeno je uporedo testiranje oba pristupa, i to sa i bez primene aktivnog učenja. U slučaju kada aktivnog učenje nije primenjeno modeli su trenirani na celokupnim trening skupovima, dok su u drugom slučaju obučavani primenom dve iteracije aktivnog učenja, uz korišćenje one metrike informativnosti koja se za taj model pokazala najboljom.

U tabeli 3 prikazane su ostvarene vrednosti $F1$ mere za svaki od modela. Druga kolona sadrži vrednosti u slučaju kada ova tehnika nije primenjena (Klasično učenje), a treća u slučaju kada jeste (Aktivno učenje). Poslednji red u tabeli sadrži ukupnu vrednost ove mere za drugi pristup, budući da se on zasniva na upotrebi tri različita modela.

Tabela 3. Rezultati testiranja predloženog rešenja

Model/Pristup	Klasično učenje	Aktivno učenje
CNN-1	0.9361	0.9786
CNN-2	0.9952	0.9881
CNN-D	0.9762	0.9521
CNN-C	0.9845	0.9667
CNN-2S	0.9738	0.9476

Dobijeni rezultati pokazuju da se drugi pristup pokazao osetno boljim u slučaju klasičnog učenja, ali da se u slučaju aktivnog učenja pokazao nešto gorim od prvog pristupa.

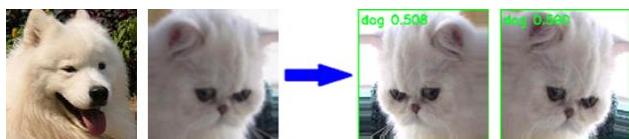
Takođe, može se zaključiti da su se po pitanju performansi modeli dobijeni primenom aktivnog učenja pokazali kao uporedivi modelima dobijenih primenom klasičnog učenja, a uz korišćenje tek samo oko 53% ukupnog trening skupa. Izuzetak predstavlja model CNN-1, koji je pokazao osetno bolje performanse prilikom primene aktivnog učenja.

5.3. Analiza grešaka modela

Svi korišćeni modeli pokazali su odlične performanse uz mali broj grešaka. Razlozi za nastale greške se u većini slučajeva mogu naći u niskom kvalitetu slika ili jakim deformacijama nastalim primenom transformacija tokom augmentacije test skupa.

Određen broj grešaka može se objasniti i sličnošću u izgledu između nekih rasa pasa i mačaka (npr. boja krzna).

Model CNN-2 dobijenom primenom aktivnog učenja je tako pogrešno klasifikovao sliku sa mačke kao sliku psa (slika 2). Ova greška se lako objašnjava velikom vizuelnom sličnošću u izgledu mačke rase *Persian*, koja može imati krzno bele boje i psa rase *Samoyed*, koji po pravilu ima krzno bele boje.



Slika 2. Greške modela (desno) usled sličnosti sa drugom rasom (levo)

Takođe, model CNN-2 dobijen bez primene aktivnog učenja je pogrešno klasifikovao psa rase *Basset* kao mačku (slika 3). U ovom slučaju je kombinacija loših karakteristika originalne slike, to jest sličnosti u boji glave psa i okoline i delimične vidljivosti njegovog lica, i primenjenih transformacija prilikom augmentacije (rotacija i zamućenje) dovela do greške u klasifikaciji.



Slika 3. Greška modela (desno) usled deformisanosti finalne i loših karakteristika polazne slike

6. ZAKLJUČAK

Razmotren je problem utvrđivanja rasa pasa i mačaka, kao i značaj za njegovo rešavanje. Predložena su dva različita pristupa za njegovo rešavanje, koja se zasnivaju na upotrebi konvolucionih neuronskih mreža i tehnike augmentacije podataka. Dat je opis korišćenog skupa podataka, kao i primenjenog načina njegove augmentacije, zatim koncepta aktivnog učenja, kao i arhitekture korišćene neuronske mreže. Predloženo rešenje je testirano i rezultati verifikovani.

Na osnovu dobijenih rezultata, pokazalo se da su oba pristupa dala odlične rezultate. Dodatno, kod oba pristupa obavljeno je poređenje rezultata dobijenih sa i bez primene tehnike aktivnog učenja, koje je pokazalo da je moguće smanjenje trening skupa od oko 47%, a uz očuvanje ili unapređenje performansi treniranih modela.

Iako su oba pristupa dala uporedive rezultate, prednost treba dati drugom pristupu zbog njegove modularnosti. Naime, dodavanje podrške za novu vrstu životinja bi se svodilo na treniranje modela koji obavlja klasifikaciju rasa u okviru te vrste i dodatno treniranje modela koji se koristi u prvom koraku, dok narušavanja performansi modela korišćenih u drugom koraku za postojeće vrste ne bi bilo.

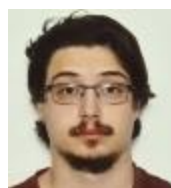
Na osnovu svega navedenog, klasifikacija u dva koraka, gde se u prvom određuje vrsta, a u drugom rasa psa ili mačke, u kombinaciji sa primenom tehnika aktivnog učenja i augmentacije podataka, predstavlja dobro rešenje problema utvrđivanja rasa pasa i mačaka na slikama.

Moguća proširenja predloženog rešenja jesu implementacija podrške za detekciju pasa i mačaka na slikama i klasifikaciju više životinja na slici, kao i povećanje broja podržanih rasa. Dobijeni rezultati bi se mogli unaprediti primenom naprednijih arhitektura konvolucionih neuronskih mreža, drugih transformacija slike u sklopu augmentacije podataka i drugih metrika informativnosti. Dodatno, u cilju bolje analize grešaka modela, mogle bi se primeniti i neke od tehnika vizuelizacije modela mašinskog učenja.

7. LITERATURA

- [1] <https://www.dogbuddy.com/> (pristupljeno u oktobru 2022.)
- [2] <https://www.meowtalk.app/> (pristupljeno u oktobru 2022.)
- [3] <https://relaxmycat.com/> (pristupljeno u oktobru 2022.)
- [4] Zisserman, Andrea Vedaldi Andrew, and C. V. Jawahar. "Cats and dogs." IEEE conference on computer vision and pattern recognition. 2012.
- [5] Settles, Burr. "Active Learning Literature Survey." (2009).

Kratka biografija:



Petar Nikolić rođen je u Pirotu 1998. godine. Diplomirao je na Fakultetu tehničkih nauka iz oblasti Softversko inženjerstvo i informacione tehnologije 2021. godine. Master rad iz iste oblasti odbranio je 2022. godine.

**ANALIZA POREMEĆAJA KVALITETA ELEKTRIČNE ENERGIJE KROZ
SIMULACIJU I MERENJA****POWER QUALITY DISTURBANCES ANALYSIS THROUGH SIMULATIONS AND
MEASUREMENTS**Stefan Pajkanović, Marko Vekić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj: U radu su predstavljeni modeli koji simuliraju propade napona usled kratkog spoja, kao i propad napona usled pokretanja asinhronog motora. Pored toga prikazan je uticaj priključenja kondenzatorskih baterija na električnu mrežu. Na kraju rada prikazan je dijagram promjene ukupnog harmonijskog izobličenja po fazama, čije mjerenje je izvršeno na glavnoj crpnoj stanici u Novom Sadu.

Ključne riječi: kvalitet električne energije, analiza poremećaja, transformator, propad napona

Abstract – This paper describes the models that simulate the voltage drops due to a short circuit, as well as the voltage drop due to starting of an asynchronous motor. In addition, the impact of connecting condenser capacitor bank to the electrical network is shown. At the end of the paper, a diagram of the change in total harmonic distortion by phase is shown, which was measured in the main pumping station in Novi Sad.

Keywords: power quality, analysis of disturbances, transformer, voltage drop,

1. UVOD

Pojam kvaliteta električne energije dobija na velikom značaju iz nekoliko razloga, kao što su: povećanje korišćenja osjetljive opreme i uređaja, povećanog korišćenja opreme i potrošača koji dovode do „zagađenja“ električne mreže, povećanog uvezivanja elektroenergetskih sistema, itd. Zbog snažnog međudejstva [1] potrošača i električne mreže (potrošači međusobno trpe, ali i uzrokuju poremećaje), potrebno je utvrditi oblasti ograničenja i standarde kvaliteta električne energije.

Praćenje mjerenja kvaliteta električne energije [2] ima veliki značaj u fazi održavanja jer, u toku rada dolazi do poremećaja koji učestvuju u narušavanju standardnih uslova rada, stoga mjerenjem i nadzorom se omogućava utvrđivanje uzroka i mogućnost pronalaska kvara.

S druge strane, simulacijama [3] je moguće predvidjeti razne scenarije u toku projektovanja nekog sklopa.

NAPOMENA:

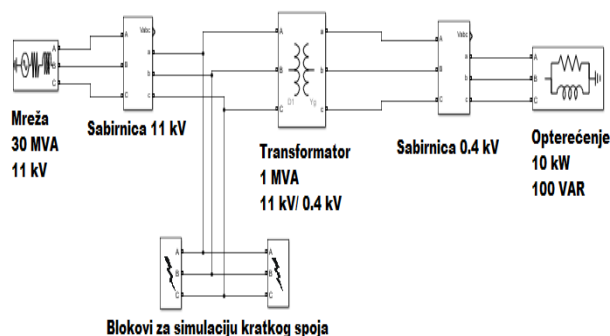
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Marko Vekić, vanr. prof.

2. ANALIZA POREMEĆAJA SIMULACIJAMA

Kreirani su modeli koji simuliraju pojave koje se dešavaju u datom elektroenergetskom sistemu [4] kao što su propadi napona nastali prilikom kratkog spoja, kao i propad napona usled pokretanja asinhronog motora. Za modelovanje i simulaciju modela korišteno je softversko okruženje MATLAB/Simulink -a.

2.1 Uticaj transformatora na propad napona usljed kratkog spoja

Na slici 1 prikazan je model koji simulira propad napona na sekundaru transformatora pri kratkom spoju na njegovom primaru. Posmatrani sistem se sastoji od izvora napajanja nazivne snage 30 MVA, napona 11 kV i frekvencije 50 Hz. Preko transformatora prenosnog odnosa 11 kV/0.4 kV i snage 1 MVA povezano je na mrežu sa niskonaponske strane opterećenje aktivnom snagom 10 kW i reaktivnom snagom 100 VAR. Kratak spoj se simulira preko blokova koji prespajaju fazne provodnike u kojima se simulira kvar.

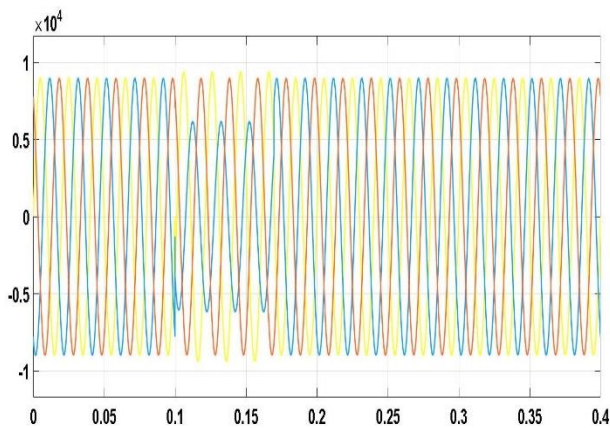


Slika 1. Simulacija propada napona pri kratkom spoju

Na slici 2 prikazan je dvopolan kratak spoj faza A i B sa zemljom na visokonaponskoj strani transformatora.

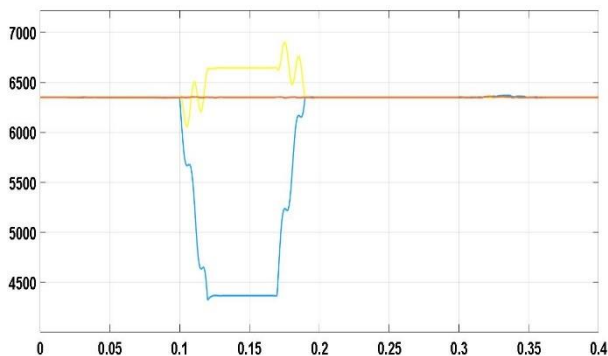
Kratak spoj je simuliran u trenutku od 0.1 do 0.168 sekundi, dok ukupno trajanje simulacije iznosi 0.4 sekunde.

Žutom linijom je predstavljena faza A, plavom faza B i na kraju crvenom faza C. Posmatrajući sliku 2. može se uočiti značajan propad napona faze B pri kratkom spoju, dok je vrijednost faze A nešto iznad nominalne vrijednosti napona na visokonaponskoj strani transformatora. Dok amplituda napona faze C ostaje ista.



Slika 2 Dvopolni kratak spoj sa zemljom između faza A i B na visokonaponskoj strani transformatora

Efekat kvara, odnosno poslednične promjene napona se jasnije uočavaju sa slike 3 na kojoj su prikazane efektivne vrijednosti napona.



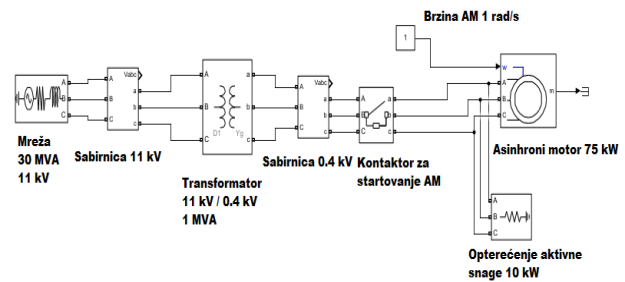
Slika 3. Efektivna vrijednost propada napona usled dvopolnog kratkog spoja na visokonaponskoj strani transformatora

2.2 Propad napona pri pokretanju asinhronog motora

Kada je u pitanju propad napona koji nastaje prilikom pokretanja asinhronog motora, on je posljedica velikih struja koje motor povlači iz mreže, koje su 5 do 7 puta veće od nazivnih vrijednosti.

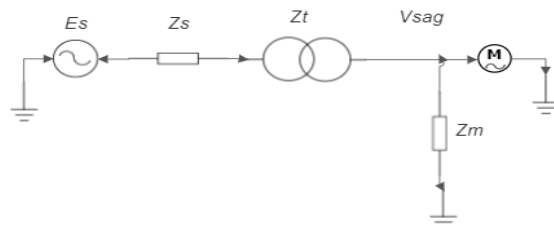
Ovakve struje uključanja obično uzrokuju propad napona dubine do 15 % od nazivnih vrijednosti. Dubina propada napona koja nastaje prilikom pokretanja asinhronog motora zavisi od impedanse mreže. Sa porastom impedanse, padovi napona u mreži su veći, dok dubina propada napona zavisi i od snage asinhronog motora. Dok trajanje propada napona zavisi od inercije motora i njegovog opterećenja.

Na slici 4 prikazan je model koji simulira propad napona koji nastaje prilikom pokretanja asinhronog motora. Model se sastoji od izvora napajanja snage 30 MVA, napona 11 kV, visokonaponske sabirnice 11 kV, transformatora snage 1 MVA i prenosnog odnosa 11 kV/0,4 kV, sabirnice 0,4 kV, kontaktora za startovanje asinhronog motora koji se uključuje u trenutku od 0,08 sekundi, pri čemu je ukupno trajanje simulacije 0,4 sekunde, asinhronog motora snage 75 kW, kao i čisto aktivnog opterećenja snage 10 kW.



Slika 4. Simulacija propada napona pri pokretanju asinhronog motora

U nastavku rada će biti prikazan proračun propada napona asinhronog motora prilikom njegovog pokretanja. Na slici 5 je prikazana ekvivalentna šema datog sistema, koji se sastoji od izvora napajanja, transformatora i trofaznog asinhronog motora. Odgovarajući model iz MATLAB/Simulinka prikazan je na slici 4. U nastavku će biti proračunat propad napona V_{sag} prilikom pokretanja asinhronog motora.



Slika 5. Šema pokretanja asinhronog motora

Propad napona V_{sag} koji nastaje prilikom pokretanja asinhronog motora, se može proračunati na osnovu naponskog razdjelnika, čiji je izraz dat u sledećoj relaciji:

$$V_{sag} = \frac{Z_m}{Z_m + Z_s + Z_t} \cdot E_s = \frac{0,61}{0,61 + 4,03 + 27,5} \cdot 11000 \quad (1)$$

$$V_{sag} = \frac{0,61}{32,14} \cdot 11000 = 207,9 \text{ V}$$

Pri čemu je:

V_{sag} – Propad napona prilikom pokretanja asinhronog motora

Z_m – Impedansa motora.

Z_s – Impedansa izvora napajanja.

Z_t – Impedansa transformatora.

Impedansa motora se računa po sledećoj relaciji:

$$Z_m = \frac{V_n^2}{\beta \cdot S_{motor}} = \frac{400^2}{3,5 \cdot 75 \cdot 10^3} = \frac{121000}{262,500} = 0,61 \, \Omega \quad (2)$$

Pri čemu je:

V_n – Nominalan napon asinhronog motora.

S_{motor} – Snaga asinhronog motora.

β – Vrijednost koja predstavlja koliko je puta struja polaska veća od nominalne.

Impedansa izvora napajanja se računa na osnovu sledeće relacije:

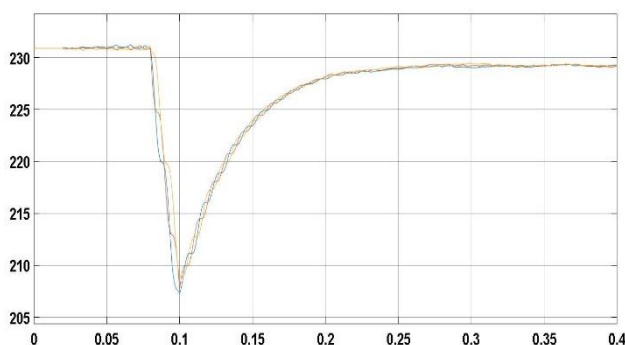
$$Z_s = \frac{V_n^2}{S_s} = \frac{11000^2}{30 \cdot 10^6} = \frac{121 \cdot 10^6}{30 \cdot 10^6} = 4,03 \, \Omega \quad (3)$$

Pri čemu je: S_s – Snaga izvora napajanja.

Impedansa transformatora se računa po sledećoj relaciji:

$$Z_t = \frac{V_{1n}}{V_{2n}} = \frac{11000}{400} = 27,5 \, \Omega \quad (4)$$

Na slici 6 je prikazana efektivna vrijednost propada napona na niskonaponskoj strani transformatora prilikom pokretanja asinhronog motora napajanog sa sekundarne strane u direktnom startu. Može se uočiti da dubina propada napona iznosi oko 10 % od nazivne vrijednosti napona prilikom pokretanja, potom vraćanje vrijednosti napona na nominalnu vrijednost, pri dostizanju nominalne brzine asinhronog motora.



Slika 6. Efektivna vrijednost propada napona na niskonaponskoj strani transformatora 0,4 kV

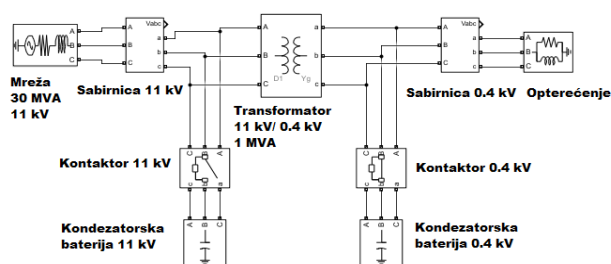
2.3 Uticaj priključenja kondenzatorskih baterija na električnu mrežu

Prilikom priključenja kondezatorske baterije na mrežu, u cilju popravke faktora snage, dolazi do pojave oscilatornih prelaznih pojava. Oscilatorni tranzijent je iznenadna pojava koje ne uzrokuje promjene frekvencije ustaljenog stanja napona, struje ili obje veličine. Jedan oscilatorni tranzijent sastoji se od napona ili struje čija vrijednost brzo mijenja znak. On se opisuje spektralnim sastavom (pretežno frekvencijom), trajanjem i amplitudom.

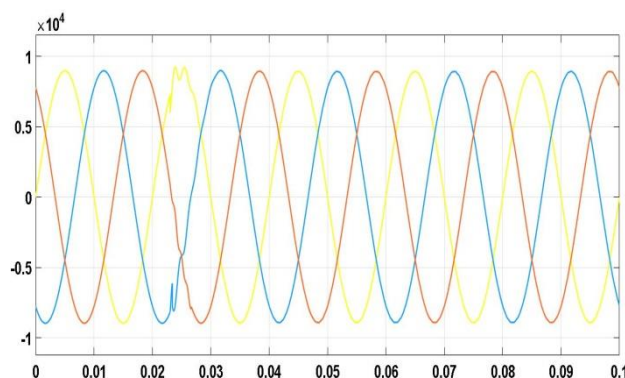
Prilikom priključenja kondezatorske baterije u prenosnu ili distributivnu mrežu dolazi do pojave tranzijenta niske frekvencije, čija je osnovna frekvencija manja od 5 kHz, a trajanje od 0,3 ms do 50 ms. Na slici 7 je prikazan model za simulaciju naponskog tranzijenta usljed priključenja kondezatorskih baterija. Model se sastoji od mreže snage 30 MVA, napona 11 kV, potom visokonaponske sabirnice 11 kV, transformatora prenosnog odnosa 11 kV/0,4 kV, snage 1 MVA, sprege trougao/zvezda, zatim opterećenja na niskonaponskoj strani aktivne snage 100 kW i reaktivne 100 kVAr. Između visokonaponske sabirnice i transformatora je preko kontaktora, paralelno povezana kondezatorska baterija snage 100 kVAr, dok se na niskonaponskoj strani, nalazi kondezatorska baterija manje snage od 40 kVAr.

Na slici 8 prikazana je pojava oscilatornog naponskog tranzijenta na visokonaponskoj strani transformatora, usljed priključenja kondezatorske baterije na niskonaponskoj strani transformatora. Može se uočiti da imamo manju pojavu oscilatornog tranzijenta na visokonaponskoj

strani transformatora, koja je posljedica velike snage izvora napajanja.

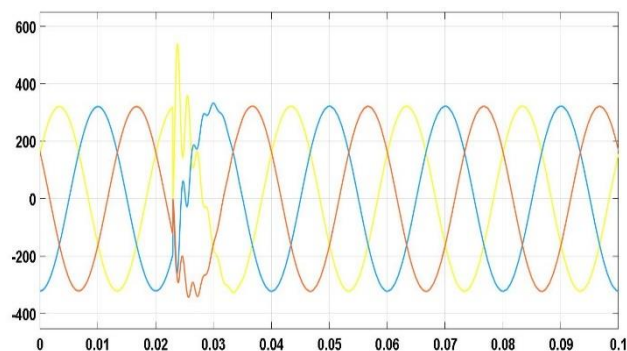


Slika 7. Model za simulaciju naponskih tranzijeneta usljed priključenja kondezatorskih baterija



Slika 8. Pojava naponskog tranzijenta na visokonaponskoj strani 11 kV

Pojave na niskonaponskoj strani daleko su izraženije, što možemo uočiti na slici 9. Oscilatorni naponski tranzijent zavisi od veličine kondezatorske baterije. Kondezatorske baterije veće snage uzrokuju veću pojavu oscilatornog naponskog tranzijenta prilikom priključenja na mrežu. Brzina smirivanja oscilatornih pojava, zavisi od veličine opterećenja aktivnom snagom. Veće opterećenje aktivnom snagom dovodi do većeg faktora prigušenja i samim tim bržeg smirivanja prelaznih pojava samih oscilacija.

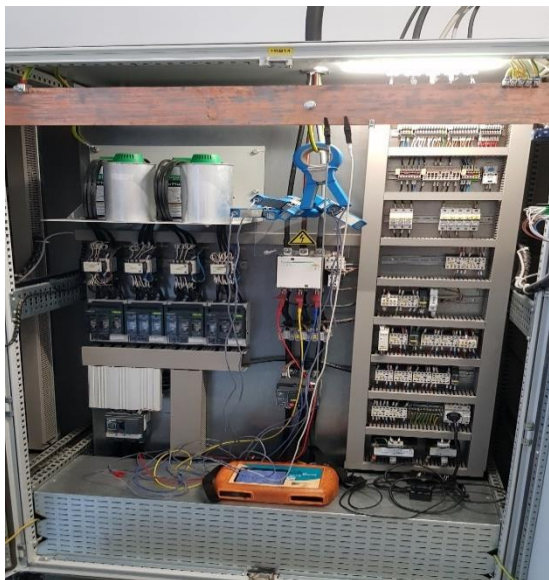


Slika 9. Pojava naponskog tranzijenta na niskonaponskoj strani 0,4 kV

3. ANALIZA POREMEĆAJA MJERENJIMA

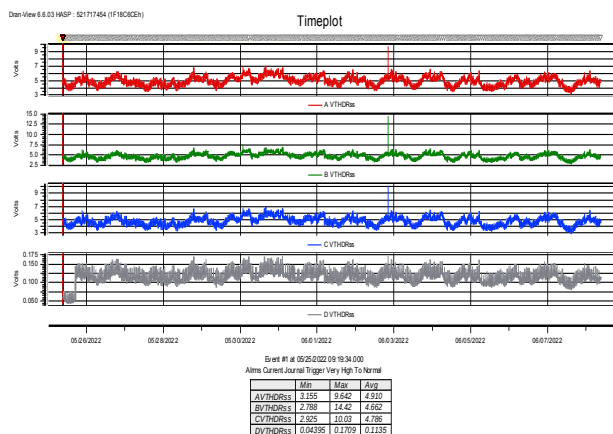
Dana 26.05.2022. izvršeno je mjerenje kvaliteta električne energije na glavnoj crpnoj stanici (NGC) u Novom Sadu, nedaleko od Žeželjevog mosta. Mjerenje je izvršeno na razvodnom ormanu RB14, koji je napajan putem

transformatorske stanice prenosnog odnosa 10/0.4 (KV/KV). Na sabirnicu ovog razvodnog ormara je povezana pumpa C1, asinhroni motor snage 90 KW i 176 A, čija je uloga prečišćavanje otpadnih voda. Mjerenje je izvršeno u trajanju od 12 dana, u periodu od 26.05.2022 do 07.06.2022. Za dobijanje potrebnih rezultata mjerenja korišten je uređaj Dranetz PowerGuide 4400. Na slici 10 prikazano je mjerno mjesto, gdje možemo uočiti strujna klješta koja su povezana na tri faze i neutralni vod: L1, L2, L3 i N, tj. u pitanju je trofazno četverožično mjerenje. Pored toga, povezane su i naponske sonde za mjerenje napona.



Slika 10. Prikaz mjernog mjesta

Na slici 11 je prikazan dijagram promjene ukupnog harmonijskog izobličenja napona po fazama u posmatranom vremenu (Eng. THDU – Total harmonic distortion voltage). Sa dijagrama se vidi da, iako je THD u nekim trenucima značajan, u prosjeku ne prelazi 5 %.



Slika 11. Dijagram promjene ukupnog harmonijskog izobličenja po fazama

4. ZAKLJUČAK

Ovaj rad se bazira na razvoju simulacionih modela u MATLAB/simulink softverskom okruženju. Prvobitno je predstavljen model koji simulira propad napona usljed pokretanja asinhronog motora. Prikazani su proračuni koji reprezentuju propad napona prilikom direktnog pokreta-

nja asinhronog motora, čiji rezultat verodostojno potvrđuju rezultati simulacije. Nakon toga prikazan je model koji simulira propad napona usljed dvopolnog kratkog spoja, kao i model koji simulira naponske tranzijente usljed priključenja kondenzatorskih baterija. Na kraju rada je predstavljena analiza poremećaja kvaliteta električne energije mjerenjima, koje je izvršeno na glavnoj crpnoj stanici u Novom Sadu u trajanju od 10 dana. Prikazano je mjerno mjesto i način povezivanja uređaja za mjerenje. Naposljetku je prikazan dijagram promjene ukupnog harmonijskog izobličenja po fazama.

Rezultati simulacija predstavljaju svojstvene događaje koji se javljaju u elektroenergetskom sistemu, a koji mogu biti od značaja prilikom projektovanja, jer se simulacijama mogu predvidjeti razni scenariji prilikom projektovanja nekog sklopa.

5. LITERATURA

- [1] Allen E, Kosterev DN, Pourbeik P. Validation of power system models. In: 2010 IEEE Power and Energy Society general meeting, Minneapolis, MN. doi:10.1109/PES.2010.5589874.
- [2] Soliman SA, Temraz HK, El-Khodary SM. Power system voltage stability margin identification using local measurements. In: 2003 power engineering conference proceedings. Energy for the future (Cat. No.03EX691). doi:10.1109/LESCPE.2003.1204669.
- [3] Liu Y, Lamine Mili L, De La Ree J, Nuqui RF. State estimation and voltage security monitoring using synchronized phasor measurement. Research paper from work sponsored by American Electric Power, ABB Power T&D Company, and Tennessee Valley Authority, Virginia Polytechnic Institute and State University, 2001-07-12.
- [4] IEC 61000-3-2(2001-10) consolidated edition. Electromagnetic compatibility (EMC) - Part 3-2: Limits for harmonic current emissions.

Kratka biografija:



Stefan Pajkanović rođen je u Bijeljini 11. januara 1996. godine. Srednju elektrotehničku školu završio je u Loparama 2015. god. Iste godine je upisao na Fakultetu tehničkih nauka u Novom Sadu smjer energetika, elektronika i telekomunikacije. Osnovne akademske studije završio je 2021. godine. Iste godine je upisao master akademske studije.



dr Marko Vekić diplomirao je 2005. godine na Fakultetu tehničkih nauka, smjer energetska elektronika i električne mašine. Doktorirao je na Fakultetu tehničkih nauka 2014. godine. Zaposlen je na katedri za energetska elektroniku i električne pretvarače. Oblasti interesovanja su energetska elektronika u distributivnim i pametnim mrežama, kvalitet električne energije, decentralizovane pametne električne mreže.

Kontakt: vekmar@uns.ac.rs

УПРАВЉАЊЕ УСПОСТАВЉАЈУЋИМ ПРЕТВАРАЧИМА ЕНЕРГЕТСКЕ ЕЛЕКТРОНИКЕ У МИКРОМРЕЖИ

CONTROL OF GRID-FORMING POWER ELECTRONIC CONVERTERS IN A MICROGRID

Дејан Јовић, Марко Векић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У овом раду је објашњено управљање успостављајућим претварачима у микромрежи помоћу друп (droop) контроле. Извршене су симулације на моделима са једним и два успостављајућа претварача повезаним са различитим потрошачима. На крају рада су дати одзиви који описују понашање успостављајућих претварача приликом друп (droop) управљања.

Кључне речи: успостављајући претварач, инвертор, друп (droop) контрола.

Abstract – This paper describes the control of grid-forming converters in a microgrid using droop control. Simulations were performed on models with one and two grid-forming converters connected to different loads. At the end of the paper, simulation results were given that describe the behavior of the grid-forming converters during droop control.

Keywords: Grid-forming converters, inverter, droop control.

1. УВОД

Све већа употреба дистрибутивних енергетских извора повлачи нове изазове за стабилан рад електроенергетских система. Термин микромрежа описан је као „скуп потрошача и извора који ради као јединствени управљиви систем који снабдева снагу и топлоту у околину“ [1].

У овом раду посматра се само ток снага између генератора и потрошача. У конвенционалним мрежама, управљање успостављајућим инверторима примењује се при различитим временским скалама, као што су примарна, секундарна и терцијарна контрола [5], [6]. Примарна контрола је унутрашња петља сваког генератора, која осигурава стабилност мреже у стационарном стању и у транзијентима. Циљ секундарне контроле јесте да врати фреквенцију у номиналну радну тачку. Терцијарна контрола регулише ток снага унутар различитих међувеза мреже, нпр. две повезане микромреже.

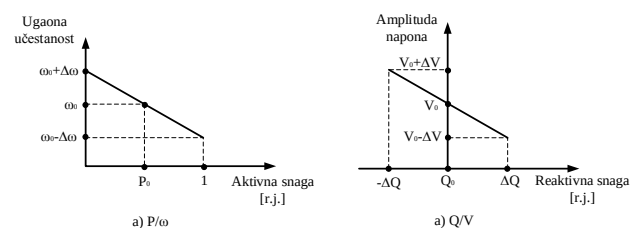
У овом раду, посматра се само примарна и секундарна контрола. Постоје два главна циља управљања инвертора повезаних на мрежу.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Марко Векић, ванр. проф.

Први циљ, код снабдевајућих инвертора, јесте да регулише једносмерни напон на улазу. На овај начин, покушава да се пренесе целокупна снага извора до мреже. Други циљ управљања, код успостављајућем режиму, јесте да се управља излазом напона и фреквенције, чиме се остварује аутономна мрежа. Друп (droop) управљање је окосница модерних електроенергетских система и представља средство за остваривање раздвајања снаге без употребе комуникационих средстава између синхроних генератора.

2. ДРУП (DROOP) КАРАКТЕРИСТИКЕ



Слика 2.1. Друп (droop) карактеристике

Друп (droop) управљање је један од принципа управљања на основу којег се остварује раздвајање снаге у мрежи. При извођењу друп (droop) управљања показано је да је активна снага спрегнута са углом напона, а реактивна снага са амплитудом напона.

-Параметри друп (droop)-а:

У овом раду усвајамо да је појачање k_p једнако:

$$k_p = -0,5 \cdot 10^{-4} \frac{\text{Hz}}{\text{W}} \quad (1)$$

Уважавањем номиналних вредности активне снаге и фреквенције, 10kW и 50Hz редно, и линеарне функције са слике 2.1.а добијамо члан n_p :

$$n_p = 50 + 0,5 \cdot 10^{-4} \cdot 10 = 50,5\text{Hz} \quad (2)$$

На основу горњих параметара добијамо једначину P/ω друп (droop) карактеристике:

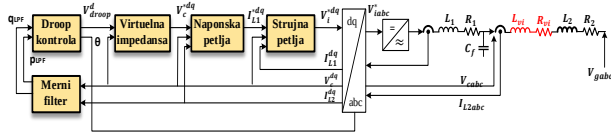
$$f = -0,5 \cdot 10^{-4} \cdot P + 50,5 \quad (3)$$

Аналогно P/ω карактеристици, уважавањем појачања $k_q = 12 \cdot 10^{-3} \frac{\text{V}}{\text{Var}}$ и номиналних вредности реактивне

снаге и напона, $5kVar$ и $400V$ редно, добија се једначина Q/V друп (droop) карактеристике:

$$V = -12 \cdot 10^{-3} \cdot Q + 460 \quad (4)$$

3. УПРАВЉАЊЕ УСПОСТАВЉАЈУЋИМ СКЛОПОМ



Слика 3.1. Шема управљања успостављајућег инвертора са друп (droop) управљањем, LCL филтром и виртуелном импедансом

Главни елементи контролног дела шеме су друп (droop) управљање, напонска петља и струјна петља. Ту су још и мерни филтар и виртуелна импеданса, као и Паркова трансформација. Будући да без обзира на филтар, ипак постоји валовитост напона и струје услед прекидања PWM-а, потребно је убацити и нископропусни мерни филтар у коло. На овај начин се може успоставити микромрежа састављена од једног успостављајућег претварача и потрошача под условом да постоји сталан доток активне снаге.

3.1. LCL филтар

Прорачун параметара LCL филтра се врши у четири корака [1]:

1. Прорачун потребне индуктивности са инверторске стране на основу унапред дефинисане максималне дозвољене вредности валовитости струје.
2. Индуктивност LCL филтра са мрежне стране бира се много мањом од оне са стране претварача
3. Капацитивност кондензатора изабрана је у зависности од избора резонантне учестаности.
4. Пасивни пригушни отпорници се бирају на основу жељеног фактора пригушења.

-Прорачун вредности параметара:

Резонантну фреквенцију бирамо у вредности:

$$10f_g < f_f < \frac{f_s}{10} \quad (5)$$

где је f_g мрежна (50Hz), а f_s прекидачка учестаност (10kHz). Вредност $L_f = L_1 \parallel L_2$ рачунамо на основу једначине за вршну вредност валовитости струе:

$$\Delta I_n = \frac{V_{DC} m_{max}}{8\sqrt{3} f_s L_f} \quad (6)$$

На основу израчунате вредности L_f , и усвајања да је $L_2 = 0.25 \cdot L_1$ [5], могу се израчунати вредности индуктивности са инверторске и мрежне стране. Након што је изабрана резонантна учестаност и

израчуната индуктивност L_f , може се израчунати вредност капацитивности филтра:

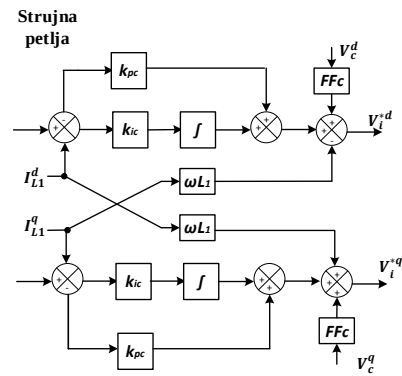
$$C_f = \frac{1}{L_f \omega_f^2} \quad (7)$$

Вредност пригушне отпорности R_d се израчунава на основу:

$$\zeta_p = \frac{R_d}{2} \sqrt{\frac{C_f}{L_f}} \quad (8)$$

3.2. Струјна петља

Струјна петља се састоји од ПИ регулатора у обе осе, садржи спрежне чланове по обе осе и прескочну везу кондензаторских напона v_c^{dq} преко појачања FFc.



Слика 3.2.1. Струјна петља

Два распрегнута линеарна система првог реда су дата:

$$L_1 \frac{di_{L1}^d}{dt} = -R_1 i_{L1}^d + u^d \quad (9)$$

$$L_1 \frac{di_{L1}^q}{dt} = -R_1 i_{L1}^q + u^q \quad (10)$$

Појачање петље у директној грани са ПИ регулатором и пригушником са инверторске стране износи:

$$l(s) = \left(\frac{k_{p,c}}{Ls} \right) \frac{s + k_{i,c}/k_{p,c}}{s + R_1/L_1} \quad (11)$$

Након поништења пола је $l(s) = \frac{k_{p,c}}{Ls}$, а преносна функција затворене петље по d и q осе износи:

$$\frac{i_{L1}^d}{i_{L1}^{*d}} = \frac{1}{\tau_i s + 1} \quad (12)$$

При чему је:

$$k_{p,c} = \frac{L_1}{\tau_i} \quad (13)$$

$$k_{i,c} = \frac{R_1}{\tau_i} \quad (14)$$

3.3. Виртуелна импеданса

Додатно пригушење у колу се постиже убацивањем виртуелне импедансе у коло. Референтне вредности напона на филтарским кондензаторима у том случају изnose:

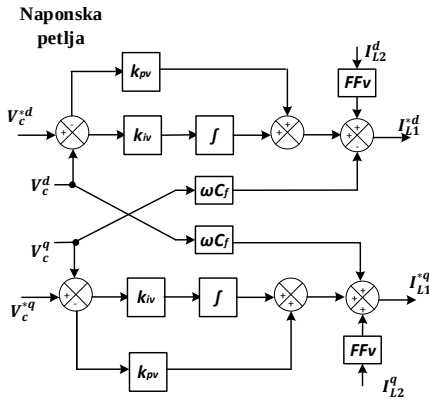
$$v_c^{d*} = v_{droop}^d - r_{vi} i_{L2}^d + \omega l_{vi} i_{L2}^q \quad (15)$$

$$v_c^{q*} = -r_{vi} i_{L2}^q - \omega l_{vi} i_{L2}^d \quad (16)$$

где су r_{vi} и l_{vi} отпорни и индуктивни део виртуелне импедансе представљене у релативним јединицама редом.

3.4. Напонска петља

Циљ алгоритма управљања успостављајућег инвертора је да регулише амплитуду напона и фреквенцију на излазу и да достави потребну активну и реактивну снагу остатку мреже. Циљ управљања напонске петље је да регулише напон на кондензатору C_f .



Слика 3.4.1. Напонска петља

Динамика ове петље је приказана у следећим једначинама у dq домену:

$$C_f \frac{dV_C^d}{dt} = C_f \omega V_C^q + i_{L1}^d - i_{L2}^d \quad (17)$$

$$C_f \frac{dV_C^q}{dt} = -C_f \omega V_C^d + i_{L1}^q - i_{L2}^q \quad (18)$$

Преносна функција ПИ регулатора гласи:

$$G_{PIv}(s) = k_{pv} \frac{s + z}{s} \quad (19)$$

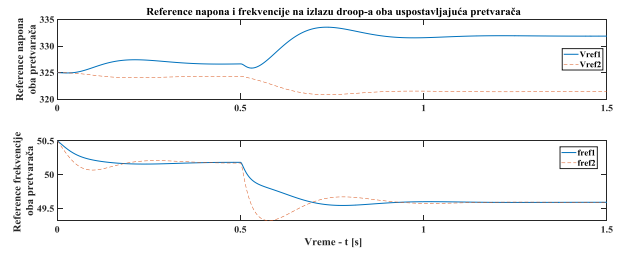
Појачање отворене петље може да се израчуна:

$$l(s) = \frac{k_{pv}}{2T_s C_f} \left(\frac{s + z}{s + T_i^{-1}} \right) \frac{1}{s^2} \quad (20)$$

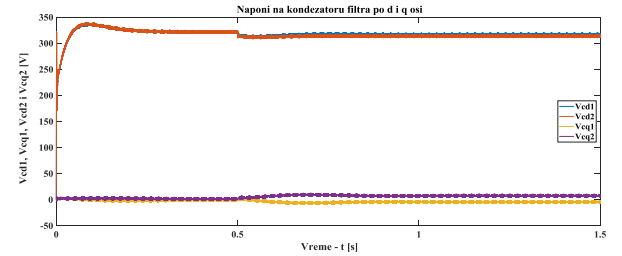
4. РЕЗУЛТАТИ СИМУЛАЦИЈА

У овом раду извршене су симулације у програму MATLAB Simulink на моделима са једним и два успостављајућа склопа у микромрежи. У наставку су дати одзиви симулација са два успостављајућа претварача у микромрежи. На слици 4.1. су дати одзиви референтног напона и фреквенције на излазу друп (droop) контроле оба претварача. На сликама 4.2. и 4.3. су приказани одзиви напона на кондензатору и активне и реактивне снаге при моделу са два успостављајућа склопа, редом. Симулације су започете са претварачима повезаним са само једним потрошачем,

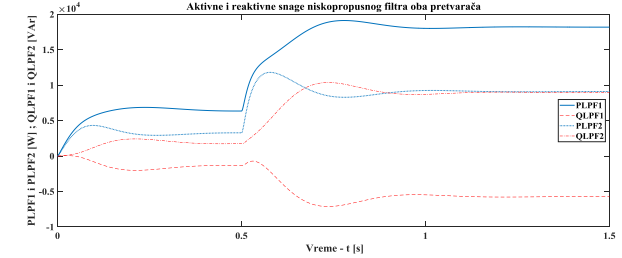
а након неког тренутка је у мрежу убачен и други потрошач.



Слика 4.1. Референце напона и фреквенције на излазу droop-a оба претварача



Слика 4.2. Одзиви напона на кондензаторима филтра два успостављајућа склопа у микромрежи



Слика 4.3. Одзиви активних и реактивних снага два успостављајућа склопа у микромрежи

5. ЗАКЉУЧАК

У овом раду изведен је модел управљачке шеме успостављајућих претварача. То извођење укључује извођење параметара друп (droop) контроле, LCL филтра, струјне петље, напонске петље и виртуелне импедансе. Након извођења модела успостављајућег инвертора, тај модел је искоришћен у софтверском окружењу MATLAB/Simulink да би се анализирано понашање претварача у микромрежи. Демонстриран је утицај друп (droop) управљања и различитих врста оптерећења на рад једног и два претварача повезана у микромрежи.

6. ЛИТЕРАТУРА

- [1] Modeling Techniques and Control Strategies for Inverter Dominated Microgrids – Aris Gkountaras, Universitätsverlag der TU Berlin
- [2] Small-signal stability modelling, sensitivity analysis and optimization of droop controlled inverters in LV microgrids – Simon Eberlein, Krzysztof Rudion, University of Stuttgart, Pfaffenwaldring 47A, 70569 Stuttgart, Germany

- [3] Modelling and analysis of Shunt-connected Voltage Source Converter for voltage dip mitigation – PAOLO CILONA, Göteborg, Sweden 2010
- [4] R. Teodorescu, M. Liserre, and P. Rodriguez, Grid converters for photovoltaic and wind power systems. Wiley Ltd, 2011
- [5] A. Rockhill, M. Liserre, R. Teodorescu, and P. Rodriguez, “Grid-filter design for a multimegawatt medium-voltage voltage-source inverter,” Industrial Electronics, IEEE Transactions on, vol. 58, pp. 1205–1217, April 2011
- [6] L. Malesani, L. Rossetto, P. Tenti, and P. Tomasini, “Ac/dc/ac pwm converter with reduced energy storage in the dc link,” Industry Applications, IEEE Transactions on, vol. 31, pp. 287–292, Mar 1995.
- [7] M. Winkelnkemper, Reduzierung von Zwischenkreiskapazitäten in Frequenzumrichtern für Niederspannungsantriebe. PhD thesis, FG Leistungselektronik, Technische Universität Berlin, 2005. (na nemačkom).
- [8] He J, Li YW. Analysis, design, and implementation of virtual impedance for power electronics interfaced distributed generation. IEEE Trans Ind Appl 2011;47:2525–38
- [9] Wu X, Shen C, Iravani R. Feasible range and optimal value of the virtual impedance for droop-based control of microgrids. IEEE Trans Smart Grid 2017; 8: 1242–51
- [10] Hans F, Schumacher W, Harnefors L. Small-signal modeling of three-phase synchronous reference frame phase-locked loops. IEEE Trans Power Electron 2018;33:5556–60.
- [11] Zou Z, Liserre M. Modeling phase-locked loop-based synchronization in grid-interfaced converters. IEEE Trans Energy Convers 2019. pp. 1–1.
- [12] Nagliero A, Mastromauro RA, Liserre M, Dell’Aquila A. Synchronization techniques for grid connected wind turbines. In: 2009 35th annual conference of IEEE industrial electronics, Nov. 2009. p. 4606–13.
- [13] Midtsund T, Suul JA, Undeland T. Evaluation of current controller performance and stability for voltage source converters connected to a weak grid. In: The 2nd international symposium on power electronics for distributed generation systems, (Hefei, China), IEEE, June 2010. p. 382–8.
- Chung S. Phase-locked loop for grid-connected three-phase power conversion systems. IEE Proc-Electric Power Appl 2000;147:213–9
- [14] Golestan S, Guerrero JM. Conventional synchronous reference frame phase-locked loop is an adaptive complex filter. IEEE Trans Industr Electron 2015;62:1679–82.
- [15] Power Quality in Modern Power Systems – edited by P.Sanjeevikumar, C. Sharmela, Jens Bo Holm-Nielsen, P.Sivaraman
- [16] F. Jenni and D. Wüest, Steuerverfahren für selbstgeführte Stromrichter. vdf Hochschulverlag an der ETH Zürich, 1995. (in German).

Кратка биографија:



Дејан Јовић рођен је у Новом Саду, 17. октобра 1997. године. Средњу школу је завршио у родном граду 2016. године. Исте године је уписао на Факултету техничких наука у Новом Саду, смер Енергетика електроника и телекомуникације. Основне академске студије завршио је 2021. године. контакт: liv.dejan@gmail.com



Др Марко Векић је ванредни професор на Факултету техничких наука у Новом Саду, на Катедри за Енергетску електронику и претвараче. Области интересовања су му енергетска електроника у преносним и дистрибутивним мрежама и квалитет електричне енергије.

контакт: vekmar@uns.ac.rs

OSOBINE I UPOTREBLJIVOST Gin Gonic RADNOG OKVIRA**CHARACTERISTICS AND USABILITY OF THE Gin Gonic FRAMEWORK***Dragiša Simeunović, Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu su detaljno istražene i opisane osobine web aplikacije pisane u Go programskom jeziku pomoću Gin Gonic radnog okvira, razmatrane su prednosti i mane Gin radnog okvira u odnosu na Spring radni okvir programskog jezika Java.

Ključne reči: Gin Gonic, Spring, Golang, Java, web aplikacija

Abstract – The paper analyzes basic characteristics of the web application written in Go programming language using Gin Gonic framework. The advantages and disadvantages of the Gin Gonic framework for Go programming language compared to the Spring framework for the Java programming language are discussed.

Keywords: Gin Gonic, Spring, Golang, Java, web application

1. UVOD

Programski jezik Go, razvijen je od strane kompanije Google 2007. u cilju poboljšanja produktivnosti. Sintaksa i okruženje Go jezika usvajaju obrasce koji su češći u dinamičkim jezicima. Podržani su koncepti kao što su interfejsi, funkcionalnosti za konkurentno programiranje, kanali za tog podataka između procesa – channels kao i rutine – routines [1]. Za razvoj REST API-ja koriste se okviri kao što su Gin Gonic.

Cilj rada jeste istraživanje i opisivanje web aplikacije pisane u Go programskom jeziku upotrebom Gin Gonic radnog okvira i njeno poređenje sa web aplikacijom pisanom pomoću Spring radnog okvira programskog jezika Java.

2. STANJE U OBLASTI

Tri rada na temu Go programskog jezika, opisana su u ovom poglavlju.

2.1. Information System in Go Language

Autor se bavi analizom informacionih sistema kreiranih u Go programskom jeziku. U radu je kreirana web aplikacija pomoću Ravel i Gorp radnih okvira. Opisane su osnovne karakteristike web aplikacije, proces instalacije radnih okvira i prikazani su primeri izvršavanja upita i rada sa rezultatima upita [2].

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Željko Vuković.

2.2. GoAutoBash: Golang-based multi-thread automatic pull-execute framework with GitHub webhooks and queuing strategy

Ideja autora bila je da predstavi radni okvir za testiranje pod nazivom GoAutoBash. Ovaj radni okvir izvršava niz zadataka nakon što klijent pokrene događaj koji pokreće WebHook u VCS-u [3].

2.3. The GO programming language– characteristics and capabilities

U radu, detaljno je opisan tok razvoja Go programskog jezika, navedene su ideje i ciljevi njegovih dizajnera. Autori su se bavili utvrđivanjem prednosti i mana kao i analizom sintakse i semantike ovog jezika [4].

3. WEB APLIKACIJE U GO PROGRAMSKOM JEZIKU

U ovom poglavlju opisane su osnovne karakteristike web aplikacije kao i osobine

3.1. Web aplikacija

Web aplikacija definiše se kao program koji je obično smešten na izdvojenom serveru, koristi se putem interneta uz podršku internet pretraživača.

3.2. Web aplikacija u Go programskom jeziku

Go jezik se pokazao odlično kao programski jezik za pisanje serverskog dela programa koji imaju potrebu da budu brzi. Veoma brzo dobija na popularnosti kao jezik za razvoj web aplikacija. Dobar je za razvoj aplikacija velikih razmera koje obično moraju da budu skalabilne, modularne, lako održive i sa visokim performansama.

3.2.1. Skalabilne web aplikacije u Go

Postoje dva tipa skaliranja, vertikalno i horizontalno. Vertikalno skaliranje je zapravo povećavanje hardverskih resursa na jednoj mašini dok je horizontalno skaliranje povećavanje broja mašina u cilju proširenja kapaciteta [5].

3.2.2. Modularne web aplikacije u Go

Aplikacije treba da omogućavaju dodavanje novih, uklanjanje starih kao i modifikaciju postojećih funkcionalnosti bez narušavanja postojeće implementacije.

3.2.3. Lako održive web aplikacije u Go

Aplikacije imaju stalnu tendenciju za rastom i širenjem. Čest je slučaj izmene određene funkcionalnosti i refaktORIZACIJA postojećeg koda. Samim tim, potrebno je da kod bude što jasniji i organizovaniji.

3.2.4. Web aplikacije visokih performansi u Go

Najbitnija karakteristika web aplikacija sa visokim performansama je to što mogu da prihvate veoma velik broj klijentskih zahteva i da ih obrade u kratkom vremenskom periodu.

3.3. HTTP

HTTP protokol namenjen je prenosu hiperteksta na internetu. Definiše pravila po kojima se informacije prenose između veb servera i veb klijenta.

3.3.1. HTTP zahtevi

HTTP radi po principu zahtev-odgovor. Da bi komunikacija započela, potrebno je kreirati zahtev. Svaki HTTP zahtev mora biti sastavljen od linije zahteva (request line), zaglavlja zahteva, prazne linije i tela poruke (message body).

3.3.2. Metode zahteva

Prva reč navedena prilikom kreiranja zahteva u liniji zahteva je metoda zahteva i predstavlja akciju koja treba da bude izvršena. HTTP metode su GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, CONNECT, PATCH.

3.3.3. HTTP odgovori

Na svaki HTTP zahtev server vraća jedan HTTP odgovor. Svaki HTTP odgovor mora biti sastavljen od statusne linije, zaglavlja odgovora, prazne linije i tela poruke.

3.4. Radni okviri u Go programskom jeziku

Radni okvir dizajniran je u cilju olakšavanja i pružanja podrške programerima u toku razvoja web aplikacije, uključujući web servise kao i web API-je.

3.4.1. Prednosti korištenja radnih okvira u Go programskom jeziku

Statički tipiziran jezik obezbeđuje bolje performanse u toku rada. Svaki program u pisan u Go programskom jeziku mora biti deo nekog paketa. Postoje dva tipa paketa, Executable i Reusable paketi.

Vreme razvoja radnih okvira je brzo i jednostavno. Paketi su već dostupni i mogu se lako importovati.

Konkurentnost je sposobnost programa da poseduje više niti i da onog momenta kada je jedna nit blokirana program izabere neku drugu nit i krene je izvršavati.

3.4.2. Mane korištenja radnih okvira u Go programskom jeziku

Loša strana koja se može uzeti u obzir je upravljanje greškama. Go programski jezik ne poseduje try/catch metode za rukovanje greškama već se greške vraćaju kao regularna povratna vrednost.

3.5. Beego

Beego najveći fokus ima na poslovnim aplikacijama, koje imaju tendenciju da budu velike, sa mnogo koda, koji pokreće mnoge funkcije.

3.6. Echo

Echo se smatra mikro radnim okvirom, koji je više standardna biblioteka i ruter. Ima potpuno pripremljenu dokumentaciju koja mnogo olakšava korišćenje. Podržava HTTP-/2.

3.7. Fiber

Fiber može da se pohvali malom upotrebom memorije i bogatim rutiranjem. Njegova prednost u odnosu na druge radne okvire je ta što je definitivno jedan od najbržih.

3.8. Gin Gonic

Jedan od najpopularnijih radnih okvira. Osim brzine, Gin je veoma jednostavan za upotrebu. Ispraćen je odličnom dokumentacijom i tutorijalima. Gin je open source i dostupan je za korišćenje u bilo koje svrhe.

3.8.1. Rutiranje

Rutiranje je jedna od osnovnih funkcionalnosti koje pruža moderni radni okvir. Bilo kojoj web stranici ili API endpointu pristupa se putem URL-a.

3.8.2. Renderovanje

Web aplikacija može da renderuje odgovor servera u više različitih formata kao što su HTML, tekst, JSON, XML i mnogi drugi.

3.8.3. Middleware

Softverski međuslojevi su zapravo delovi koda koji mogu biti povezani u bilo kojoj fazi rukovanja HTTP zahtevom. Najčešća primena je u enkapsulaciji funkcija koje imaju potrebu biti primenjene više puta.

3.8.4. Primena Gin-a za razvoj web aplikacije

Za prikaz primene Gin-a u toku razvoja web aplikacije kreirana je aplikacija sastavljena od par HTML stranica sa tri endpointa kroz koje je prikazan jednostavan primer GET metode bez parametara, POST metode za dodavanje novog objekta i GET metode sa jednim parametrom.

4. POREĐENJE RADNOG OKVIRA GIN SA RADNIM OKVIROM SPRING BOOT U JAVA JEZIKU

Cilj rada je diskutovanje razlika radnog okvira Gin programskog jezika Go u odnosu na radni okvir Spring programskog jezika Java, kao i prednosti i mane svakog od njih.

4.1. Handlers

U Spring Boot-u pisanje handler-a je jednostavno. Za ulazni parametar koriste se anotacije kao što su RequestBody ili PathVariable. Anotacijama se definišu i tipovi metode zahteva. Odgovor se vraća kroz anotaciju ResponseBody automatski je serijalizovan u JSON i prosleđen u HttpResponseMessage objekat.

U Go jeziku, parametri i povratna vrednost nisu predstavljani u zaglavlju funkcija, sve se dešava unutar tela funkcije pomoću gin.Context.

4.2. Model i ORM

Objektno relaciono mapiranje je mehanizam koji omogućava rad sa klasama i objektima na aplikativnom nivou uz automatski rad sa bazom podataka.

Spring Boot koristi anotacije. Neke od anotacija polja su @Column, @Id, @GeneratedValue. Anotacije klase su @Entity, @Table a za kreiranje veza koriste se @OneToOne, @OneToMany, @ManyToOne i @ManyToMany.

Go jezik i Gin ne podržavaju anotacije. U Gin-u nije potrebno navoditi id entiteta, jer gorm.Model u sebi to već sadrži. Gorm takođe obezbeđuje sve CRUD operacije.

4.3. Dependency injection

Dependency injection je obrazac koji se koristi za implementaciju IoC (eng. Inversion of Control), gde kontrola koja se invertuje postavlja zavisnosti objekta.

U Spring-u, interfejs ApplicationContext predstavlja IoC kontejner. Najčešći način primene dependency injection-a u Springu je putem anotacija. Neke od anotacija koje se koriste su @Autowired, @Profile I @Component.

U Gin random okviru, za upljavljanje profilima ključna tačka je init() funkcija. Profil se dobija pomocu os.Getenv metode.

5. ZAKLJUČAK

U radu su istražene karakteristike web aplikacije pisane u Go programskom jeziku upotrebom Gin radnog okvira.

Go jezik nije vezan za određenu platformu, veoma je dobar za pisanje API-ja i ima dobru podršku za JSON format. Odličan je izbor za pisanje web aplikacija, sastavljenih od velikog broja servisa i baza podataka.

Programski jezik Go veoma je jasan i eksplicitan, korisnik može veoma lako da piše i održava kod.

Zahvaljujući svojim dobrim karakteristikama postaje sve popularniji, u narednih nekoliko godina sigurno će postati jedan od najpopularnijih programskih jezika za pisanje softverskih aplikacija.

6. LITERATURA

- [1] Tony, Go With Me: Go Design Philosophy, Novembar 7, 2021, <https://medium.com/nerd-for-tech/go-with-me-go-design-philosophy-cae7db565fd9>
- [2] Htt Polák, Slavomír. "Information System in.", https://is.muni.cz/th/dg1nk/thesis_polak.pdf
- [3] Bai, Hao. "GoAutoBash: Golang-based multi-thread automatic pull-execute framework with GitHub webhooks and queuing strategy." International Conference on Automation Control, Algorithm, and Intelligent Bionics (ACAIB 2022). Vol. 12253. SPIE, 2022., <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/12253/1225315/GoAutoBash--Golang-based-multi-thread-automatic-pull-execute-framework/10.1117/12.2639536.full?SSO=1>
- [4] Todorova, Magdalena, et al. "The Go Programming Language: Characteristics and Capabilities." Annual of "Informatics" Section Union of Scientists in Bulgaria 6 (2013): 76-85, http://old.usb-bg.org/Bg/Annual_Informatics/2013/SUB-Informatics-2013-6-076-085.pdf
- [5] Sau Sheong Chang, Go web programming, 2016

Kratka biografija:



Dragiša Simeunović rođen je 16.02.1994. u Bijeljini. 2013. godine upisuje „Fakultet tehničkih nauka“ u Novom Sadu, smer „Računarstvo i automatika“. 2017. godine dobija zvanje Diplomirani inženjer elektrotehnike i računarstva. Iste godine upisuje master akademske studije smer „Primenjene računarske nauke i informatika – Elektronsko poslovanje“. kontakt: feddele@gmail.com

IMPLEMENTACIJA PERFORMANTNOG SISTEMA POMOĆU MIKROSERVISIA IMPLEMENTATION OF PERFORMANT SYSTEM USING MICROSERVICES

Stanko Jevtić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – Ovaj rad bavi se opisivanjem *.NET Core* radnog okvira kao i implementacijom performantnog sistema pomoću mikroservisa.

Ključne reči: API, mikroservisi, *.NET Core*, logistika

Abstract – Describing the *.NET Core* framework and implementation of performant system using microservices.

Keywords: API, microservices, *.NET Core*, logistics

1. UVOD

Tema ovog rada jeste da analizira *.NET Core* frejmwork, kao i opis implementacije zadatka koji na frontendu koristi **React** i **Android**, dok na bekendu *.NET 5* u cilju analize performanse sistema.

Zadatak koji je implementiran i opisan u ovom radu uz pomoć pomenutih radnih okvira jeste sistem za opsluživanje paketa i kontejnera. Kada se kaže opsluživanje, tu se misli na kreiranje porudžbina, generisanje labela za te pakete, praćenje statusa paketa, dostavljanja izveštaja u različitim formatima (**EDI214**, **CSV**), skeniranje paketa i njihovo sortiranje unutar centra za sortiranje.

Sort centri su skladišta koja se nalaze u Sjedinjenim Američkim Državama, svrha im je da prihvataju pakete i šalju ih dalje ka Američkoj pošti.

Komunikacija između frontend i backend dela ove aplikacije se vrši preko **HTTP** protokola, koristeći **REST** arhitekturu.

2. *.NET CORE* radni okvir

.NET Core frejmwork [1] je platforma za izvršavanje aplikacija na **Windows**, **macOS** i **Linux** operativnim sistemima.

Glavna namena je implementacija različitih tipova aplikacija kao što su **mobile**, **desktop**, **web**, **cloud**, **IoT**, **machine learning**, **microservice** i **gaming** aplikacije kao što je prikazano na slici 1.

3. SPECIFIKACIJA SISTEMA

Cilj aplikacije jeste logistika i vođenje evidencije o paketima koji pristižu u sort centar, njihovo slanje na sledeću destinaciju, kao i generisanje labela za pakete i kontejnere.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio doc. dr Željko Vuković.



Slika 1. Funkcionalnosti i infrastruktura *.NET Core* frejmworka [2]

Dodatno, potrebno je implementirati portal koji će obaveštavati krajnje korisnike o statusima paketa.

Ideja samog projekta je da se ubrza dostavljanje paketa u Sjedinjenim Američkim Državama korišćenjem pametnog algoritma za sortiranje paketa po zip kodu.

API koji je implementiran integrisan je sa šiperima koji imaju namenu da dostave pakete na adresu krajnjih korisnika. Sama aplikacija je integrisana sa američkom poštom (**USPS** – **United States Postal Service**) kao i **SmartyStreets** provajderom koji omogućava verifikaciju adresa.

3.1. Entiteti

Na slici 2. prikazan je dijagram klasa sistema za upravljanje paketima.

4. IMPLEMENTACIJA BEKEND DELA SISTEMA

U ovoj sekciji će biti opisana implementacija backend dela sistema za opsluživanje paketa, **design pattern**-i korišćeni prilikom formiranja arhitekture, kao i **clean code** prakse koje su primenjene u implementaciji.

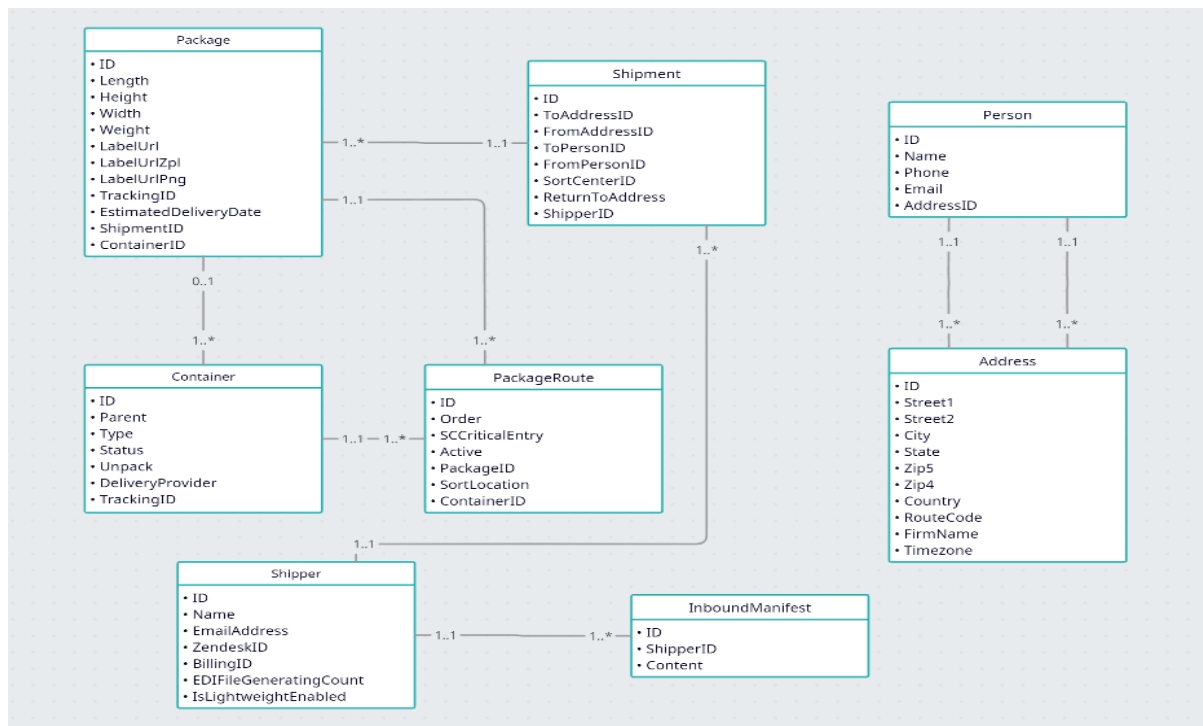
4.1. Arhitektura backend aplikacije

Arhitektura aplikacije predstavlja samu strukturu projekta i biblioteka koje su implementirane i korišćene. Prilikom implementacije sistema za opsluživanje paketa, korišćena je N-slojna arhitektura.

Ona predstavlja razdvajanje logičkih celina kao što su prezentacioni sloj, logički sloj i **data-access** sloj. Ovim pristupom se zadovoljava **design pattern - separation of concerns**.

Slojevi sistema:

- **API** sloj - u ovom sloju se nalaze kontroleri koji su otvoreni spolja i sadrže **endpoint**-e za logovanje, kreiranje i sortiranje paketa, dobavljanje statusa paketa itd.



Slika 2. Klas dijagram entiteta

- **Core** sloj - u ovom sloju se nalazi biznis logika aplikacije, npr. Servisi koji interaktuju sa repozitorijumima koji imaju pristup bazi. Ovaj sloj nema direktan pristup bazi, sa ovim pristupom smo omogućili da sa zahtevom za izmenu **ORM**-a kao što je **Entity Framework** menjamo samo jedan sloj, a sloj koji ima pristup bazi ostane netaknut.
- **Data** sloj - u ovom sloju se nalaze migracije koje se izvršavaju prilikom prve inicijalizacije projekta i koje kreiraju praznu bazu podataka.
- **Domain** sloj - u ovom sloju se nalaze svi modeli tj. entiteti aplikacije. Pristup koji se koristio u ovoj aplikaciji je **code-first**.

Repository sloj - u ovom sloju se nalaze repozirojumi koji imaju pristup samoj bazi podataka. Uz pomoć repozitorijuma je moguće dodati određeni rekord u bazu, dobiti ga, obrisati ili izmeniti.

4.2. Mikroservisna arhitektura

Mikroservisna arhitektura [3] je arhitektonski stil koji se koristi kod modernih aplikacija koje je potrebno skalirati. Neke od glavnih osobina mikroservisne arhitekture su:

- **Highly maintainable and testable** - Olakšano održavanje manjih delova sistema, kao i mogućnost testiranja određenih **unit**-a
- **Independently deployable** - Izmena malog dela aplikacije ne zahteva ponovan **deployment** cele aplikacije
- **Scalability** - Obezbeđuje karakteristike koje doprinose horizontalnoj skalabilnosti odvojenih delova sistema, ukoliko sistem poraste ovo doprinosi uštedi novca za servere potrebne da bi aplikacija radila

- **Separation of concerns** - Odvojeni timovi mogu biti zaduženi za različite delove sistema što doprinosi podeli odgovornosti između timova

U mikroservisnoj arhitekturi, aplikacija je izdvojena na više servisa za razliku od monolitne arhitekture. Svaki mikroservis sadrži svoju poslovnu logiku i u većini slučajeva sadrži i zasebnu bazu podataka. U monolitnoj arhitekturi, nema podele, čime se eliminiše *separation of concerns* svojstvo.

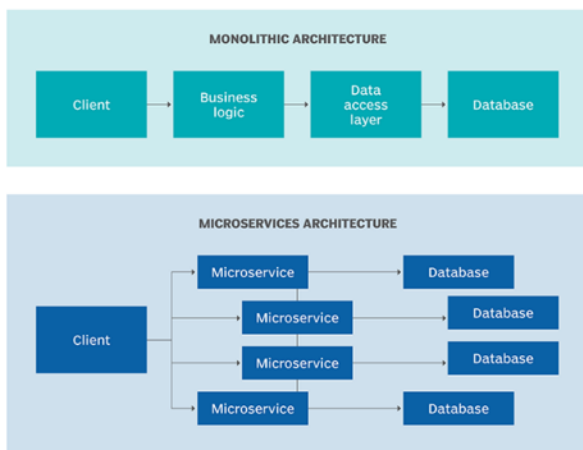
Aplikacije implementirane po ugledu na monolitnu arhitekturu otežavaju testiranje i izmenu softvera, jer problem koji je nastao može da se nalazi bilo gde u sistemu.

Ukoliko je sistem porastao, to može uzeti dosta više vremena u pronalaženju problema. Bilo koja manja promena u sistemu zahteva **deployment** celog sistema. Skaliranje monolitnih aplikacija može da bude teško i skupa za razliku od mikroservisne arhitekture, iz razloga jer je nemoguće skalirati samo odvojene manje delove sistema gde postoji veći **load**.

Primer monolitne i mikroservisne arhitekture prikazan je na slici 3.

U sistemu za opsluživanje paketa koristi se mikroservisna arhitektura. Aplikacija se sastoji iz četiri različita mikroservisa:

- **Autorizaciji mikroservis** – Ovaj servis, kao što sam naziv kaže, ima namenu da autorizuje sve zahteve koji dolaze ka sistemu. Ovaj servis je ujedno i **API Gateway**, sadrži bazu podataka koja ima informacije o validnim korisnicima koji mogu nesmetano da koriste aplikaciju kao i njihovim dozvolama i rolama.



Slika 3. Primer monolitne i mikroservisne arhitekture

- Servis za generisanje labela – Servis čija je namena generisanje labela. Labele je potrebno generisati u **ZPL**, **PNG** i **PDF** formatima.
- Servis za komunikaciju sa **USPS**-om – Servis koji sadrži biznis logiku za kreiranje **outbound manifest**-a, povlačenje statusa paketa od momenta kada **USPS** dostavljač preuzme pakete, kao i svu logiku za kategorizaciju paketa za **USPS**.
- Servis koji opslužuje sve akcije unutar sort centra – Biznis logika same aplikacije, namena je kreiranje paketa, sortiranje, pravljenje kontejnera i slanje kontejnera i paketa do sledeće destinacije.

4.3. Verifikacija adresa

Prilikom kreiranja porudžbine, potrebno je bilo verifikovati da li je adresa krajnjeg korisnika validna kako bi se izbeglo dostavljanje na nepostojeće adrese i kako bi se eliminisao dodatni napor koji bi bio potreban da se započne dostavljanje paketa pa ponovno vraćanje na povratnu adresu. Za ove potrebe postoji više rešenja na tržištu, neki od **provider**-a koji nude verifikaciju adresa u Sjedinjenim Američkim Državama su:

- **USPS Address Verification API** [4]
- **SmartyStreets** [5]

Prvobitno, sistem za opsluživanje paketa se integrisao sa **USPS Address Verification provider**-om. Sama integracija nije bila najjednostavnija jer se podaci šalju preko **XML**-a i sam **API** je zastareo i nije u skladu sa današnjim arhitekturnim konvencijama.

Nakon uspešne integracije, počeli su da se dešavaju određeni izazovi kao što je nedostupnost **USPS Address Verification API**-ja, a u određenim momentima **API** vraća response za 250-300 milisekundi, a ta vrednost neretko ode i na 500 milisekundi.

Nakon susretanja sa ovim problemima, odlučeno je da se pređe na **SmartyStreets provider**-a i da se on koristi za verifikaciju adrese. Međutim, usled situacija gde je i **SmartyStreets** bio nedostupan, krajnja odluka je bila da se koriste oba provajdera za verifikaciju adrese u paraleli.

4.4. Generisanje labela

Za generisanje labela koristio se **ZPL** ili **Zebra programming language** [6]. Zebra uređaji su kompatibilni i često se integrišu sa **android** uređajima, imaju ugrađenu

podršku za automatsko štampanje formata bez ikakve potrebe za dodatnim instalacijama i podešavanjima.

Komande **ZPL** jezika uvek počinju sa jednim od dva znaka, ^ ili ~. Postoji ukupno 170 komandi i svaki format komande mora da počne sa ^XA i da se završi ^XZ. Jedan primer komande je ^AND,n,m gde je n font veličina, dok je m razmak karaktera.

Na slici 4 vidi se jedan primer **USPS** labela izgenerisane **ZPL** komandama.



Slika 4. Primer izgleda labela za pakete

Alat koji se koristi za konverziju **ZPL** formata u **PDF** i **PNG** naziva se **Labelary**. **Labelary** je **online viewer** koji omogućava okruženje gde se može testirati i pogledati izgled labela izgenerisane pomoću određenih **ZPL** komandi.

Labelary je besplatan, nudi **public API** koji može da koristi svako ko ima pristup internetu, ali **public API** nije namenjen za produkciju, pa samim tim **Labelary** nudi **offline** verziju koja se koristi lokalno, čime se eliminiše i kašnjenje.

5. ZAKLJUČAK

U okviru ovog rada predstavljen je **.NET Core**, jedan od najperspektivnijih frejmworka sadašnjice. **Microsoft** sa svakom verzijom poboljšava skup funkcionalnosti i ubrzava performanse samog frejmworka. Od kako je **.NET Core** zamenio **.NET Framework**, ovaj frejmwork ima veliku primenu u praksi, s obzirom da je **cross-platform**. Iz tog razloga je i odabran taj frejmwork za implementaciju sistema za opsluživanje paketa.

Sam sistem je performantan i veoma lako ga je migrirati na novije verzije **.NET Core**-a koje će izaći u budućnosti, to je još jedna odlična odlika ovog radnog okvira. Neke od stvari kojima bi sistem za opsluživanje paketa mogao biti unapređen jeste ubrzavanje poziva ka bazi podataka korišćenjem čistih **SQL query**-a umesto **ORM**-a.

Ovo je izazov sa kojim se susreću svi veći sistemi koji dođu u ovu fazu. Na početku je **Entity Framework** bio najbrže rešenje za efikasnost prilikom implementacije, međutim to može da bude dug koji kasnije dolazi na

naplatu. Primer **ORM**-a koji je performantniji od **Entity Framework**-a je **Dapper**.

6. LITERATURA

- [1] <https://dotnet.microsoft.com>
- [2] <https://agileviet.vn/whats-new-in-net-5>
- [3] <https://www.techtarget.com/searchapparchitecture>
- [4] <https://www.usps.com/business/web-tools-apis>
- [5] <https://www.smarty.com/products/single-address>
- [6] <https://www.zebra.com>

Kratka biografija:



Stanko Jevtić rođen je u Novom Sadu 1994. god. Završio je tehničku školu „9. Maj“ u Bačkoj Palanci 2013. godine. Osnovne akademske studije na Fakultetu tehničkih nauka u Novom Sadu je završio 2017. Master akademske studije na Fakultetu tehničkih nauka u Novom Sadu upisao je 2017. godine.

kontakt: stanjevtic@gmail.com

MODELOVANJE TEMA U TEKSTU NA OSNOVU NASLOVA DOKUMENATA**TOPIC MODELING IN TEXT BASED ON TITLES OF DOCUMENTS**Minja Lepar, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu je predstavljen pristup za modelovanje tema i klasifikaciju tekstualnih dokumenata. Konkretno, vršena je 1) primena LDA (Latent Dirichlet Allocation) nad tekstom zarad dobijanja tema, pri čemu je evaluacija rađena kvalitativno, kroz semantiku pronađenih tema; 2) klasifikacija dokumenta primenom reprezentacije teksta dobijene kombinacijom tf-idf obeležja i tema izvučenih pomoću LSA (Latent Semantic Analysis); nad ovom reprezentacijom treniran je Naive Bayes klasifikator, a evaluacija je vršena računanjem F-mere, 3) klasifikacija dokumenta primenom tf-idf reprezentacije teksta, gde je eksperimentisano sa treniranjem SVM (Support Vector Machines) i RF (Random Forest) modela; I u ovom slučaju evaluacija je vršena računanjem F-mere.

Ključne reči: modelovanje tema, LDA, klasifikacija, SVM, Naive Bayes, Random Forest

Abstract – The paper presents an approach for topic modeling and document classification. Concretely, the paper explores 1) the application of LDA (Latent Dirichlet Allocation) to obtain topics from the text; This approach was evaluated qualitatively, relying on the semantics of the found topics. 2) document classification, where documents were represented using tf-idf features and topics extracted by applying LSA (Latent Semantic Analysis); Naive Bayes classifier was trained on the obtained representation, and F-measure was used for evaluation, 3) document classification, where the tf-idf representation was used to train a classification model, where we experimented with using SVM (Support Vector Machines) and RF (Random Forest) models; In this case, F-measure was used for evaluation.

Keywords: Topic modeling, LDA, classification, SVM, Naive Bayes, Random Forest

1. UVOD

Potreba za organizacijom teksta, sve je veća i veća. Internet je preplavljen tekstom u elektronskoj formi. Informacija se potapa u nestruktuiranom formatu, u smislu da niko ne zna da je tu, i niko ne zna šta da radi s njom, te prisutna informacija biva neiskorišćena. Za potrebe ljudskog napretka, funkcionisanja i kvaliteta života, skrivena informacija može biti neprocenljiv izvor mnogih koristi, inovacija i razvoja.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Jelena Slivka, vanr. prof.

Analiza teksta je tehnika koja se koristi za izvlačenje skrivene informacije iz teksta u nestruktuiranoj formi [4][6][7][8]. Zadaci modelovanja teksta su:

- organizacija i filtriranje vesti [8],
- rezimiranje dokumenata [1][5][6][7],
- praćenje zdravstvene nege [8],
- primena modelovanja tema u geografiji za ekstrahovanje tema iz geografskih dokumenata [4],
- primena modelovanja tema u političkim naukama za identifikovanje tema u političkim govorima [4], itd.

Modelovanje tema otkriva skrivene teme u tekstu [4]. Modelovanje tema može da unapredi postupak automatske klasifikacije dokumenata time što, umesto da tekst reprezentujemo putem pojedinačnih reči, reprezentujemo ga kroz ekstrahovane teme [8]. Ovim postupkom se u znatnoj meri redukuje dimenzionalnost reprezentacije teksta. U ovom cilju, rešenja primenjuju LSA (Latent Semantic Analysis) za modelovanje tema, a kao modele mašinskog učenja (*machine learning*, ML) tipično primenjuju SVM (Support Vector Machines) i Naive Bayes (NB) modele [8]. U ovom radu isproban je opisani postupak, uz dodatnu primenu Random Forest (RF) modela koji se pokazao kao jedan od superiornijih modela za klasifikaciju teksta zbog svojih odlika u klasifikaciji, odabiru atributa i dodavanja težina [9].

Cilj ovog rada je:

1. Pronalaženje tema u skupu ulaznih tekstova primenom nenadgledanog učenja, odnosno, bez korišćenja anotiranih tema dostupnih u korišćenom skupu podataka.
2. Korišćenje anotiranih tema u skupu podataka radi klasifikacije dokumenata po temi.

U radu je korišćen je skup podataka dostupan na Kaggle stranici [10]. Kao ulaz u opisani postupak koristi se sirovi tekst, konkretno, naslov iz kolekcije naslova dokumenata dostupnih u skupu podataka. Dokumenti su u skupu podataka anotirani pripadnošću jednom od osam tema ('TECHNOLOGY', 'HEALTH', 'WORLD', 'ENTERTAINMENT', 'SPORTS', 'BUSINESS', 'NATION', 'SCIENCE'), a cilj ovog rada je da se obučim model koji, na osnovu naslova dokumenta, automatski može da zaključi kojoj temi dokument pripada.

U cilju 1. zadatka (pronalaženja tema), u radu se primenjuje LDA (Latent Dirichlet Allocation), nenadgledani model. Eksperimentisano je sa odabirom broja tema, gde su isprobane vrednosti 8, što odgovara broju tema u skupu podataka, 10 i 12. U slučaju 8 tema,

model je evaluiran poređenjem rezultujućih tema sa anotiranim temama. U slučaju 10 i 12 tema, model je evaluiran kvalitativno. Kvalitativnom evaluacijom zaključeno je da se dobijaju semantički kvalitetne teme. Za veći broj tema dobijaju se specifičnije i rafiniranije teme. Međutim, u svim slučajevima je primećeno određeno preklapanje tema. Treba naglasiti da je rezultate teško evaluirati bez uključivanja domenskog eksperta jer se, na primer, u slučaju teme zdravlja javljaju podoblasti poput patologije, genetike i sl. koje zahtevaju dublje poznavanje domena.

U cilju 2. zadatka (klasifikacija dokumenata), trenirani su nadgledani modeli za klasifikaciju u postojeće teme. Izvršeni su sledeći eksperimenti:

- a) naslov se pretvara u vektor obeležja (reči naslova) *tf-idf* (*term frequency - inverse document frequency*) transformacijom. Dalje, nad *tf-idf* reprezentacijom se primenjuje LSA radi ekstrakcije tema. Nad tako dobijenom reprezentacijom se primenjuje NB model za klasifikaciju u jednu od osam predefinisanih tema.
- b) naslov se pretvara u vektor obeležja (reči naslova) *tf-idf* transformacijom. Zatim se primenjuje nadgledani model, gde je eksperimentisano sa: SVM i RF.

Cilj ovog 2. eksperimenta je da se utvrdi koliko primena LSA modela kao koraka pretprocesiranja doprinosi klasifikaciji.

Radi evaluacije oba zadatka, skup podataka je podeljen na trening i test skup. Za potrebe 1. zadatka za test skup je uzeto 2 300 primera, a model je evaluiran kvalitativno. Za potrebe 2. zadatka, skup podataka je podeljen 2:1, a kao glavna metrika evaluacije je korišćena *F*-mera. Kao najbolji model pokazao se SVM, čija *F*-mera iznosi 81%.

U literaturi ne postoji rad koji je primenio modelovanje tema i klasifikaciju teksta baš na skupu podataka [10]. Tipično se koristi skup podataka *Reuters-21578* i najbolje performanse u literaturi za ovaj skup podataka su iz rada koji vršio klasifikaciju dokumenata nad nižedimenzionom reprezentacijom reči indukovanom od LDA (uz 50 tema) i primenio SVM model [5]. Ovaj postupak postiže i do 97% tačnosti. U radu je pokazano da ovaj pristup značajno unapređuje performanse u odnosu na nižedimenzionu LDA reprezentaciju. Shodno tome, rad [5] predlaže tematsko-baziranu LDA reprezentaciju, kao filtrirajući algoritam za odabir reči u klasifikaciji teksta. *Reuters-21578* skup podataka korišćen u [5] sadrži 8 000 dokumenata i 15 818 reči, gde broj klasa nije naveden. To čini manji broj dokumenata, ali približno isti broj reči, u odnosu na skup [10] korišćen u ovom radu.

Ovaj rad je organizovan na sledeći način. Poglavlje 2 prikazuje prethodne radove i koncepte na koje se oslanja ovaj rad. Poglavlje 3 daje teorijske osnove primenjenog metoda. Poglavlje 4 definiše metodologiju, a poglavlje 5 opisuje eksperiment i ukazuje na hiperparametre koji su upotrebljeni u modelima. U poglavlju 6 dati su rezultati, dok poglavlje 7 analizira greške modela. Poglavlje 8 zaključuje rad.

2. PRETHODNI RADOVI

U ovom poglavlju predstavljeni su srodna rešenja, vezana za koncepte koji se implementiraju u ovom radu.

2.1. Modelovanje tema

U radu [3] evaluira se primena LDA za preporuku tagova (oznaka), imajući resurse i dodeljene tagove. U radu [5] LDA je poređen sa drugim modelima. U radu je opisan LDA model i kako se vrši njegova primena nad skupom podataka.

U radu [6] predstavljen je pristup klasifikaciji dokumenata, gde su poređeni sledeći pristupi za modelovanje tema: treniranje Naive Bayes modela, LSA i LDA modelovanje tema. Pokazalo se da LSA i LDA postižu znatno bolje performanse pri modelovanju tema.

2.2. Klasifikacija dokumenata po temama sa i bez primene LSA

U radovima [1] i [2] opisana je primena LSA modela, što je analiza koja se primenjuje i u ovom radu.

U radu [4] pomenute su ekstenzije LDA pristupa koje pokazuju bolje performanse. U radu koji će se pisati, primenila se ekstenzija LSA na model NB.

U radu [7] opisana je strategija za klasifikaciju tekstualnih dokumenata. Navedeni su osnovni problemi vezani za odabir atributa iz ogromnog broja atributa, i odabir ML tehnika za klasifikaciju teksta. U ovom radu, primenjen je *tf-idf* statistički metod za uticaj na važnije tokene i postizanje boljih rezultata, sa ML tehnikom SVM za klasifikaciju dokumenata.

U radu [8] je primenjen pristup klasifikacije dokumenata primenom NB modela uz redukciju dimenzionalnosti reprezentacije teksta primenom LSA. Metodom je pokazano da tematsko kategorisanje dokumenata može dati tačnije klasifikovanje dokumenata u predefinisane teme. LSA je rezultirala redukcijom dimenzionalnosti. Ista ova ideja primenjena je u ovom radu.

U radu [9] opisan je RF model, njegove prednosti i nedostaci i razne sugestije uz koje se tehnika može poboljšati. Ovaj rad primeniće RF za klasifikaciju, pri čemu je rad [9] koristan za razumevanje i dublje analiziranje RF-a.

3. TEORIJSKE OSNOVE

LDA je generativni probabilistički model koji dokumente reprezentuje kao slučajne mešavine nad latentnim temama, gde je svaka tema karakterizovana kao distribucija verovatnoća nad rečima [5]. Reči sa najvećim verovatnoćama daju ideju o tome šta je tema [4].

SVM pronalazi hiper-ravan koja razdvaja klase, sa maksimalnom marginom (hiper-ravan koja je najdalja od svih tačaka skupa podataka).

NB je probabilistički klasifikator koji primenjuje *Bayes-ovu* teoremu [8]. Pretpostavka modela je da je u dokumentu verovatnoća pojavljivanja svake reči nezavisna od pojavljivanja ostalih reči [8]. Redukcija dimenzionalnosti se implementira primenom SVD (*Singular Value Decomposition*) [1]. Primenom LSA, procesirane reči separatišu se u značajne grupe [6].

RF je ansambl model koji koristi stablo odlučivanja kao bazni klasifikator koji određuje klasnu oznaku instance koja nema oznaku [9].

Većinskim glasanjem svaki klasifikator daje jedan glas (predviđa klasnu labelu), a labela sa najviše glasova klasifikuje instancu [9].

4. METODOLOGIJA

4.1. Eksplorativna analiza

Primenom eksplorativne analize ovde je analiziran skup podataka [10]. Zaključeno je da skup podataka ne sadrži nedostajuće vrednosti. Postoje duplikati naslova, ali su oni zadržani jer pripadaju različitim temama. Skup ima 111 444 naslova i osam tema kojima ti naslovi mogu pripasti.

U skupu podataka postoje dve kolone: tekst kolona, koja sadrži naslove dokumenata i služi kao ulaz u modele prikazane u ovom radu i kolona sa temama koja služi kao ciljno obeležje. Jedinstvene vrednosti ciljnog obeležja su 'NATION', 'SPORTS', 'ENTERTAINMENT', 'BUSINESS', 'WORLD', 'TECHNOLOGY', 'HEALTH' i 'SCIENCE' i one su relativno balansirane u skupu podataka.

Kao priprema podataka za vektorizaciju teksta i pronalaženje rečnika, izvršena je tokenizacija i lematizacija (ekstrakcija gramatičkog korena reči). Takođe su uklonjene stop reči (skup reči koje će biti isključene iz rečnika).

4.2. Latent Dirichlet Allocation

U ovom poglavlju je opisana primena LDA modela za ekstrakciju tema. Dostupni podaci se dele na trening i test skup. Naslovi dokumenata se reprezentuju putem *count vectorizer*-a (svaki element vektora je broj pojave određene reči u dokumentu). Nad ovom reprezentacijom treniran je LDA model. Isprobano je više varijanti LDA modela, gde je kao broj tema (n) korišćeno 8, 10 i 12 tema. Osam tema je izabrano u skladu sa brojem anotiranih tema u skupu podataka, ali je isprobano i 10 i 12 tema u cilju ispitivanja da li će ovo rezultovati čistijom podelom na teme.

Nakon treniranja LDA modela, vizualizovane su naučene teme da bi se utvrdila njihova semantika. LDA transformiše trening i test skup, da bi se dobila raspodela tema za trening i test skup (2 300 naslova). Predefinisane kategorije skupa podataka mapiraju se u naučene kategorije LDA metoda za trening i test skup. Poređenjem trening i test skupa dobijaju se slični rezultati. Nad test skupom, poređenjem stvarne kategorije sa naučenom kategorijom se može vršiti analiza grešaka modela. Za sva tri modela kvalitet se ocenjuje semantikom pronađenih tema (kvalitativna evaluacija).

4.3. Klasifikacija

Da bi se izvršila klasifikacija dokumenata, najpre se vrši priprema podataka opisana u potpoglavljju 4.1. Konvertor oznaka konvertuje oznake iz tekstualne forme u numeričku. Za sve algoritme isti su koraci, 'counts' (*CountVectorizer()*), 'tfidf' (*TfidfTransformer()*), i primena klasifikatora. U slučaju NB-a i LSA proces iza 'tfidf' je 'SVD' (*TruncatedSVD()*) i 'normalize' (*Normalizer()*). LSA je kombinacija SVD i *Normalizer*-a, prvi redukuje

veličinu matrice, drugi normalizuje vrednosti u SVD matrici.

4.4. Korišćeni alati

U cilju implementacije postupka prikazanog u radu, korišćene su sledeće biblioteke i alati: *pandas*¹ (za manipulaciju skupom podataka), *sklearn*² (priprema podataka, konstrukcija modela, korišćenje metrika), *pyLDAvis*³ i *wordcloud*⁴ (vizualizacija tema), *numpy*⁵ (manipulacija nizova), *matplotlib*⁶ (vizualizacija uz grafikone), *seaborn*⁷ (vizualizacija uz *heatmap*-e, toplotne mape), *spacy*⁸, *nltk*⁹ i *gensim*¹⁰ (manipulacija teksta), *Anaconda*¹¹ (manipulacija *Jupyter Notebook*-a).

5. EKSPERIMENT

U ovom poglavlju opisano je kako su određeni hiperparametri za sve eksperimente i kako je vršena evaluacija rešenja.

5.1. Latent Dirichlet Allocation

Određuju se LDA hiperparametri, kako je broj tema u skupu podataka osam, ovde se vrši treniranje za osam tema, ali se vrši i eksperiment za 10 i 12 tema.

5.2. Klasifikacija

Određuju se hiperparametri za klasifikacione modele. SVM model radi linearno odvajanje klasa, odnosno, *kernel* (jezgro) je *linear*. Za NB i redukciju dimenzionalnosti *TruncatedSVD()* ima parametar 100 tema, prema preporuci dokumentacije. Za RF nisu definisani hiper-parametri, već su korišćene njihove predefinisane vrednosti.

5.3. Evaluacija

LDA model ocenjen je kvalitativno. Kroz rečničke reprezentacije naučenih tema, posmatra se semantika tema, i uviđa se kvalitet. Veličina skupa za testiranje postavljena je na 2 300 naslova.

Klasifikacioni modeli evaluirani su *F*-merom, koja je glavna metrika evaluacije. Za sve klasifikacione modele urađen je jedan način podele na trening i test skup i on iznosi 2:1.

6. REZULTATI

Na slici 1 prikazan je rezultat primene LDA za 8 tema. Slika prikazuje vizualizaciju raspodele verovatnoća tema *Topic 1 – 8* za svaku od stvarnih tema (*topic*). Za sve odabire broja tema, naime $n = 8, 10, 12$, vizualizacije bez referenciranja istinske kategorije rezultiraju u semantički kvalitetne teme.

¹<https://pandas.pydata.org>

²<https://scikit-learn.org/stable>

³<https://pyldavis.readthedocs.io>

⁴<https://pypi.org/project/wordcloud/>

⁵<https://numpy.org>

⁶<https://matplotlib.org/stable>

⁷<https://seaborn.pydata.org>

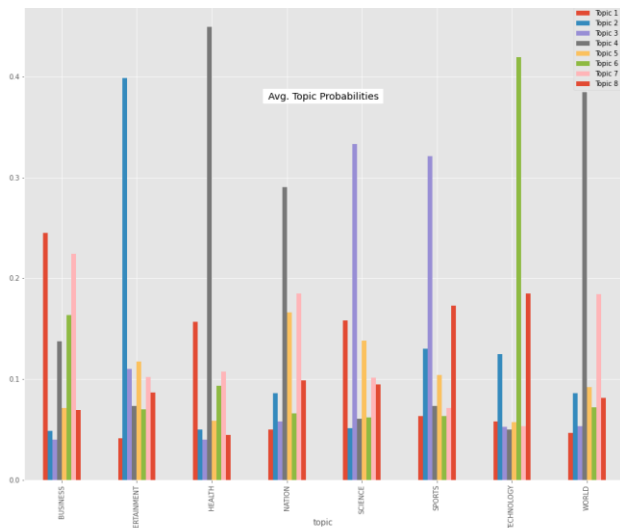
⁸<https://spacy.io/api/doc>

⁹<https://www.nltk.org>

¹⁰<https://pypi.org/project/gensim/>

¹¹<https://docs.anaconda.com>

Izvršena je analiza grešaka LDA modela za 8 tema, gde znamo stvarnu kategoriju na osnovu skupa podataka. Za



10 i 12 tema ova analiza nije vršena. Razlog pogrešno klasifikovanih naslova je prisustvo reči koje su indikativne o nekoj naučnoj kategoriji koja nije istinska kategorija naslova.

Slika 1. Vizualizacija raspodele verovatnoća tema Topic 1 – 8 za svaku od stvarnih tema (topic)

Kod klasifikacije dokumenata, SVM model je postigao F -meru od 81%, RF je postigao F -meru od 75%, a NB uz LSA pretprocesiranje 52%. podataka. Utvrđeno je da LAS kao korak pretprocesiranja nije doprineo poboljšanju performansi klasifikacije dokumenata na ovom skupu podataka.

Izvršena je analiza pogrešno klasifikovanih naslova. Utvrđeno je da do grešaka dolazi jer su naslovi jesu previše kratki. Greške su se javljale i kod naslova koji sadrže reči koje su pripale nekoj temi, a koje ne postoje u temi koja je za taj naslov anotirana u skupu

8. ZAKLJUČAK

U radu su predstavljeni metodi za automatsku klasifikaciju tekstualnih dokumenata na osnovu njihovih naslova. Korišćen je skup podataka [10].

Prva grupa metoda modelovala je teme u tekstu nenadgledano, odnosno, bez korišćenja anotacija tema dostupnih u skupu podataka. Toj grupi metoda pripada LDA koji je treniran za 8, 10, i 12 tema. Modeli su ocenjeni kvalitativno i utvrđeno je da su dobijene semantički kvalitetne teme, iako postoje određena preklapanja. Sa porastom broja tema raste, teme postaju specifičnije za poddomene. Nad test skupom objašnjeno je pogrešno klasifikovanje naslova na osnovu reči iz kojih se naslov sastoji, poređenjem sa istinskom temom naslova. Predlaže se da je za detaljniju analizu tema potreban domenski ekspert.

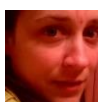
Drugoj grupi metoda pripadaju metodi za klasifikaciju koji koriste anotirane teme naslova. Najvišu F -meru od 81% postigao je SVM model treniran nad $tf-idf$ obeležjima. Pokazalo se da LSA kao korak pretprocesiranja nije doprineo poboljšanju performansi klasifikacije.

Predlog za unapređivanje modela je da se pronade optimalan broj tema koji će jasno definisati teme, koje se ne preklapaju. Primena LDA umesto LSA modela, kao i implementacija boljih transformera podataka bi mogla poboljšati postupak ekstrakcije tema.

9. LITERATURA

- [1] Evangelopoulos, N., Zhang, X. and Prybutok, V.R., 2012. Latent semantic analysis: five methodological recommendations. *European Journal of Information Systems*, 21(1), pp.70-86.
- [2] Wiemer-Hastings, P., Wiemer-Hastings, K. and Graesser, A., 2004, November. Latent semantic analysis. In *Proceedings of the 16th international joint conference on Artificial intelligence* (pp. 1-14).
- [3] Krestel, R., Fankhauser, P. and Nejdl, W., 2009. Latent dirichlet allocation for tag recommendation. In *Proceedings of the third ACM conference on Recommender systems* (pp. 61-68).
- [4] Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y. and Zhao, L., 2019. Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, 78(11), pp.15169-15211.
- [5] Blei, D.M., Ng, A.Y. and Jordan, M.I., 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan), pp.993-1022.
- [6] Rajasundari, T., Subathra, P. and Kumar, P.N., 2017. Performance analysis of topic modeling algorithms for news articles. *Journal of Advanced Research in Dynamical and Control Systems*, 11, pp.175-183.
- [7] Dalal, M.K. and Zaveri, M.A., 2011. Automatic text classification: a technical review. *International Journal of Computer Applications*, 28(2), pp.37-40.
- [8] Sedghpour, A.S. and Sedghpour, M.R.S., 2020. Web Document Categorization Using Naive Bayes Classifier and Latent Semantic Analysis. *arXiv preprint arXiv:2006.01715*.
- [9] Fawagreh, K., Gaber, M.M. and Elyan, E., 2014. Random forests: from early developments to recent advancements. *Systems Science & Control Engineering: An Open Access Journal*, 2(1), pp.602-609.
- [10] Abdelsalam, K. topic_balanced_dataset, Version 1. Retrieved June 13, 2021 from <https://www.kaggle.com/karimamd95/topic-balanced-dataset/version/1>.

Kratka biografija:



Minja Lepar rođena je 1987. god. u Odžacima. Osnovne akademske studije završila je na Prirodno-matematičkom fakultetu, odseku za primenjenu matematiku, smer Inženjer matematike. Master rad na Fakultetu tehničkih nauka iz oblasti Softverskog inženjerstva i informacionih tehnologija – Inteligentni sistemi brani 2022.god. kontakt: minja.lepar@gmail.com

РАЗВИЈАЊЕ МУЛТИПЛАТФОРМСКЕ ВИДЕО ИГРЕ СА ПОДРШКОМ ЗА ИГРАЊЕ ПРЕКО МРЕЖЕ**DEVELOPING MULTIPLATFORM VIDEO GAME THAT CONTAINS SUPPORT TO PLAY ONLINE**

Александар Комазец, Предраг Теодоровић, *Факултет техничких наука, Нови Сад*

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У овом чланку је описана реализација и функционисање мултиплатформске игре са подршком за играње преко мреже, као и независне серверске апликације. Омогућено је покретање апликације на оперативним системима Windows 10 OS, Linux OS, као и Android OS.

Кључне речи: Libgdx, Node, Box2D, MongoDB, Mongoose

Abstract – This article describes realization and functioning of multiplatform video game that contains support to play online. Here is provided to run the application on operating systems like Windows 10 OS, Linux OS, and Android OS.

Keywords: Libgdx, Node, Box2D, MongoDB, Mongoose

1. УВОД

У овом раду ће бити представљена структура онлајн платформерске видео игре као и сам развој видео игре и сви коришћени алати.

Игра се састоји од две независне компоненте (Клијент и сервер). Клијент је развијан коришћењем библиотеке које припадају радном оквиру Libgdx [1], док је серверска апликација развијана помоћу Node runtime окружења.

2. АЛАТИ, ЕКСТЕРНЕ БИБЛИОТЕКЕ, РАДНИ ОКВИРИ

Као што је већ поменуто у уводу, игра се састоји од клијента, сервера и базе података, тако да ће сви коришћени алати и библиотеке бити груписани у три групе.

2.1 Клијент**2.1.1 “Libgdx” радни оквир**

“Libgdx” радни оквир поседује подршку за развијање видео игара на различитим платформама и системима попут Windows, Mac, Linux, Android, iOS, and HTML5.

2.1.2 “Box2D” енџин

Box2D је енџин (енг. *engine*) за симулирање физике у дводимензионалном простору. Box2D енџин програ

мерима пружа апликативни програмерски интерфејс помоћу кога је могуће креирати свет са реалистичном физиком.

2.1.3 “Ashley” библиотечка

Ashley је библиотека заснована на ECS [2] (енг. *Entity Component system*) архитектуралном шаблону који је се обично креира искључиво при развоју видео игара. ECS софтверски шаблон се састоји из компоненти (То су модули који представљају контејнере за податке), ентитета који представља скуп компоненти, као и система (Систем представља модул који је задужен за једну функционалност, на пример рендеровање слике или детекцију контакта између тела)

2.2 Сервер

Сервер развијан за потребе видео игре писан је у програмском језику JavaScript.

2.2.1 “Node.js”

Node.js је асинхроно run-time окружење које се заснива на V8 [3] енџину који је имплементиран у претраживач “Google Chrome” и служи за преводјење JavaScript кода у машински код, тако да је могуће поред веб апликација развијати и системске апликације користећи главни JavaScript фајл из Node пројекта. У овом случају, Node је кориштен за развијање серверске апликације.

2.2.2 “Express.js”

Express.js [4] представља радни оквир базиран на већ постојећем Node.js. Express.js проширује могућности постојећег веб сервер модула који је састави део Node-а.

2.2.3 “Socket.io-server”

Socket.io [5] представља библиотеку за бидирекциону комуникацију између клијента и сервера засновану на догађајима користећи WebSockets протокол.

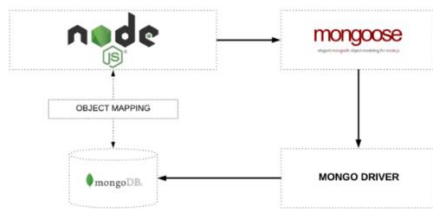
2.2.4 “Mongoose” и “MongoDB”

MongoDB [6] је NoSQL база података докумената без шеме. Подаци који се чувају у MongoDB бази података су представљени у JSON формату, а структура ових докумената може варирати на начин да се један објекат у бази састоји од свих понуђених поља док други објекат не садржи сва понуђена поља, већ искључиво обавезна поља.

Ово је једна од предности коришћења NoSQL-а јер убрзава развој апликација и смањује сложеност имплементације.

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Предраг Теодоровић, доцент.



Слика 1. Мапирање објеката између Node-а и MongoDB помоћу Mongoose-а (преузето из [7])

Mongoose [7] је библиотека за моделовање података објеката (ODM) за MongoDB и Node.js. Управља односима између података, пружа проверу валидности шеме и користи се за превођење између објеката у коду и представљање тих објеката у MongoDB.

3. РАЗВОЈ 2D ОНЛАЈН ПЛАТФОРМЕР ВИДЕО ИГРЕ

Назив видео игре на којој се заснива овај рад је “Brick it”. Назив игре је настао тако што се комбинују енглеска речи “Brick” што значи цигла и енглеска реч “Break” што значи поломити.

Цео ниво је израђен од квадратних циглица различитих облика које могу бити уништене од стране играча. Играч има мотивацију да уништава циглице како би пронашао различите награде попут блага или различитих супер моћи.

Игра има могућност самосталног играња, као и могућност играња преко интернета са другим играчима у кооперативном моду са још једним играчем или у “Player vs Player” моду, где су играчи противници.

Игра такође подржава више платформи, то јесте може се играти на персоналном рачунару или Андроид телефону.

Игра се састоји из три велика система:

- Административни систем који обухвата обраду података који су неопходни за почетак меча (Све класе неопходне за кориснички интерфејс, делегатори екрана и тако даље);
- Играчки (енг. *Gameplay*) систем који обухвата обраду свих података кад меч почне;
- Мрежни систем који обухвата компоненте одговорне за слање и примање података преко мреже.

3.1 Административни систем

Административни систем обухвата све компоненте од кад играч покрене апликацију па до момента када почиње меч.

Административни систем се састоји од:

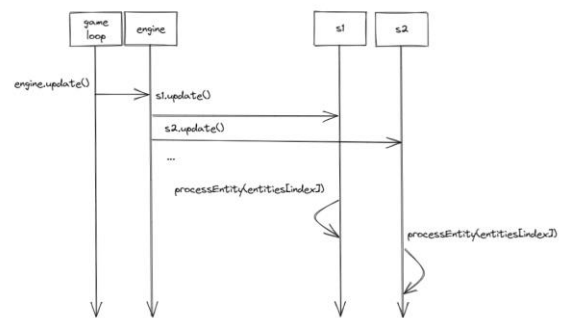
- Специјализованих платформских покретача (енг. *Android Launcher* и *Desktop Launcher*);
- Game класе (*MyGdxGame*);
- Група корисничких екрана (енг. *Screens*);
- Менаџера ресурса (енг. *Asset Management System*);
- Систем за испис порука (енг. *Logger*);
- Конфигурационе класе (*GameConfig*).

3.2 Играчки (енг. *Gameplay*) систем

Играчки систем сачињавају две велике групе:

•Модули који се налазе у склопу ECS-а, попут ECS компоненти које представљају контејнере за податке као и ECS системи који представљају специјализоване функционалности где су сви системи смештени у један ред и синхронно се извршавају (слика 2.) приликом сваке итерације главне играчке петље;

•Сви остали модули који нису део ECS групе попут *MatchTracker* класе, *BodyCreator*, *TileMapHandler*, *InputController* класа и осталих. То су функционални модули који пружају одређени сервис по позиву или асинхронно.

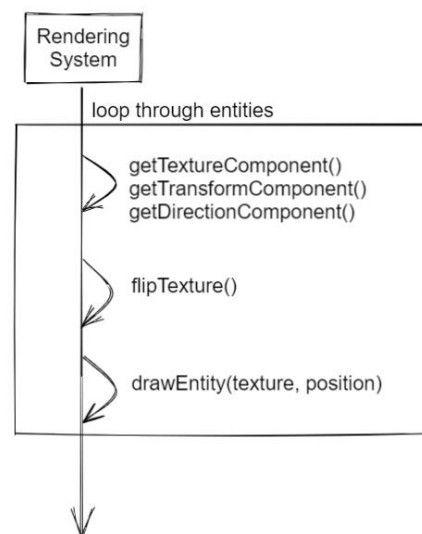


Слика 2. Итерација енџин-а кроз системе

На слици 3 налази се пример једног ECS система под називом “Rendering” систем. Рендеринг систем оперише само са ентитетима које је потребно приказати, то јесте који имају текстуру као што је играч или противник.

Рендеринг систем захтева и позицију где се ентитет налази на мапи, а ту информацију добија из *Transform* компоненте.

Direction компонента садржи информацију о смеру ентитета онда када је потребно заокренути текстуру. Метода за исцртавање прима текстуру и позицију на којој треба да је исцрта. Процедура се понавља за сваки релевантан ентитет.



Слика 3. Функционалност Рендлинг система

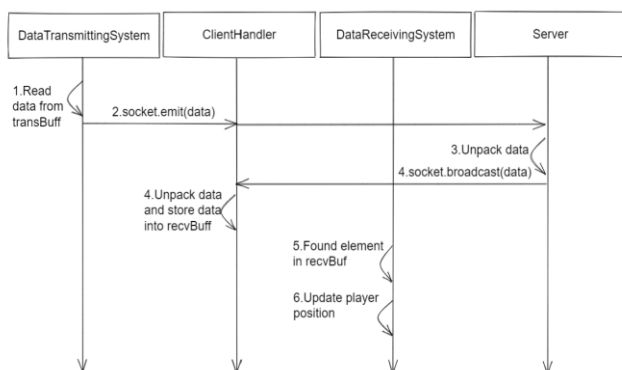
3.3 Мрежни систем

Мрежни систем видео игре је задужен за слање и примање података преко мреже онда када играч учествује у онлине мечу. ClientHandler је модул који је задужен за слање и примање порука.

3.3.1 Пример комуникације клијент - сервер (Ажурирање позиције играча)

У случају онлајн меча где учествују два играча, сваки клијент у свом свету види себе као локалног играча и преостале играче (У овом случају једног играча) као онлајн играча. Сваки играч је дужан да након промене позиције на мапи пошаље своју позицију серверу и да сервер проследи позицију тог играча преосталим играчима. У наставку ће бити описан процес ажурирања позиције у коме учествују два играча са јединственим ИД-јевима (0 и 1). Претпоставимо да играч 0 промени позицију (слика 4.).

1. Затим DataTransmittingSystem чита податке из бафера за слање података и прави пакет за слање тако што поред података о позицији пакује и socket ИД као и назив догађаја.
2. Пакет података се емитује користећи socket-e (утичнице) из ClientHandler-a;
3. Сервер региструје нови догађај, распакује податке, пакује их и прослеђује податке свим осталим клијентима (Играчу 1) користећи догађај под именом "updatePlayerInputPositionResp";
4. ClientHandler играча 1 прихвати пристигли догађај, распакује податке, пакује их и смешта у бафер за примљене поруке;
5. DataReceivingSystem играча 1 проверава да ли има садржаја у баферу за примљене поруке;
6. Чита елемент из бафера и ажурира позицију на којој се налази играч 0.



Слика 4. Ажурирање позиције играча

3.4 Серверски систем

Серверска компонента је развијена коришћењем Node runtime окружења. Као веб сервер је кориштен Http сервер користећи Express.js радно окружење. За бидирекциону комуникацију се користе утичнице из библиотеке socket.io. База података која је кориштена је MongoDB. Mongoose библиотека се користи како би се једноставно дефинисале колекције података као и једноставније коришћење упита.

Такође треба имати на уму да је Node предвиђен за споре операције попут ИО/мрежних операција као на пример слање и примање пакета као читање и писање са екстерног диска и да није намењен за захтевне процесорске операције тако да серверска апликација углавном служи са прослеђивање порука између клијената.

3.4.1 Компоненте серверског система

Сервер се састоји из две компоненте:

- Главни модул која представља улазну тачку за серверску апликацију. Има улогу да се при иницијализацији конектује са MongoDB сервером, иницијализује се http сервер (подеси се ip адреса и порт на којем сервер треба да слуша), и врши иницијализацију TrafficHandler модула. Одговоран је за регистрацију и пријаву клијената као и за слање упита бази података;
- TrafficHandler модул је задужен за комуникацију са клијентима као и за праћење стања соба и играча током меча.

4. ЗАКЉУЧАК

На основу развоја видео игре у овом пројекту, може се закључити да је неопходно познавање више области. За самостални развој видео игре треба бити упознат са технологијама које су актуелне, зато што постојеће технологије знатно убрзавају развој. Бројне елементарне операције (детекција кретања, исцртавање текстура на екрану, ефикасно учитавање ресурса и тако даље) су имплементирани тако да девелопер може да се посвети развијању механике саме игре.

Libgdx радни оквир пружа много могућности при развоју 2D видео игара, док Node и MongoDB пружају много могућности при развијању серверске апликације.

Архитектура апликације мора бити добро осмишљена како би развијање карактеристика (енг. *Features*) апликације било што једноставније. Добро установљена архитектура доноси униформност при развијању апликације, тако да нови развојни инжењер на пројекту треба да се упозна са шаблонима који се користе и да их користи при даљем развоју апликације. ECS шаблон поред високе ефикасности пружа и униформни развој. Девелопери који развијају нова својства игре користећи ECS шаблон треба да логику новог својства представе ECS системом, а податке представе ECS компонентом.

Такође треба напоменути да се до развоја квалитетног софтвера долази кроз вишеструке итерације.

5. ЛИТЕРАТУРА

- [1] Sanchez Cejas Alberto, Marquez Saltares David, "Libgdx Crossplatform Game Development Cookbook", Packt Publishing, 2014
- [2] Härkönen Toni, "Advantages and implementation of entity-component-systems", Faculty of Information Technology and Communication Sciences, Tampere, 2019
- [3] V8.dev, Septembar 2022. [online]. Available: <https://v8.dev/>

- [4] Expressjs.com, Septembar 2022. [online]. Available: <https://expressjs.com/>
- [5] Socket.io, Septembar 2022. [online]. Available: <https://socket.io/docs/v4/>
- [6] Mongodb.com, Septembar 2022. [online]. Available: <https://www.mongodb.com/>
- [7] Freecodecamp.org, Septembar 2022. [online]. Available: <https://www.freecodecamp.org/news/introduction-to-mongoose-for-mongodb-d2a7aa593c57/>

Кратка биографија:



Александар Комазец рођен је у Врбасу 1995. год. Дипломски рад на Факултету техничких наука из области Електротехника и рачунарство – Ембедед системи и алгоритми одбранио је 2020.год.
Области интересовања су му развој видео игара коришћењем C++ и C# језика.

контакт:
aleksandarkomazec5@gmail.com

СОФТВЕРСКО РЕШЕЊЕ ИНТЕРНЕТ ПРОДАВНИЦЕ

A SOFTWARE SOLUTION OF A WEB SHOP

Катарина Нинковић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У овом чланку описана је имплементација и рад интернет продавнице. Реч је о трослојном систему који укључује имплементацију сервера и клијента, као и креирање и управљање базом података. Пројекат представља приступ развијања модерних web апликација уз последње технологије и програмска окружења као што су Node, React, Redux и MongoDB. Подржана је евиденција купаца и порубина, динамичко прикупљање производа и осталих информација, администраторски простор за управљање сајтом и друге функционалности које покрива савремена интернет продавница.

Кључне речи: Web, Node, React, Redux, MongoDB

Abstract – This article describes implementation and functionality of a web shop. It is about a three-layer system that includes the implementation of a server and a client, as well as the creation and management of a database. Project presents the approach to development of modern web application including latest technologies such as Node, React, Redux and MongoDB. It supports notation of customers and orders, dynamical loading of products and rest of the information, admin space for site management and other functionalities that web shops cover nowadays.

Keywords: Web, Node, React, Redux, MongoDB

1. УВОД

У данашње време процес *online* куповине или куповине преко интернета постао је општеприхваћен и заступљен у модерном друштву. Динамика живота данашњег човека често оставља без могућности да пролази кроз многобројне радње у потрази за оним што му је потребно, што поред напорног обилажења, укључује и велик губитак слободног времена, те овакав вид куповине представља олакшавајућу околност. Путем интернет продавнице (енг. *WebShop*) корисник је у могућности да прегледа све производе, одабере оне одговарајуће и “пошаље их у корпу”, одакле ће даље моћи да изврши куповину.

У оквиру овог пројекта реализован је систем који се састоји од два основна блока. Први блок представља, серверску страну, колоквијално названу *Back-end*,

која укључује серверску апликацију базирану на Node.js [1] платформи и њој специфичном JavaScript програмском језику, и базу података, која је у овом случају MongoDB [2]. Други блок обухвата клијентску страну, илити *Front-end* који садржи React.js [3] кориснички интерфејс, који је такође базиран на JavaScript програмском језику, али и на HTML и CSS језицима за креирање и дизајнирање web страница. Поред тога на “фронту” се користи *Local Storage* за складиштење података у претраживачу, те Redux [4] који контролише стања чувана у локалној бази података и PayPal као *third party* платни систем.

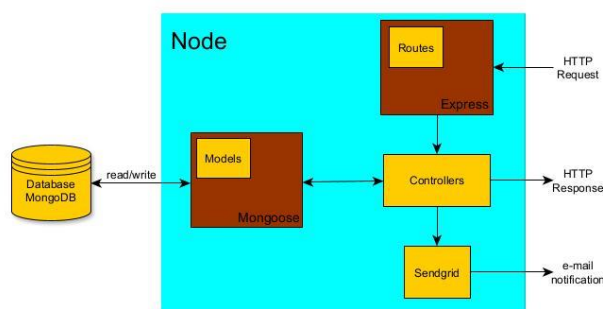
2. СИСТЕМ

У овом поглављу наведени су кључни аспекти и модули система и описани са теоријским освртом.

2.1 Серверска страна

Сервер је сачињен од следећих ентитета:

- Node.js - за покретање сервера
- MongoDB - за управљање базом података
- Express.js - за рутирање
- Sendgrid - за слање e-mail порука
- Контролера - за обраду захтева



Слика 1. Блок шема серверске стране

Node.js

Node.js представља JavaScript платформу базирану на Google V8 Engine-у, која пружа једноставан и брз модел програмирања, а уз то подржава и асинхрони рад web апликације. Захтеве обрађује у само једном процесу, без потребе за креирањем додатне нити (енг. *Thread*) услед нових захтева. Захваљујући томе омогућен је бржи процес јер не постоји потреба за чекањем извршења захтева, већ апликација наставља даље са извршавањем наредне линије кода.

Уз Node долази и Node Package Manager - *npm*. У питању је алат који је задужен за добављање екстерних Node.js пакета, као и за складиштење и

НАПОМЕНА:

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

проналажење истих. Инсталација нових пакета обавља се помоћу *npm* команди командне линије (терминала), чиме се пакети аутоматски преузимају и додају у листу зависности (енг. *Dependency*) у датотеку *package* са *JSON* екстензијом. Пакети који су коришћени на *back-end*-у су:

- *Mongoose* - за рад са *MongoDB*
- *Dotenv* - за учитавање променљивих из *env* датотеке у процесу
- *Express* - за рутирање сервера и обраду *HTTP* захтева
- *BCrypts* - за енкриптовање шифре
- *JSONWebToken* - за генерисање токена
- *Path* - за добијање *jpg* и *png*
- *Multer* - за учитавање датотека
- *Morgan* - за приказ *HTTP* захтеva ради анализе
- *Sendgrid* - за слање *e-mail* порука ка корисницима

MongoDB

Node поседује пакет (библиотеку) која омогућава приступање бази и извршавање различитих упита. Реч је о пакету који се назива *Mongoose* и пакету који поседује веома једноставан интерфејс за моделирање података објеката (енг. *Object Data Modeling - ODM*). Користи се као спрега између објеката у самом коду и њихову репрезентацију у бази. *MongoDB* представља нерелациону врсту базе података или *NoSQL*, односно документациону базу, у оквиру које податке је могуће складиштити као *JSON* документе. Структура докумената може да варира, услед чега је смањена сложеност примене, а самим тим и бржи развој саме апликације.

Express.js

Express је *Node* пакет који представља уграђену библиотеку чије основно задужење јесте руковођење *HTTP* захтевима са клијентске стране. Такође, *Express* пружа комуникацију између сервера и базе података, где серверска страна тражи од модела извршавање операција од базе. Коришћене су *GET*, *POST*, *PUT* и *DELETE* методе.

Након што сервер пошаље захтев или добије исти, потребно је проследити одговор. Да би било лакше разазнати врсте захтева и упита, креирају се путање или руте (енг. *Routes*) које су креиране за одређену крајњу тачку (енг. *Endpoint*). Изворна (енг. *Root*) тачка у фази развоја јесте <http://localhost:5000/api>. Оваква путања могућа је само за машину у локалној мрежи, док би за остале уређаје у мрежи на рути уместо *localhost* кључне речи била прослеђена конкретна *IP* адреса уређаја на којем се сервер заправо извршава. У продукцији би крајња тачка била назив *web* странице, која би била *host*-ована. Руте служе за прослеђивање захтева одговарајућим контролерима, који даље обављају сву потребну функционалност.

Контролери

Контролери функционишу тако што преко модела шаљу захтеве ка бази, креирају одређене упите у зависности од тога за шта су им потребни ти подаци и уколико испуњавају услове онда као одговор добијају жељене податке.

Уколико би се узео пример пријаве (енг. *Sign in*) корисника на *web* страници, први корак би био унос података (*e-mail* адреса и лозинка) на клијентској страни, након чега следи паковање унетих података у *HTTP* захтев ка рути на серверској страни <http://localhost:5000/api/users/login> који сервер преузима и потом прослеђује одговарајућем контролеру. Контролер даље за улогу има проверу да ли је корисник евидентиран у бази тако што шаље захтев ка бази, тражећи да прочита податке и изврши одређена испитивања као што је у овом случају постојање и поклапање унете *e-mail* адресе и шифре. Уколико корисник постоји као одговор од базе добијамо податке везане за тог корисника.

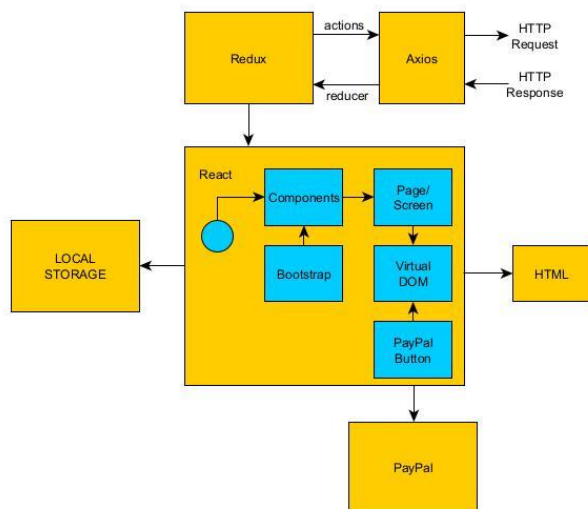
Sendgrid

У питању је екстерни (*third-party*) систем за реализацију слања *e-mail* порука ка корисницима од стране сервера. Ова платформа поседује подршку за *Node*, те се може увући и користити као остали пакети. Неопходно је креирати *developer* налог и *API* кључ како би слање порука било могуће. *Sendgrid* је једна од платформи која изузетно брине о безбедности, те је потребно испоштовати неке од њихових процедура и гарантовати доставу сигурног садржаја, а њихова техничка подршка редовно надгледа рад налога и сигурност *API* кључа. Поред примаоца, у заглављу поруке, потребно је проследити *e-mail* адресу пошиљаоца, коју је претходно потребно верификовати.

2.2 Клијентска страна

За развој корисничког интерфејса коришћена је *React* библиотека са својом *ES6 JavaScript* верзијом. Као и на *back-end* страни, за креирање пројекта и инсталацију пакета треће стране користи се *npm*. Кључни ентитети на *front-end* делу су:

- *React - JavaScript front-end framework*
- *Redux* - за контролу локалне базе података
- *Bootstrap* - за изглед корисничког интерфејса
- *PayPal* - систем за плаћање



Слика 2. Блок шема клијентске стране

React

Ова платформа функционише тако што се за сваки *HTML* елемент креира специјални *JavaScript* објекат

уз чију се помоћ врши контрола над *DOM* моделом сајта. Објекти могу бити елементи или компоненте (сложени елементи) и сваки од њих може укључити неке друге чиме се образује структура “стабла” у којем сваки чвор представља један објекат. Тако се креира виртуални *DOM* - идентична копија реалног модела, чија је улога упоређивање и ажурирање истог, који се користи за рендеровање странице. Уколико би дошло до промене стања неког од објеката виртуални *DOM* ће заправо ажурирати само њега, док остале неће дирати, за разлику од реалног модела који би ажурирао цело стабло чиме представља далеко спорију имплементацију.

React елементи се користе за описивање изгледа *HTML* елемената. За њихов опис користи се *JavaScript* синтаксно проширење *JSX* (енг. *JavaScript Syntax Extension*) који обезбеђује писање *HTML* структуре и *JavaScript* кода у истој датотеци. За управљање тим елементима користе се компоненте.

Redux

Redux је библиотека која се користи за управљање стањима. Стања из *Redux*-а могуће је само читати (енг. *Read-only*), док се за њихову промену и манипулацију користе акције и редуктори (енг. *Reducer*). Акција описује жељене измене, док редуктори те измене извршавају. Редукторске методе се не позивају директно него уз помоћ специјалне функције *dispatch* из *Redux* библиотеке.

Bootstrap

Bootstrap је *framework* који је задужен за естетику, тј. за изглед корисничког интерфејса. Представља колекцију *CSS* класа које су задужене за унапред дефинисан и стандардизован изглед *HTML* елемената или, у овом случају, *React* компонената. Поред лепог изгледа, пружа и додатне функционалности кроз уграђени *jQuery*, али једна од најбитнијих одлика јесте одзивност страница, односно скалиран и контролисан приказ елемената на различитим резолуцијама и величинама екрана. Поседује и *npm* пакет чиме поседује и готове *React* компоненте.

Pay Pal

Америчка компанија *Pay Pal* поседује дугу историју пословања у свету, један је од зачетника новчаних трансакција преко интернета, али и незаобилазни платни систем до данас. Уколико нека продавница жели да има велик домаћај (енг. *Reach*) ка корисницима, *Pay Pal* је у том случају *must-have*. Њена интеграција се врши уз помоћ скрипте која се увлачи приликом учитавања странице, где је потребно проследити претходно креирани *developer*-ски *API* кључ. Увлачењем скрипте омогућено је убацивање неког од њених популарних дугмића који прослеђују корисника на њен прозор. Овим је сва одговорност у вези са сигурношћу трансакција пребачена на страну разрађеног платног система, те је ризик од малверзација сведен на статистичку грешку.

3. ФУНКЦИОНАЛНОСТИ

У овом поглављу, урађен је осврт на неке од кључних функционалности *web* апликације.

Почетна страна и странице производа

На почетној страници *web* апликације која је видљива свима, приказан је преглед свих производа. Наиме, приказани су основни подаци као што су цена производа, слика, назив и просечна оцена потрошача. Кликком на слику или назив продукта помоћу *React Link*-а отвара се нова страница са његовим детаљнијим описом и листом рецензија потрошача. Такође, на овој страници обезбеђено је и убацивање жељених производа у корпу, где су даље могућности блокиране све док се корисник не пријави. Сваки производ саржи поље количина (енг. *Quantity*), које купцу говори да ли га има на стању. Ажурирање стања омогућено је администратору о чему ће више речи бити у наставку.

Претрага производа

Како би кориснику било омогућено што лакше коришћење апликације, креирана је компонента *SearchBox*, која обезбеђује једноставну претрагу производа. Помоћу *Regex*-а обезбеђена је претрага где није неопходан унос идентичног назива већ сви производи који садрже тражене речи ће бити исфилтрирани и приказани на новој страници. Такође, обезбеђено је и игнорисање малих и великих слова, чиме је олакшана претрага тражених производа.

Регистрација и активација налога

Уколико би корисник желео да креира поруџбину неопходно је да буде пријављен. Као нови корисник први корак би била регистрација где би се унели тражени подаци као што су име, *e-mail* адреса и шифра. Након тога, на *e-mail* адресу стиже порука за активацију корисничког налога преко *Sendgrid*-а чијом потврдом, уласком на линк, се завршава процес регистрације.

Аутентикација

Како би се заштитили подаци, као што је, на пример, профил корисника креиран је *authMiddleware* за аутентикацију. Заштита је обезбеђена помоћу *JWT* токена, чије се креирање врши уз помоћ *npm* пакета који се назива *JSONWebToken*.

Пријава и одјава корисника

Након што је корисник креирао налог омогућена је пријава. Приликом пријаве долази до провере унетих података са подацима из базе. Уколико се подаци не поклапају долази до приказа грешке која је добијена од сервера. У супротном корисник се прослеђује на почетну страну апликације, те се дешава ажурирање навигационог менија које сада садржи његов профил. Као што је горе већ речено, профил је заштићен помоћу токена. Након пријаве, на *web* претраживачу се чувају подаци о кориснику у *Local Storage*-у докле год је он пријављен. По одјави, подаци о кориснику се бришу као и друга чувана поља.

Ажурирање профила

Докле год је корисник пријављен може манипулисати подацима које је приложио. Та манипулација представља ажурирање профила и може се извршити на посебној страници преко навигационог менија. Ажурирање профила подразумева измену података

као што су корисничко име, *e-mail* адреса и шифра. За промену корисничког имена или *e-mail* адресе проверава се да ли исти већ постоје у бази, а приликом ажурирања лозинке потребно је нову лозинку унети два пута.

Рецензије

Могућност остављања и прегледа рецензије пружа кориснику додатно искуство приликом одабира производа, где може непосредно увидети искуства претходних купаца и видети да ли је овај производ стварно за њега. Сваки корисник може једном оставити оцену од 1 до 5 и опционо додати свој коментар, што се потом додаје у базу података и постаје видљиво осталим корисницима. Овим се просечна оцена аутоматски прерачунава за нову оцену. Ова опција омогућена је искључиво за оне кориснике који су пријављени.

Корпа

Корпа представља складиште производа које корисник одабере за своју наредну поруџбину. Уласком у корпу корисник може прегледати све производе које је одабрао. Уколико је одустао од куповине неког од артикала, исти може једноставно избацити кликом на икону кантице.

Поруџбине

Када је корисник задовољан са одабиром производа које жели да купи прелази на следећи корак а то је креирање поруџбине. Докле год поруџбина није плаћена дозвољено је отказивање исте. У било ком тренутку корисник може да провири статус своје поруџбине одласком на посебну секцију свог профила.

Плаћање

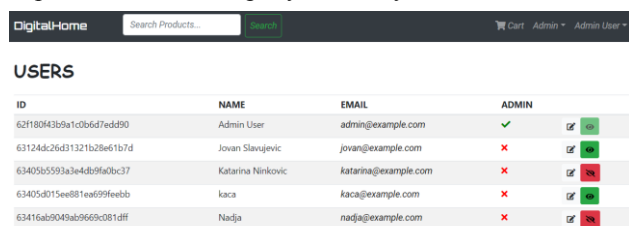
Уколико је креирао поруџбину, корисник може у било ком тренутку реализовати плаћање путем платног промета *Pay Pal* где се успешном трансакцијом стање по аутоматизму мења у плаћено.

Администраторски мени

Поред корисника креиран је и администратор који управља сајтом. Пријављивањем као администратор отвара се ново поље у навигационом менију са називом админ. Ово поље садржи мени са секцијама корисници, производи и поруџбине. Администратор има могућност прегледа и манипулације подацима из тих секција.

Преглед и (де)активација корисника

Уласком у секцију корисника, отвара се преглед свих корисника и њихов тренутни статус.



ID	NAME	EMAIL	ADMIN
62f180f43b9a1c0b6d7ed490	Admin User	admin@example.com	✓
63124dc26d31321b28e61b7d	Jovan Slavujevic	jovan@example.com	✗
6340565593a3e4db9fa0bc37	Katarina Ninkovic	katarina@example.com	✗
63405d015ee881ea699feebb	kaca	kaca@example.com	✗
63416ab9049ab9669c081df	Nadja	nadja@example.com	✗

Слика 3. Приказ листе корисника из админ менија

Администратор има могућности ажурирања профила корисника са могућношћу постављања неког од

корисника за админа. Такође, администратор може деактивирати налог уколико то сматра потребним или га вратити у активно стање. Деактивираним кориснику је онемогућена пријава, а тиме и све могућности везане за пријаву.

Преглед и манипулација производима

Све креиране производе је могуће ажурирати, где су сва поља отворена за измену, али се могу и креирати потпуно нови производи. Још једна од опција манипулацијом производа јесте њихово брисање.

Преглед, достављање и отказивање поруџбина

Администратор има на увид све поруџбине које су креиране и њихове статусе. Уколико неку од поруџбина није могуће извршити постоји могућност отказивања, али само у случају да је корисник није платио. Када се провери да ли је уплата прошла администратор може покренути процес доставе, чиме се ажурира стање поруџбине.

4. ЗАКЉУЧАК

Може се закључити да је систем чији је циљ био примена клијент-сервер комуникације са практичном применом успешно реализован. Пројекат је укључио развијање модерног *web* сајта са последњим технологијама за развијање истог.

Предлози за побољшање:

- Категоризација и филтрирање производа
- Подршка за *Android* и *iOS* кроз мобилне апликације
- *Chat bot* за добављање генеричких информација
- Додавање више слика на једном производу

5. ЛИТЕРАТУРА

- [1] SHAN, DHRUTI NA. *Node .JS*. BPB PUBLICATIONS, 2018.
- [2] Chodorow, Kristina. *MongoDB*. O'Reilly, 2013.
- [3] Dunk, Jim. *React*. Van Duuren Media, 2019.
- [4] Lee, Stan, and Jack Kirby. *Redux*. Tunbridge Wells, Kent: Panini Publishing, 2018.

Кратка биографија:



Катарина Нинковић рођена је у Руми 1996. год. Дипломски рад на Факултету техничких наука из области Електротехника и рачунарство – Примењена електроника одбранила је 2020.год. Области интересовања су јој развој софтвера у програмским језицима *Java* и *Javascript*.

контакт: kaca.ninkovic.69@gmail.com

RAZVOJ WEB APLIKACIJE „Sistem za prihvatanje rezultata ispita i statistika predmeta“ DEVELOPMENT OF WEB APPLICATION „System for accepting exam results and course statistics“

Milena Vujičić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je prikazan razvoj web aplikacije – Sistem za prihvatanje rezultata ispita i statistiku predmeta Programski jezici i strukture podataka. Pri razvoju aplikacije korišćeni su Django i Angular framework, Google Drive API, kao i Docker alat za kontejnerizaciju. Aplikacija obavlja konverziju podataka iz najčešće korišćenih formata za beleženje rezultata ispita u format Google Sheet tabele i omogućava da svi korisnici imaju na raspolaganju sve podatke o rezultatima testova i ispita na jednom mestu, u istom formatu, podesnom za statističku obradu.

Ključne reči: web aplikacija, rezultati ispita, statistička obrada podataka

Abstract – This paper presents development of a web application - System for accepting exam results and statistics of the subject Programming Languages and Data Structures. Django and Angular framework, Google Drive API, and Docker tool for containerization were used in the development of the application. The application converts data from the most commonly used formats for recording results into a Google Sheet table format and allows all users to have available all data on test and exam results in one place, in the same format, suitable for statistical processing.

Keywords: web application, exam results, statistical data processing

1. UVOD

Zadatak rada je izrada web aplikacije koja olakšava unos i obradu rezultata ispita iz predmeta Programski jezici i strukture podataka, na Fakultetu tehničkih nauka u Novom Sadu. Rezultati polaganja preispitnih obaveza i ispita unose se u različitim formatim i ručno se prenose u tabelu dostupnu svim učesnicima u postupku. Teorijski deo ispita se automatski pregleda i rezultati se dostavljaju dežurnom nastavniku u podrazumevanom formatu korišćenog programa.

Aplikacija koja je predmet rada omogućava konverziju iz najčešće korišćenih formata za beleženje rezultata u format Google Sheet tabele. Time će biti omogućeno da svi korisnici imaju sve podatke raspoložive na jednom mestu, u istom formatu, podesnom za statističku obradu.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Srđan Popov, red. prof.

Praćenje promena, implementirano u aplikaciji, jeste poboljšanje koje predmetnom profesoru daje mogućnost da proveri da li su svi uključeni u izvođenje nastave, pregled radova i evidentiranje rezultata ispita uneli potrebne informacije, koje se prosleđuju Studentskoj službi. Takođe, jedinstven format i postojanje centralne baze podataka omogućava primenu statističkih metoda obrade podataka.

Aplikacija je razvijena korišćenjem Django [1] i Angular [7] framework-a, Google Drive API-ja, kao i Docker [4] alata za kontejnerizaciju rešenja.

2. OPIS Aplikacije

Serverska aplikacija kreirana je korišćenjem Django framework-a, biblioteke django-rest-framework i Google Drive API. Kontejnerizacija aplikacije je obavljena u Docker-u.

Klijentska aplikacija implementirana je upotrebom Angular framework-a. Ona služi za posrednu interakciju korisnika sa centralnim Google Sheet-om. U klijentskoj aplikaciji implementirane su funkcije za dodavanje korisnika, unos i upload podataka na server, kao i prikaz logova i statistike.

2.1. Serverska aplikacija

Model podataka

U programu postoje sledeći modeli:

AppUser - modeluje korisnika aplikacije. Nasleduje klasu `django.contrib.auth.models.AbstractBaseUser`. Ova klasa sadrži prototip za kreiranje korisnika aplikacije,

LogEntry - modeluje podatke unete ili izmenjene od strane korisnika

FileUpload - modeluje fajl koji je upload-ovan na server.

Klase `AppUser` i `LogEntry` čuvaju se trajno u bazi podataka, dok klasa `FileUpload` služi za mapiranje i privremeno čuvanje putanje fajla na serveru

Za korisnika aplikacije kreirana je i `AppUserManager` klasa. Klasa nasleduje ugrađenu klasu `django.contrib.auth.models.BaseUserManager`.

`AppUserManager` klasa definiše način na koji će se kreirati korisnik i superuser aplikacije. Funkcijom `django.contrib.hashers.make_password()` omogućeno je čuvanje heširane lozinke.

Serijalizatori podataka

S obzirom da se podaci potrebni za kreiranje instance klase FileUpload ne prosleđuju putem REST zahteva, za ovu klasu nije bilo potrebno kreirati serijalizatore.

Klasa AppUser ima implementirane tri vrste serijalizatora: UserSerializer, EmailSerializer, SuperUserSerializer. UserSerializer se koristi za serijalizaciju podataka vezanih za običnog korisnika. EmailSerializer serializuje samo email i id korisnika. SuperUserSerializer ima mogućnost da serializuje sva polja vezana za superuser-a aplikacije.

Klasa LogEntry ima implementirana dva serijalizatora: LogEntrySerializer i TableSerializer. Razlika između ova dva serijalizatora je samo u poljima user i date. LogEntrySerializer serializuje podatke vezane za podatke o izmenama. TableSerializer se koristi za statistiku izmena koju vidi korisnik, pa podaci o datumu izmene i korisniku nisu potrebni.

Rad sa korisnicima programa

U aplikaciji postoje dve vrste korisnika: običan korisnik i superuser. Superuser ima mogućnost da vidi podatke koji se čuvaju u log-u, kao i da dodaje nove korisnike aplikacije. Prilikom pokretanja aplikacije prvi put, potrebno je ručno dodati superuser-a aplikacije. Funkcija user_count() proverava broj korisnika aplikacije. Operacije nad korisnicima, koje nisu dodavanje prvog superusera, nisu moguće, ako u aplikaciji ne postoji ni jedan korisnik. Broj korisnika aplikacije se pamti u request sesiji.

Sa serverske strane prijavljivanje na aplikaciju se vrši preko funkcije user_login(). Nakon slanja zahteva proveravaju se email i password korisnika. Funkcija django.contrib.auth.hashers.check_password() proverava ispravnost heša unete lozinke.

Rad sa unetim podacima

Rad sa unetim podacima se sastoji iz tri dela: preuzimanje i parsiranje podataka poslatih u zahtevu, obrada unetih podataka i postavljanje podataka na Google Sheet putem Google Drive API-ja.

Rad sa csv i zip fajlovima

Korisnik, u zavisnosti od vrste testa čije rezultate unosi, na server šalje ili .csv ili .zip datoteku. Datoteke se šalju u string base64 formatu. Nakon što se datoteka prevede iz string u binarni format, privremeno se čuva kao statički fajl, a njena putanja u bazi podataka. Nakon što je završena obrada datoteke, ona se briše iz baze podataka i sa servera. Pandas dataframe podržava direktno učitavanje .csv datoteke funkcijom pandas.read_csv(). Struktura svakog .zip fajla je takva da u njoj postoji jedna .csv datoteka. Pošto pandas ne podržava direktno čitanje .zip datoteke, potrebno je prvo ekstrahovati .csv datoteku. Za ekstrakciju .csv datoteke korišćena je biblioteka ZipFile. Ona poseduje funkcije za čitanje, pregled strukture i ekstrahovanje .zip fajlova.

Funkcija parse_file_data, koja pravi fajl i stavlja ga na server.

Obrada podataka

Obrada unetih podataka obavlja se putem pandas biblioteke. Nakon što se od unete datoteke napravi DataFrame nad njim se vrši niz neophodnih operacija. Tabela koja se čuva na centralnom Google Sheet dokumentu ima različit način imenovanja kolona. Zbog toga je potrebno prvo formatirati imena kolona i ukloniti nepotrebne kolone iz DataFrame-a. Format broja indeksa, koji se koristi kao ključ po kome se podaci upisuju u tabelu, je drugačiji na centralnom Google Sheet-u i u .csv fajlu. Parsiranje i reformatiranje broja indeksa izvršeno je putem regex izraza. Regex izraz očitava smer, broj upisa i godinu iz indeksa i čuva te podatke za dalju upotrebu. Na osnovu godine upisa određuje se na koji list u centralnom Google Sheet-u je potrebno upisati podatke.

Program u svom settings.py fajlu sadrži promenljivu CURRENT_YEAR koja čuva podatak o tekućoj školskoj godini. Studenti upisani pre ove školske godine se razrstavaju na poseban list. Ostali studenti se razrstavaju na osnovu smeru na koji su upisani. Nakon očitavanja podataka iz centralnog Google Sheet-a i njihovog prevođenja u DataFrame, uneti podaci se upoređuju sa njima. Funkcija pandas.merge() i pozivanje agregatora .groupby() vrši spajanje podataka dva DataFrame-a na osnovu broja indeksa. Ovakvim spajanjem se ažuriraju podaci u odgovarajućem redu tabele.

Rad sa Google Sheet tabelama

Za rad sa Google Sheet [2] tabelama potrebno je koristiti Google Drive API. google_auth_oauthlib.flow biblioteka sadrži funkcije i klase za autorizaciju korisnika. Klasa InstalledAppFlow iz ove biblioteke implementira funkciju from_client_secrets_file uzima putanju do ključa kojim se vrši autentifikacija korisnika. Putanja do ovog fajla čita se iz environment variable. Nakon što je autentifikacija uspešno izvršena moguće je vršiti operacije nad Google Sheet tabelom. Funkcije open_sheet_for_reading i open_sheet_for_writing implementirane su tako da omogućuje čitanje tabele prema unetom id-ju. Upis i čitanje tabele se vrši na osnovu imena listova u tabelama.

Log i statistika

Log korišćenja aplikacije se čuva nakon upisa podataka na centralni Google Sheet. Log sadrži sve unete izmene. Pošto je potrebno kreirati više od jedne instance klase LogEntry, koristi se funkcija bulk_create koju implementiraju objekti modela.

Statistika koju korisnik vidi su svi podaci koji postoje u centralnom Google Sheet-u. Na klijentski deo aplikacije potrebno je poslati sve podatke iz centralnog Google Sheet-a. Funkcija get_statistics se svodi na čitanje podataka iz Google Sheet tabele.

2.2. Klijentska aplikacija

Model podataka

Model podataka implementiran je putem interfejsa AppUser, Log, Count i ResultsFormat. Osim interfejsa

Count, model se poklapa sa modelom na serverskoj strani. Interfejs Count služi za proveru trenutnog broja korisnika u aplikaciji.

Rad sa korisnicima

Da bi bio omogućen rad sa ostalim funkcionalnostima aplikacije, tokom prvog pokretanja, potrebno je dodati superuser-a.

Unos podataka za kreiranje superuser-a vrši se putem forme. Forma je mapirana na model korisnika. U TypeScript [6] fajlu komponente definisan je način slanja REST zahteva na serverski deo aplikacije. REST zahtev se šalje putem klase HttpClient. Ova klasa stvara publish subscribe relaciju za REST zahtev.

Superuser aplikacije ima mogućnost da dodaje druge korisnike aplikacije. Forma slična onoj za dodavanje superuser-a postoji i za dodavanje drugih korisnika aplikacije. Tokom dodavanja korisnika traži se autentifikacija superuser-a koji dodaje novog korisnika.

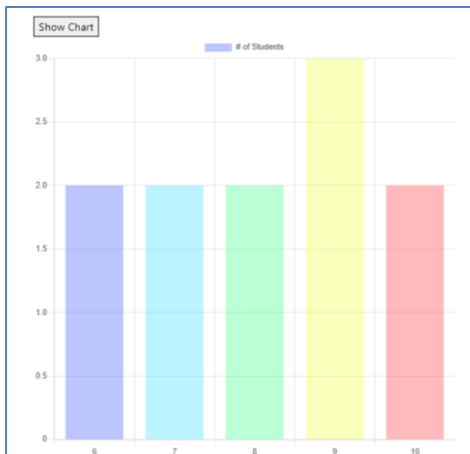
Unos podataka

Forma za unos podataka sadrži polje i radio button-e za izbor tipa testa za koji se unose rezultati i polje za upload fajla. Polje za upload fajla učitava fajl sa uređaja korisnika i prosleđuje ga u tekstualnom formatu na server, sl. 1.

Slika 1. Forma za unos podataka

Prikaz statistike i log-a

Statistički proračuni obavljaju se na osnovu podataka iz centralnog Google Sheet-a. Korisnik ima mogućnost da vidi dve vrste podataka, sve podatke o studentima, zajedno sa ocenama ili statistiku na nivou predmeta (prolaznost, prosečna ocena, histogram raspodele ocena), slika 2.



Slika 2 – Histogram

Na klijentskoj strani aplikacije implementirane su funkcije za računanje prolaznosti i prosečne ocene.

Tabela koja prikazuje podatke o studentima ima mogućnost filtriranja podataka po broju indeksa, imenu ili prezimenu.

Crtanje histograma raspodele ocena urađeno je sa Chart.js bibliotekom. Ova biblioteka nudi različite funkcije za grafičko predstavljanje podataka korišćenjem canvas-a.

Log podaci

Log podatke može da vidi samo superuser aplikacije. Podaci koji se dobiju sa serverske strane aplikacije se smeštaju u tabelu Change log, slika 3.

Index number	Last name	First name	T1234	SDV	P1	P2	Exam	User	Date
RA 1/2023	Markovic	Marko	-1	-1	13	-1	-1	admin@example.com	2022-10-17T18:45:34.823351Z
RA 2/2023	Ivic	Ana	-1	-1	12	-1	-1	admin@example.com	2022-10-17T18:45:34.823351Z
RA 123/2023	Peric	Darko	-1	-1	10	-1	-1	admin@example.com	2022-10-17T18:45:34.823351Z
RA 5/2023	Nedic	Nikola	-1	-1	10	-1	-1	admin@example.com	2022-10-17T18:45:34.823351Z
RA 15/2023	Radic	Nemanja	-1	-1	1	-1	-1	admin@example.com	2022-10-17T18:45:34.823351Z
RA 18/2023	Radic	Nemanja	-1	-1	14	-1	-1	admin@example.com	2022-10-17T18:45:34.823351Z
RA 6/2023	Filipovic	Svetislav	-1	-1	13	-1	-1	admin@example.com	2022-10-17T18:45:34.823351Z
PR 14/2023	Popovic	Nenad	-1	-1	13	-1	-1	admin@example.com	2022-10-17T18:45:34.823351Z
PR 120/2023	Ubic	Dunja	-1	-1	7	-1	-1	admin@example.com	2022-10-17T18:45:34.823351Z
PR 5/2023	Arsic	Anja	-1	-1	9	-1	-1	admin@example.com	2022-10-17T18:45:34.823351Z
PR 15/2023	Sretic	Ana	-1	-1	5	-1	-1	admin@example.com	2022-10-17T18:45:34.823351Z
PR 6/2023	Andjelic	Zoran	-1	-1	12	-1	-1	admin@example.com	2022-10-17T18:45:34.823351Z
RA 260/2018	Radic	Nemanja	-1	-1	5	-1	-1	admin@example.com	2022-10-17T18:45:34.823351Z
PR 14/2018	Kovacic	Djordje	-1	-1	8	-1	-1	admin@example.com	2022-10-17T18:45:34.823351Z
RA 1/2023	Markovic	Marko	-1	-1	-1	10	-1	admin@example.com	2022-10-17T18:45:34.823351Z
RA 123/2023	Peric	Darko	-1	-1	-1	16	-1	admin@example.com	2022-10-17T18:45:34.823351Z
RA 18/2023	Radic	Nemanja	-1	-1	-1	15	-1	admin@example.com	2022-10-17T18:45:34.823351Z
RA 5/2023	Nedic	Nikola	-1	-1	-1	12	-1	admin@example.com	2022-10-17T18:45:34.823351Z
RA 15/2023	Radic	Nemanja	-1	-1	-1	5	-1	admin@example.com	2022-10-17T18:45:34.823351Z
PR 6/2023	Andjelic	Zoran	-1	-1	-1	10	-1	admin@example.com	2022-10-17T18:45:34.823351Z
PR 120/2023	Ubic	Dunja	-1	-1	-1	4	-1	admin@example.com	2022-10-17T18:45:34.823351Z
RA 260/2018	Radic	Nemanja	-1	-1	-1	8	-1	admin@example.com	2022-10-17T18:45:34.823351Z
PR 14/2018	Kovacic	Djordje	-1	-1	-1	5	-1	admin@example.com	2022-10-17T18:45:34.823351Z

Slika 3 – Change log

2.3 Kontejnerizacija rešenja

Serverska i klijentska aplikacija su kontejnerizovane uz pomoć Docker i Docker compose alata. U CMD direktivi Dockerfile-a serverske aplikacije poziva se skript fajl koji vrši pokretanje produkcionog servera (Gunicorn [8]) i migraciju modela u bazu podataka.

Pre nego što je kontejnerizovana klijentska aplikacija, pozvana je komanda ng build, Angular framework-a. Ova komanda vrši prevođenje svih fajlova Angular aplikacije u HTML i JavaScript format. Produkcioni server koji se koristi za serviranje statičkih fajlova je NGINX [5]. U konfiguracionom fajlu navedena je putanja do početne stranice. Takođe, navedena je i direktiva za rutiranje adresa.

Docker compose fajl objedinjuje serversku i klijentsku aplikaciju, kao i bazu podataka i interfejs za upit podataka. Svi osetljivi podaci iz aplikacija stavljeni su u docker_vars.env fajl. Nalaze se u formatu ključ=vrednost.

3. ZAKLJUČAK

Korišćenjem Django i Angular framework-a, Google Drive API-ja, kao i Docker alata za kontejnerizaciju rešenja ostvaren je cilj rada: konverzija podataka iz najčešće korišćenih formata za beleženje rezultata u format Google Sheet tabele i omogućeno je da svi korisnici imaju na raspolaganju sve podatke o ispitima i testovima na jednom mestu, u istom formatu, podesnom

za statističku obradu. Kreiranjem Web aplikacijom poboljšan je postupak unosa i obrade rezultata ispita iz predmeta Programski jezici i strukture podataka, čime se ostvaruje ušteda vremena utrošenog na evidentiranje podataka, smanjuje mogućnost pogrešnog unosa i omogućava analiza ostvarenih rezultata.

4. LITERATURA

- [1] Django - <https://docs.djangoproject.com>
- [2] Google Sheets - <https://developers.google.com/sheets/api/quickstart/python>
- [3] Django REST biblioteka - <https://www.django-rest-framework.org/api-guide/filtering/>
- [4] Docker - <https://docs.docker.com/>
- [5] NGINX - <https://nginx.org/en/docs/>
- [6] Typescript <https://www.typescriptlang.org/docs/>
- [7] Angular <https://angular.io/docs>
- [8] WSGI <https://peps.python.org/pep-0333/>

Kratka biografija:



Milena Vujičić rođena je u Vršcu 1997. godine. Gimnaziju „Jovan Jovanović-Zmaj“ u Novom Sadu završila je 2016. godine i upisala je Fakultet tehničkih nauka, odsek Računarstvo i automatika. Osnovne akademske studije završila je 2020. godine. Iste godine upisala je master studije na istom fakultetu, odsek Primenjene računarske nauke i informatika-Računarstvo visokih performansi.
kontakt:
vujicic.milena.97@gmail.com

PROJEKTOVANJE I RAZVOJ PULSNOG OKSIMETRA**DESIGN AND DEVELOPMENT OF A PULSE OXIMETER**Igor Đeviki, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu je opisan sistem za merenje oksigenisanosti krvi i pulsa. U radu su predstavljeni proces i sistem razmene kiseonika u ljudskom organizmu i princip rada pulsno oksimetra. Takođe je predstavljeno projektovanje i realizacija sistema, odnosno izgradnja svih hardverskih komponenti kao i programiranje mikrokontrolera za merenje oksigenisanosti i pulsa.

Ključne reči: Merenje oksigenisanosti, Merenje pulsa, Merna sonda, Easy PIC Pro v7

Abstract – This paper describes a system for measuring blood oxygenation and pulse. The paper presents the process and system of oxygen exchange in the human body and the principle of operation of the pulse oximeter. The design and realization of the system was also presented, i.e. the construction of all hardware components as well as the programming of the microcontroller for measuring oxygenation and heart rate.

Keywords: Measuring oxygenation, Measuring pulse, Measuring probe, Easy PIC Pro v7

1. UVOD

Kiseonik je neophodan za ispravno funkcionisanje svake ćelije u ljudskom organizmu, tj. neophodan je za proces ćelijskog disanja i metabolizma. Ćelije stvaraju energiju potrebnu za sve životne procese i uz pomoć kiseonika hemijski razgrađuju proteine, ugljene hidrate i masti. Ćelije mogu da izdrže neko vreme bez kiseonika ali pri njegovom odsustvu tokom nekog dužeg perioda počinju polako da odumiru, npr. skeletna mišićna masa može da preživi 60 min do 90 min bez kiseonika, bubrezi i jetra 15 min do 20 min, a mozak manje od 3 min. Dakle, isporuka kiseonika ćelijama je važan pokazatelj zdravlja ljudskog organizma zbog čega je razvijeno nekoliko dijagnostičkih metoda za analizu isporuke kiseonika, jedna od kojih je pulsna oksimetrija.

Pulsna oksimetrija je najrasprostranjenija neinvazivna metoda za merenje zasićenosti arterijske krvi kiseonikom i merenje pulsa. Uvedena 1983. godine i od tad priznata širom sveta kao standard u svim kliničkim oblastima kao što su anesteziologija, intenzivna nega, operacione sale, hitna pomoć, opšta medicina, porodilišta, kućna nega i veterina.

Pulsna oksimetrija se zasniva na Beer-Lambertovom zakonu, apsorpcionih osobina krvi, odnosno hemoglobina, i

pulsacijom arterijske krvi pomoću kojih se može odrediti nivo oksigenisanosti i puls pacijenta.

2. BEER-LAMBERTOV ZAKON

Pulsna oksimetrija se u osnovi zasniva na Beer-Lambertovom zakonu koji opisuje slabljenje svetlosti koja putuje kroz homogenu sredinu koja sadrži apsorbujuću supstancu. Ako monohromatska upadna svetlost uđe u sredinu, jedan deo te svetlosti se prenosi kroz sredinu dok će se drugi deo apsorbovati. U ljudskoj krvi hemoglobin igra ulogu glavne apsorbujuće supstance svetlosti.

Apsorbujuće karakteristike hemoglobina se menjaju sa promenom njegovog hemijskog sastava i od ovih promena zavisi koja će se upadna svetlost više apsorbovati. Većina hemoglobina kod zdrave osobe čine funkcionalni hemoglobini i njegova zasićenost, tj. funkcionalna zasićenost arterijske krvi kiseonikom (funkcionalna SaO₂) se meri u procentima i određuje količinom oksigenisanog hemoglobina (HbO₂) u poređenju sa zbirom oksigenisanog i deoksisogenisanog hemoglobina (Hb).

$$S_aO_2 = \frac{HbO_2}{Hb + HbO_2} \times 100\%$$

Međutim upadna svetlost koja prolazi kroz sredinu kao što je ljudsko tkivo se ne deli samo na apsorbovanu i propuštenu svetlost, kao što je prethodno opisano Beerovim zakonom, nego se deli i na reflektovanu i na rasejanu svetlost. Površina kože, tkivo, mišići, kost, a posebno krv izazivaju rasejanje svetlosti što povećava apsorpciju svetlosti. Krv je nehomogena tečnost koja je sposobna za nelinearnu apsorpciju svetlosti, razlog u varijaciji apsorpcije je promena ose crvenih krvnih zrnaca koja imaju oblik bikonkavnog diska, što utiče na njihovu apsorpciju. Pored efekta rasejanja svetlosti i nelinearne apsorpcije krvi pulsni oksimetri ipak dovoljno tačno očitavaju zasićenost arterijske krvi kiseonikom za kliničku upotrebu u normalnim okolnostima.

Ovo je zbog činjenice da većina komercijalnih pulsni oksimetara koriste kalibracionu krivu zasnovanu na empirijskim podacima, zato što je matematičko modeliranje problema rasejanja svetlosti za različite uslove veoma složeno. Za dobijanje empirijske kalibracione krive prikuplja se veliki skup podataka dobijenih u kliničkim studijama koji sadrže informacije o odnosu (R) apsorpcija, izračunatih pomoću pulsno oksimetra, i stvarne arterijske zasićenosti kiseonikom (SaO₂), izmerene veoma tačnom metodom kao što je CO-oksimehtar. Tabele ili jednačine se koriste za pronalaženje odnosa ove dve promenljive za očitavanje pulsno

NAPOMENA:

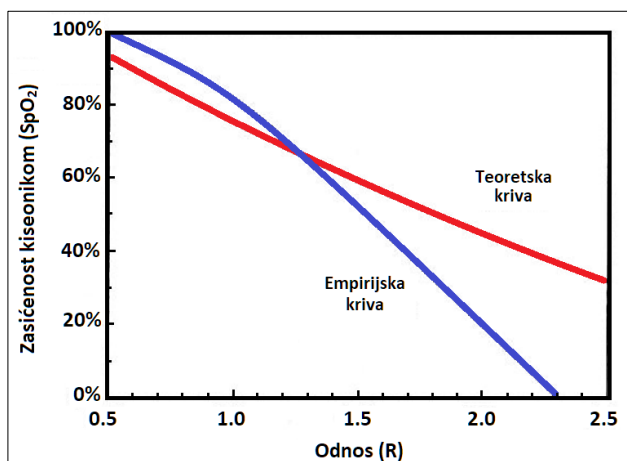
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Platon Sovilj, red. prof.

oksimetra. Da bi se izmerene vrednosti odnosa R povezale sa očitavanjem pulsog oksimetra, jednačina zasnovana na Beerovom zakonu se može modifikovati:

$$SpO_2 = \frac{k_1 - k_2 R}{k_3 - k_4 R}$$

Ove konstante k_i su određene na osnovu kliničkih studija kako bi se kriva najbolje uklapala u eksperimentalno izmerene podatke. Drugi pristup za matematičku reprezentaciju je upotreba polinoma:

$$SpO_2 = k_1 + k_2 R + k_3 R^2$$



Slika 1. Poređenje teoretske kalibracione krive sa empirijskom kalibracionom krivom

Poređenje teoretske kalibracione krive sa empirijskom kalibracionom krivom (Slika 1.) daje primer poređenja empirijske kalibracione krive sa teoretskom kalibracionom krivom. Treći oblik jednačine koji mnogi komercijalni pulsni oksimetri danas koriste, ovaj oblik je korišćen u ovom radu, je:

$$SpO_2 = A - B * R$$

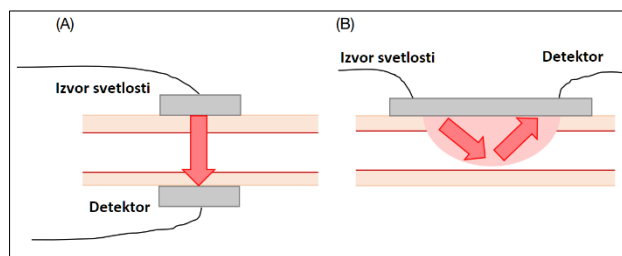
gde se uglavnom za konstantu A koristi vrednost 110, dok se za konstantu B koristi 25. Generalno komercijalni pulsni oksimetri su empirijski kalibrisani sa opsegom saturacije kiseonika od 70% do 100%. Zbog ovakvog načina kalibracije moguće je da pulsni oksimetri različitih brendova pokazuju različite vrednosti oksigenisanosti.

3. PULSNI OKSIMETAR

Zasićenost arterijske krvi kiseonikom se određuje merenjem apsorpcije svetlosti za dve različite talasne dužine i korišćenjem arterijske pulsacije. Arterijska pulsacija nam omogućava da razlikujemo apsorpciju arterijske krvi od drugih apsorbera kao što su koža, meko tkivo i kost. Svetlost koja pada na ljudsko tkivo se delimično prenosi, reflektuje i apsorbuje od strane kože, tkiva i krvi pre nego što stigne do detektora. Ovo rasipanje svetlosti u tkivu omogućava merenje u režimu transmisije i u režimu refleksije (Slika 2.). Kao što sam naziv kaže transmisioni pulsni oksimetar meri količinu svetlosti koja prolazi kroz tkivo, dok kod refleksionog pulsog oksimetra se meri količina svetlosti koja se reflektuje nazad do sonde.

Osnovna prednost transmisivnih sonda u odnosu na refleksivne jeste u intenzitetu svetlosti koju detektuje fotodetektor, zato što je količina svetlosti koja prolazi

kroz tkivo veća od količine reflektovane svetlosti i svetlost koja prolazi kroz tkivo je koncentrisana u određenom području. Glavni nedostatak transmisivnih sonda je taj što je primena senzora ograničena na periferne delove tela (prsti, uho i nos), dok se refleksivne sonde mogu postaviti bilo gde na telu.



Slika 2. A) merenje u režimu transmisije, B) merenje u režimu refleksije

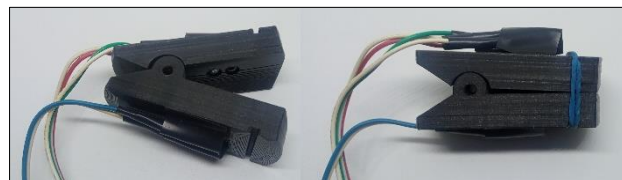
4. HARDVER

Pulsni oksimetar u ovom radu se može podeliti na analogni i digitalni deo, gde se analogni deo sastoji od merne sonde, filterskog bloka i kontrolnog bloka, dok se digitalni deo sastoji od mikroprocesorskog sistema koji obrađuje i prikazuje izmerene podatke

1) Analogni deo

- Merna Sonda

Za realizaciju pulsog oksimetra odabran je princip rada zasnovan na transmisiji zbog prethodno navedenih prednosti. Sonda koja se koristi kod ove metode je u osnovi štipaljka koja je napravljena tako da se u nju mogu postaviti senzorski elementi i da se može sa lakoćom postaviti na vrh kažiprsta, vršeći dovoljan pritisak da ne spadne i da ne bude neprijatno.



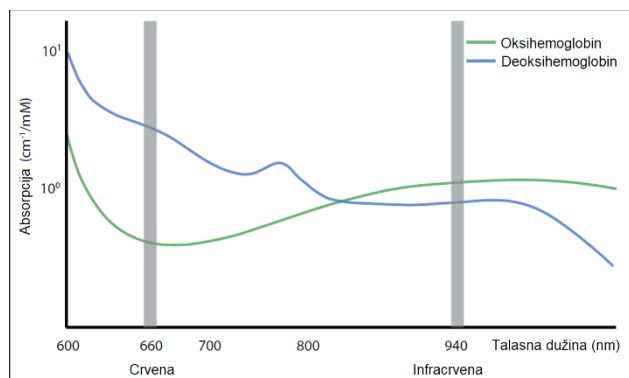
Slika 3. Prikaz merne sonde

Napravljena je tako da što više obuhvati vrh prsta, sa ciljem da se senzorski elementi fiksiraju za prst i na taj način smanje smetnje izazvane pomerajima, smetnje spoljašnjeg svetla, kao i da se smanji refleksija emitovane svetlosti. U štipaljku se mogu postaviti LED diode i fototranzistor, gde jedna od LED dioda emituje crvenu svetlost (približno 660 nm), dok druga emituje infracrvenu svetlost (približno 940 nm), a izabrani fototranzistor je SFH300, pomoću kojeg će se izmeriti intenzitet propuštene svetlosti na osnovu čega se može odrediti koliko je svetlosti apsorbovano. Oksigenisani hemoglobin (HbO2) će apsorbovati više infracrvene svetlosti, dok će dezoksigenisani hemoglobin (Hb) apsorbovati više crvene svetlosti.

- LED Kontrolni Blok

Kako bi fototranzistor mogao da meri intenzitet svake svetlosti pojedinačno potrebno je rešiti problem smenjivanja rada crvene i infracrvene LED diode. Primenom mikrokontrolera uvodi se sinhronizacija

pulsiranja LED dioda sa odabiranjem ADC-a i na taj način se odgovarajući uzorci sa fotodetektora mogu pripisati odgovarajućoj talasnoj dužini.



Slika 4. Grafik odnosa absorpcije i talasne dužine

Kako bi precizno upravljali radom LED dioda potrebno ih je povezati na mikrokontroler, međutim napajati LED diode direktno sa mikrokontrolera može negativno da utiče na njegov rad zbog čega je potrebno uvesti LED drajver. LED drajver je električno kolo koje nam omogućuje da se LED diode povežu direktno na napajanje, dok će se pomoću tranzistora upravljati njihovim radom.

- Filter

Pošto je signal sa fototranzistora slab neophodno ga je prvo pojačati, a potom izbaciti neželjene smetnje, stoga se signal prvo dovodi na inverujući pojačavač koji pojačava ukupan signal 470 puta. Nakon čega se signal propušta kroz RC visokopropusni filter, granične frekvencije 0,7 Hz, kako bi se uklonila jednosmerna (DC) komponenta. Sledeća faza je aktivni niskopropusni filter koji je napravljen primenom operacionog pojačavača MCP602. Pojačanje i granična frekvencija niskopropusnog filtera su podešeni na 101 i 2,34 Hz, respektivno. Ovakva kombinacija visokopropusnog i niskopropusnog filtera pomaže pri uklanjanju neželjene jednosmerne komponente i šuma visoke frekvencije, uključujući mrežne smetnje od 50 Hz, dok se naizmenična komponenta male amplitude dodatno pojačava 101 put.

2) Digitalni deo

- Mikrokontroler

Mikrokontroler predstavlja integrisano kolo koje se primenjuje u programabilnim elektronskim sklopovima predviđenim za ugradnju, tj. u embedded sistemima. Pomoću njega možemo očitati signal sa filterskog kola da obradimo i prikažemo na računaru i da istovremeno precizno upravljamo LED kontrolnim blokom. Kako bi sve ovo postigli, a da previše ne zakomplicujemo naše

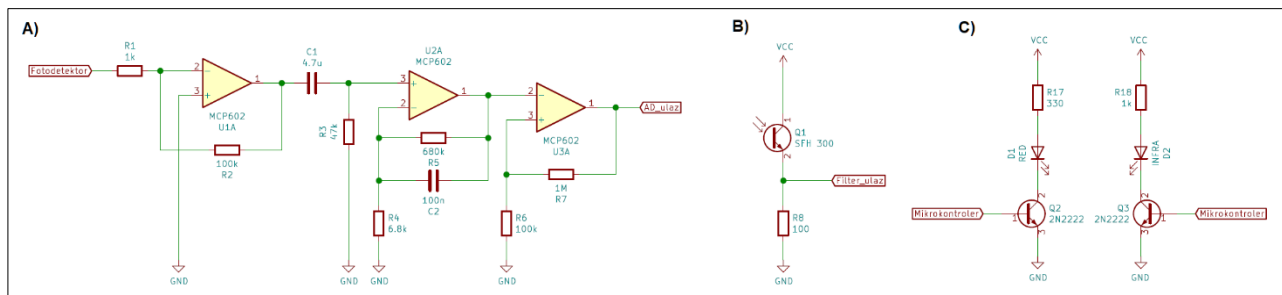
električno kolo, opredelili smo se za primenu Mikro-e-ove razvojne ploče "Easy Pic Pro V7". Ova razvojna ploča sadrži mnogo ugrađenih modula neophodnih za razvoj različitih aplikacija, kao što su u našem slučaju ADC, UART i napajanje, što nam omogućuje da se više fokusiramo na pisanje samog firmvera. Dolazi sa ugrađenim mikroProg programatorom i debagerom koji podržava preko sto pedeset različitih PIC mikrokontrolera, a sama ploča fabrički dolazi sa PIC18F87K22 mikrokontrolerom koji se nalazi na MCU pločici.

5. FIRMWARE

Na MCU pločici je pored samog mikrokontrolera fabrički ugrađen kristalni oscilator koji generiše takt frekvencije 16 MHz. Kako naš mikrokontroler poseduje PLL (Phase Lock Loop) kolo pomoću kojeg povećavamo frekvenciju sistemskog takta (FOSC) na 64 MHz. Zatim se prelazi na podešavanje parametara odabranih modula i određuju se GPIO pinovi preko kojih će se povezati analogni deo pulsog oksimetra, slika 5. Izlaz sa filterskog bloka se dovodi na ADC mikrokontrolera preko analognog ulaza u našem slučaju preko GPIO pina RA0 (analogni ulaz AN0). Za korektnu AD konverziju potrebno je da vreme konverzije jednog bita (TAD) bude što je moguće kraće ali da bude veće od mogućeg minimuma za TAD (0,8 μs). Kako je frekvencija sistemskog takta 64 MHz možemo odabrati da takt konverzije bude FOSC/64 i na ovaj način se dobija TAD koji iznosi 1 μs. Pošto A/D konverzija zahteva 14 TAD za svakih 12-bit (2 TAD pauze) naša konverzija će onda trajati 14 μs dok će vreme akvizicije (TACQ) iznositi 20 TAD, odnosno 20 μs. GPIO pinovi RB0 i RB1 su podešeni kao digitalni izlazi preko kojih mikrokontroler šalje signale kojima se upravljaju LED kontrolni blokovi. Za upravljanje state machine-a koristi se ECCP (Enhanced Capture/Compare/Pwm) modul, koji proširuje osnovne CCP funkcije koje pomažu pri merenju ili kontroli vremenskih pulsnih signala. Jedna od ovih proširenih funkcija se nalazi u Compare modu pod nazivom Special Event Trigger.

Compare mod konstantno poredi vrednost u svom registru sa vrednošću tajmerovog registra i kad se vrednosti poklope aktivira se Special Event Trigger. Special Event Trigger je generisani signal koji služi za aktiviranje radnji u drugim modulima, u ovom slučaju će se pokrenuti interrupt rutina kojom se upravlja promena stanja u state machine-u.

Samim odabirom ECCP1 modula će se automatski rezervisati pin RC2 kao digitalni izlaz. Pošto je ideja da se Special Event Trigger aktivira svakih 0,1 ms neophodno je uneti odgovarajuću vrednost u compare-ovom registru i podesiti odgovarajući korak tajmera.



Slika 5. Šema analognog dela pulsog oksimetra: A) Filterski blok, B) Detektorski blok, C) LED kontrolni blok crvene i infracrvene LED diode

Timer1 koji koristi ECCP1 je podešen tako da kao izvor takta koristi FOSC/4 sa uključenim preskalerom 1:2 kako bi se dobio korak tajmera od 125 ns. Zatim se može uneti vrednost u compare-ov registar koja se dobija deljenjem vremena za trigerovanje sa korakom tajmera, što će u našem slučaju iznositi:

$$\frac{0.1 \text{ ms}}{125 \text{ ns}} = \frac{100000 \text{ ns}}{125 \text{ ns}} = 800$$

Kako bi mikroprocesor slao podatke na računar pomoću serijske komunikacije potrebno je podesiti EUSART modul. U našem slučaju se koristi EUSART1, a samim tim su automatski rezervisani pinovi RC6 kao TX1 (Transmit) i RC7 kao RX1 (Receive). Zatim je podešen baud rate, odnosno brzina prenosa podataka, na 115200. Za računanje pulsa je korišćen Timer0 koji nakon 10 ms aktivira interrupt rutinu u kojoj se nalazi brojač, koji se inkrementira svaki put kada se aktivira interrupt. Ovo je postignuto podešavanjem tajmera da radi u 16 bitnom modu sa izvorom takta FOSC/4 i uključenim preskalerom 1:32. Nakon što su podešeni svi moduli mikrokontrolera može se preći na pisanje samog firmvera koji sadrži jednostavan "state machine" algoritam, tj. konačni automat, čija se stanja menjaju sa vremenom. Tokom jednog ciklusa koji traje 10 ms prolazi se redom kroz sva stanja automata tokom kojih se obavljaju sledeće operacije:

- 0 ms – pali se crvena LED dioda
- 1,5 ms – započinje A/D konverzija za crvenu LED diodu
- 2,5 ms – crvena LED dioda se gasi i traži se minimalna i maksimalna vrednost
- 5 ms – pali se infracrvena LED dioda
- 6,5 ms - započinje A/D konverzija za infracrvenu LED diodu nakon koje se vrednost prosleđuje funkciji za računanje pulsa
- 7,5 ms – infracrvena LED dioda se gasi i traži se minimalna i maksimalna vrednost nakon čega se računa odnos R i oksigenisanost
- 10 ms – kraj ciklusa, brojač se resetuje

Tokom jedne sekunde se prikupi sto smplova na osnovu kojih se nalaze minimalne i maksimalne vrednosti za obe LED diode, na osnovu kojih se može izračunati odnos R koji se dobija preko sledeće jednačine:

$$R = \frac{\frac{\text{max crvena}}{\text{min crvena}}}{\frac{\text{max infracrvena}}{\text{min infracrvena}}}$$

Dobijeni odnos R se ubacuje u jednačinu za računanje oksigenisanosti gde su konstante A i B dobijene na osnovu empirijske krive i iznose 110 i 25 respektivno:

$$SpO_2 = 110 - 25 * R$$

Svaka dobijena vrednost oksigenisanosti se dodaje na konačnu sumu na osnovu koje će se nakon predefinisane vremena izračunati srednja vrednost kao konačni rezultat oksigenisanosti koji će biti ispisao u terminalu MPLAB-ovog Data Visualizer-a. Za računanje pulsa se koristi algoritam za detekciju i merenja vremena između maksimalnih vrednosti, tj. pikova, u signalu pulsnog oksimetra. Algoritam radi tako što pri pojavi prvog pika restartuje brojač na nulu i istovremeno započinje njegovu

inkrementaciju. Vrednost u brojaču će da raste sve dok se ne pojavi novi pik. Kada se pojavi novi pik algoritam će očitati vrednost sa brojača, restartovati brojač i ponovo započeti njegovu inkrementaciju. Očitana vrednost sa brojača se dodaje na konačnu sumu koja će biti podeljena sa trenutnim brojem detektovanih pikova. Ovim deljenjem se dobija srednja vrednost vremena između dva detektovana pika i ona će biti prosleđena nazad u glavnu petlju gde će se primeniti u računanju vrednosti pulsa. Vrednost pulsa dobija se deljenjem jednog minuta, odnosno šezdeset sekundi, sa prosleđenom srednjom vrednošću.

$$\text{puls} = \frac{60 \text{ s}}{\text{sr. vr. pulsa}}$$

Kako bi algoritam uspešno detektovao svaki pik uvodi se granična vrednost na osnovu koje algoritam zna kada treba da započne detekciju. Ova granična vrednost se dobija tako što se svaka vrednost pika poredi sa 60% od srednje vrednosti svih prethodnih detektovanih pikova. Ovo nam pomaže da odbacimo sve neželjene nagle uspone koji se javljaju u signalu.

6. ZAKLJUČAK

Pulsni oksimetar kao dijagnostička metoda nam pruža rane informacije o problemima isporuke kiseonika koji mogu nastati zbog blokade disajnih puteva, neadekvatne plućne ventilacije, difuzije, cirkulacije, itd. Takođe se može koristiti u različitim situacijama radi brze provere respiratornih disfunkcija. Posebno je pogodan za pacijente sa srčanom bolešću ili astmom, mada se koristi i na pacijentima sa različitim bolestima respiratornog trakta, za proveru nivoa hemoglobina kod pacijenata sa apnejom, kod pušača, kao i kod ljudi koji rade u zagađenim sredinama, poput vatrogasaca. Takođe je vrlo koristan i za zdrave osobe koje jednostavno žele da vrše redovne provere svog zdravstvenog stanja, sportiste i osobe koji vežbaju na velikim nadmorskim visinama kao što su planinari, skijaši, piloti, itd. Napomenuo bih da funkcionalnost pulsnog oksimetra nije idealna, odnosno da na dobijene rezultate utiču mane merne sonde poput pozicioniranja dioda i fotodetektora, pritisak koji štipaljka viši na prst, smetnje izazvane spoljašnjim svetlom pa čak i smetnje izazvane pomerajem. Na rezultate merenja takođe utiču i drugi faktori kao što je karakteristika fotodetektora, anatomija prsta, kalibraciona kriva i brzina rada algoritma. Zbog navedenih mana projekat se ne može porediti sa radom jednog komercijalnog pulsnog oksimetra.

7. LITERATURA

- [1] Design of Pulse Oximeters – J.G. Webster – IOP Publishing Ltd 1997 – ISBN 0-7503-0467-7
- [2] Medicinska Fiziologija – Arthur C. Guyton i John E. Hall – ISBN 978-953-176-785-9
- [3] Encyclopedia Of Biomedical Engineering – Roger Narayan – ISBN 978-0-12-804829-0
- [4] Medical Device Technologies A Systems Based Overview Using Engineering Standards – Gail Baura – ISBN 978-0-12-811984-6

ИНТЕГРАЦИЈА РЕЛАЦИОНЕ И ГРАФОВСКЕ БАЗЕ ПОДАТАКА ЗА ПОДРШКУ РАДА ИНФОРМАЦИОНОГ СИСТЕМА ДРУШТВЕНЕ МРЕЖЕ**INTEGRATION OF RELATIONAL AND GRAPH DATABASES TO SUPPORT THE INFORMATION SYSTEM OF THE SOCIAL NETWORK**Тамара Говорчин, *Факултет техничких наука, Нови Сад***Област – ПРИМЕЊЕНЕ РАЧУНАРСКЕ НАУКЕ И ИНФОРМАТИКА**

Кратак садржај: У овом раду описане су релациона и графовска база података, као и проблеми на које се може наићи приликом коришћења само релационе базе података. Такође су наведени начини на које графовска база може да превазиђе поједине недостатке релационе базе података. На основу свих прикупљених информација, истражен је начин за интеграцију релационе и графовске базе података. У оквиру рада, такође је описан информациони систем друштвене мреже, који представља приказ случаја коришћења решења које је претходно изведено. Циљ изабраног дела информационог система је да олакша процес повезивања корисника у виду изражавања пријатељства у оквиру друштвене мреже, као и да омогући додавање и прегледање фотографија са додатним информацијама. Концептуална шема базе података пројектована је помоћу алата *lucid-app*, а од база података су коришћене MySQL као релациона база података и Neo4j као графовска база података.

Кључне речи: релациона база података, графовска база података, тип ентитета, друштвена мрежа.

Abstract: This paper describes the relational and graph database, as well as the problems that can be encountered when using only the relational database. The ways in which the graph database can overcome some of the issues of the relational database are also listed. Based on all the collected information, a way to integrate the relational and graph database was investigated. In this paper, the information system of the social network is also described, which represents a use case of the solution previously performed. The goal of this information system is to facilitate the process of connecting users in the form of expressing friendship within the network, as well as to enable adding and viewing photos with additional information. The conceptual database scheme was designed using the *lucid-app* tool, and the databases used were MySQL as a relational database and Neo4j as a graph database.

Keywords: relational database, graph database, entity type, social network.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Милан Челиковић, доцент.

1. УВОД

Развојем информационе технологије уочени су недостаци информационог система заснованих на принципу да свака апликација, односно скуп програма за једну област пословања, има своју колекцију података. Систем база података значајно се разликује од овакве концепције, јер подразумева колекцију података која је заједничка за више апликација. Различити начини коришћења информација довели су до експанзије и креирања разних типова база података. Поред релационих база података, све су више заступљене NoSQL, Document, Key-Value, хијерархијске, графовски оријентисане и друге базе података.

Релационе базе података су најчешће коришћене базе података због погодности које пружају. Међутим, могу да се јаве проблеми уколико се користи само релациона база података, што може да утиче и на саме перформансе система. Неке од слабости релационог модела су ограничена способност да експлицитно искаже семантику захтева и скупа операција рекурзивног повезивања табела (енгл. JOIN). Додатно, релациона база података има фиксну шему базе података, што доводи до линеарне зависности броја чворова и веза које се складиште у бази.

Истражен је начин за решавање претходно наведених проблема са којима се може сустрести у току развоја система коришћењем само релационе базе података. Са друге стране, рад се бави истраживањем нерелационе, графовске базе података која може да превазиђе поједине недостатке релационе базе података, с обзиром на то да се структура графовске базе података састоји од чворова и веза, и самим тим је употреба овакве базе података ефикасно решење када је акценат на вези и својствима између чворова. Графовска база података има флексибилну структуру, што може да смањи цену складиштења при великим количинама података.

На основу свих наведених информација, истражен је начин за интеграцију релационе и графовске базе података. Такође је описан информациони систем друштвене мреже, који представља приказ случаја коришћења решења које је претходно изведено. Употребљени су велики скупови података, и за релациону и за графовску базу података. Примарни циљ описаног информационог система, је да пружи могућност корисницима да остварују различите везе са другим корисницима, као и да пружи подршку у процесу додавања нових слика, прегледања профила

корисника, прегледања и измене веза између корисника.

Поред уводног поглавља и закључка, овај рад садржи још шест поглавља, а то су: Преглед актуелног стања у области, Анализа проблема интеграције релационе и графовске базе података, Опис реалног система, Опис шема базе података, Опис архитектуре система и коришћених технологија и Опис апликативног решења.

2. ПРЕГЛЕД АКТУЕЛНОГ СТАЊА У ОБЛАСТИ

У овом поглављу наведени су циљеви коришћења база података, након чега су класификоване и описане базе података. Осим тога представљен је преглед литературе у области којом се бави овај рад. Овај преглед извршен је у циљу бољег разумевања графовске и релационе базе података, као и самог начина функционисања друштвене мреже. Након истраживања јавно доступних радова, детаљније је истражено три рада. Сваки од радова је на свој начин имао значајан утицај у разумевању и дубљем размшљању о начину решавања проблема са којима смо се суочили током израде рада.

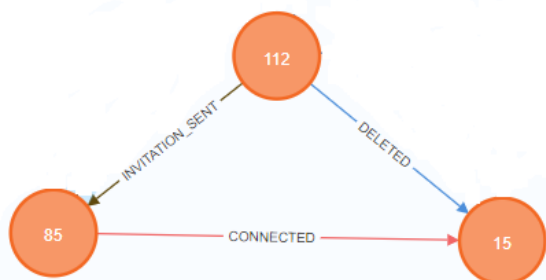
2.1 Класификација база података

База података је систематска колекција података, која подржава електронско складиштење и манипулацију подацима. Базе података олакшавају приступ, управљање и ажурирање подацима. Најчешће коришћене базе података су хијерархијска, мрежна, објектно-оријентисана, релациона и нерелациона (NoSQL) база података.

У релационој бази података, подаци се организују у облику дводимензионалних табела, са врстама и колонама. Свака табела би требала да има примарни кључ (енгл. *primary key*). Примарни кључ означава колону, или више колона, која омогућава сваком реду у табели да се јединствено идентификује.

Графовска база података је NoSQL база података која се користи за моделовање података у облику графа. Чворови графа приказују ентитете, док релације описују повезаност ових чворова.

Пример графа може се видети на слици 1. У оквиру овог рада, релације могу бити различитог типа, као што су обрисано (енгл. *DELETED*), послат захтев (енгл. *INVITATION_SENT*) и повезано (енгл. *CONNECTED*). Свака релација има смер који је означен стрелицом, а тиме се означава од ког чвора је веза усмерена ка којем чвору. Чвор је представљен са идентификационом ознаком корисника.



Слика 1 – Пример графа у Neo4j бази

Уколико би се упоредили релациона база података и графовска база података, однос појмова би изгледао као на слици 2.

Релациона база података	Графовска база података
Табела	Граф
Ред	Чвор
Колона и подаци	Својства са вредностима
Ограничења	Релације
Спајање табела (енгл. <i>JOIN</i>)	Специфицирање жељених кретања кроз граф (енгл. <i>Traversal</i>)

Слика 2 – Однос појмова између релационе и графовске базе података

2.2 Преглед литературе

У раду [7] обрађени су концепти графовске базе података користећи Neo4j. Такође су образложени случајеви у којима је потребно користити графовску базу података.

У раду [8] спроведено је истраживање које пружа упоређивање перформанси и могућности релационе и графовске базе података у складиштењу и обради огромних друштвених података. При повећању количине социјалних података, приказан је однос величина складишта, времена додавања података, као и времена претраге података између MySQL и Neo4j база података.

У раду [9] описан је начин за постизање флексибилног и ефикасног механизма за складиштење података о метапотацима књиге, у циљу креирања система за препоручивање књига.

3. АНАЛИЗА ПРОБЛЕМА ИНТЕГРАЦИЈЕ

РЕЛАЦИОНЕ И ГРАФОВСКЕ БАЗЕ ПОДАТАКА

У оквиру овог поглавља описане су релациона и графовска база података. Након описа ових база података, следи опис конкретних система за управљање базама података. Као систем за управљање релационе базе података описана је MySQL [4] база, док је као систем за управљање графовске базе података описана Neo4j [5] база података. Такође је наведен начин на који су MySQL [4] и Neo4j [5] базе података интегрисане у циљу превазилажења недостатака који постоје када се ове базе података користе самостално, а не у интегрисаној форми.

3.1 Релациона база података

Релациона база података је колекција информација која организује податке у унапред дефинисаним односима где се подаци чувају у једној или више табела. Систем за управљање релационим базама података који је коришћен у оквиру овог рада је MySQL [4].

MySQL [4] је брз, скалабилан и једноставан систем за управљање релационим базама података који, између осталог, подржава:

- имплементирање операција базе података на табелама, редовима, колонама и индексима;
- дефинисање базе података у облику табела, што представља колекцију редова и колона.
- декларативну имплементацију ограничења базе података;
- аутоматско ажурирање индекса табела; и
- коришћење SQL упита и комбиновање информације из више табела за крајње кориснике.

3.2 Графовска база података

Графовска база података је нерелациона база података која се користи за моделовање података у облику графа. Чворови графа приказују ентитете, док релације описују повезаност ових чворова. Систем за управљање графовском базом података која је коришћена у оквиру овог рада је *Neo4j* [5].

Предности *Neo4j* су:

- Пружа флексибилан и једноставан модел података.
- Пружа резултате засноване на подацима у реалном времену.
- Доступан је за велике пословне апликације у реалном времену са трансакцијским гаранцијама.
- Могуће је лако представљање повезаних и полуструктурираних података.
- Пружа декларативни језик упита за визуелно представљање графа под називом *Cypher*. Команде овог језика су у читљивом формату и врло једноставне за учење.
- Не захтева сложена спајања за преузимање повезаних података.

У оквиру информационог система који је описан у овом раду, сваки корисник система може да добије на увид листу препоручених корисника. Препоручен корисник се добија тако што се посматрају сви корисници који имају везу типа *CONNECTED* са тренутним корисником, који има идентификациону ознаку 1. Након тога се посматрају корисници који имају везу *CONNECTED* са претходно пронађеним корисницима који имају везу *CONNECTED* са тренутним корисником. Мора се узети у обзир да тренутни корисник не сме да има везу типа *INVITATION_SENT* са пријатељем његовог пријатеља. Овакав упит може се на видети на слици 3.

```
MATCH (userA: User)-[:CONNECTED]-(userB: User)-[:CONNECTED]-(userC: User)
      WHERE userA.id = 1
      AND NOT (userA: User)-[:CONNECTED]-(userC: User)
      AND NOT (userA: User {id: 1})-[:INVITATION_SENT]-(userC: User)
      AND userC.id <> 1
WITH userC, count(*) as score return userC.id as id ORDER BY score DESC LIMIT 20
```

Слика 3 – Упит који враћа листу препоручених пријатеља

3.3 Анализа интеграције MySQL и Neo4j база података

С обзиром на то да је слабост релационе базе података ограничена способност да експлицитно искаже семантику, као и у скупој операцији рекурзивног повезивања табела, неопходно је предложити решење за интеграцију *MySQL* базе са *Neo4j* базом података, како би графовска база података могла да надомести те слабости у појединим моментима. *Neo4j* је систем за управљање графовском базом података, што би значило да је сваки чвор представљен на одређен начин, а такође се на брз начин може креирати велик број веза различитог типа између корисника. Да би ове две базе података могле да се користе у исто време са синхронизованим подацима, неопходно је издвојити поље у односу на који ће се вршити

интеграција ове две базе података. Као такав податак, предлаже се коришћење поља које је дефинисано у *MySQL* бази података, а такође представља и идентификациону ознаку изабране табеле. Изабрано је поље које означава идентификациону ознаку из разлога што такво поље никада неће мењати своју вредност.

Овакав случај коришћења представљен је кроз пример информационог система друштвене мреже. На овом примеру, поље у односу на који ће се вршити интеграција представља идентификациону ознаку типа ентитета *User*.

Сваки корисник ће имати дефинисану идентификациону ознаку у оквиру *MySQL* базе података, док ће при креирању веза између корисника, бити креиран чвор у *Neo4j* графовској бази података који ће означавати корисника, а биће представљен са том идентификационом ознаком.

4. ОПИС РЕАЛНОГ СИСТЕМА

У претходним поглављима анализирале су базе података и наведени су проблеми који се могу јавити приликом коришћења само релационе базе података. Описане су погодности које графовска база података нуди и на који начин може да надомести недостатке релационе базе података.

Након тога је приказано решење за интеграцију релационе и графовске базе података. У даљем тексту биће описан пример употребе решења за интеграцију релационе и графовске базе података на конкретном случају реалног система друштвене мреже.

Опис реалног система односи се на опис класа корисника, њихових карактеристика и надлежности у оквиру информационог система друштвене мреже. У оквиру система друштвене мреже постоје два типа корисника, обичан корисник и администратор система.

Основне корисничке функционалности које могу да извршавају сви корисници система су регистрација корисника, пријава и одјава на систем и ажурирање профила. Сви корисници такође могу да извршавају и сложеније функционалности, које су приказане на дијаграму случајева коришћења на слици 4. На дијаграму се такође може приметити да администратор система наслеђује обичног корисника.



Слика 4 - Дијаграм случајева коришћења за сложене радње сваког корисника

5. ОПИС ШЕМА БАЗЕ ПОДАТАКА

У овом поглављу представљена је прво концептуална шема базе података, а након тога су описани сви типови ентитета и типови повезника у оквиру модела.

5.1. Модел концептуалне шеме базе података

Концептуални модел шеме базе података заснован је на анализи правилника за делатност система за који се пројектује информациони систем, као и на основу постојећих информационих система.

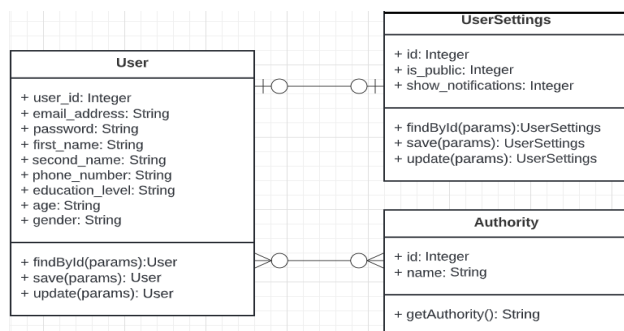
Модел је направљен помоћу алата *lucid-app* [6] и представљен је кроз више делова:

1. део концептуалне шеме базе података за смештање података о корисницима,
2. део концептуалне шеме базе података за смештање података о фотографијама,
3. део концептуалне шеме базе података за смештање података о коментарима,
4. део концептуалне шеме базе за смештање података о реакцијама на фотографијама и
5. део концептуалне шеме базе за смештање корисникових обавештења.

У релациону базу података се смештају подаци који се односе на информације о корисницима, фотографијама, коментарима и подешавања профила, док се у графовску вазу података смештају релације које постоје између корисника.

На слици 5 може се видети део дијаграма концептуалне шеме базе података за смештање података о корисницима. Корисник је представљен са типом ентитета *User* и сваки тип ентитета *User* се везује са нула или више улога корисника представљених типом ентитета *Authority*.

Сваки тип ентитета *User* се везује са тачно једним типом ентитета *UserSettings*, што означава подешавања профила корисника.



Слика 5 - Део дијаграма концептуалне шеме базе података за смештање података о корисницима

На слици 6 приказане су асоцијације типа ентитета *User*.

За сваку асоцијацију наведен је назив асоцијације, референтни ентитет, кардиналитет као и опис саме асоцијације.

Назив асоцијације	Референтирани ентитет	Кардиналитет	Опис асоцијације
Authorities	Authority	*..*	Улоге додељене кориснику
User_settings	UserSettings	1-1	Подешавања налога корисника

Слика 6 - Асоцијације типа ентитета *User*

5.2 Опис типова ентитета и типова повезника у моделу

Након представљања модела, дати су описи обележја у оквиру типова ентитета и типова повезника. Описани су следећи типови ентитета: тип ентитета *Authority*, тип ентитета *UserSettings*, тип ентитета *User*, тип ентитета *Location*, тип ентитета *Image*, тип ентитета *Comment*, тип ентитета *ImageComment*, тип ентитета *Notification* и тип ентитета *Follower*.

На слици 7 се могу видети обележја типа ентитета *User*. За свако обележје се наводи назив, тип податка, да ли сме да буде *null*, да ли је јединствено у оквиру система, као и опис самог обележја.

Редни број	Назив обележја	Тип података	Nullable	Јединствен	Опис обележја
1.	User_id	Integer	He	Да	Идентификациона ознака
2.	First_name	String	He	He	Име корисника
3.	Second_name	String	He	He	Презиме корисника
4.	Email_address	String	He	Да	Е-маил адреса корисника
5.	Password	String	He	He	Шифра корисника
6.	Phone_number	String	Да	He	Број телефона корисника
7.	Age	String	Да	He	Број година корисника
8.	Gender	String	Да	He	Пол корисника
9.	Education_level	String	Да	He	Ниво образовања

Слика 7 - Обележја у типу ентитета *User*

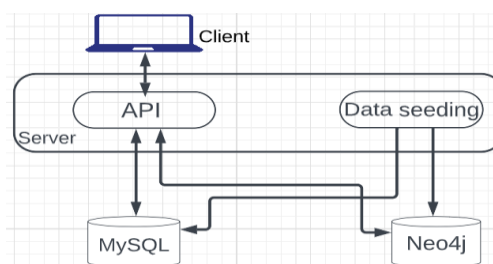
6. ОПИС АРХИТЕКТУРЕ СИСТЕМА И КОРИШЋЕНИХ ТЕХНОЛОГИЈА

С обзиром на то да је апликативно решење замишљено као веб апликација, неопходно је да постоји архитектура система као и технологије помоћу којих је могуће имплементирати овакво решење. У овом поглављу описана је и сликовно приказана архитектура система, а након тога су описане све технологије које су коришћене за израду информационог система кроз апликативно решење.

На слици 8 може се видети приказ архитектуре система апликативног решења за информациони систем друштвене мреже.

Архитектура система се састоји од три дела, а то су: *Client*, *Server* и *Databases*. Клијентска страна, односно кориснички интерфејс, је развијена помоћу *React-a* [3]. На серверској страни постоје два сервиса. Први је *API*, који се односи на инфраструктуру апликативног решења која је развијена помоћу *Spring Boot-a* [2]. Други сервис под називом *Data seeding* представља сервис у оквиру којег се врши рашчлањивање скупова података а након тога и чување података у базе података, а то се врши у оквиру програмског језика *Python* [1].

Апликација користи *MySQL* објектно-релациони систем за управљање базама података, као и *Neo4j* графовску базу података.



Слика 8 – Приказ архитектуре система

7. ОПИС АПЛИКАТИВНОГ РЕШЕЊА

У овом поглављу описано је апликативно решење које обухвата све функционалности описане у претходним поглављима, користећи делове шеме базе података намењене за чување података о корисницима, подешавањима налога, везама између корисника, фотографијама и коментарима. Апликативно решење је замишљено као веб апликација и имплементирано је помоћу технологија које су претходно наведене. Свака функционалност је детаљно описана, наводећи улогу корисника за коју је намењена као и поступак који је неопходно урадити за извршавање исте. Приликом попуњавања форми, наведени су кораци које је неопходно испратити, тако што је наглашено које информације треба унети, као и које дугме кликнути за потврђивање жељене радње.

8. ЗАКЉУЧАК

У овом раду детаљно су истражене и описане употребе релационе и графовски оријентисане базе података, као и начини за њихову интеграцију. Дат је преглед литературе у оквиру које се пореде перформансе релационе и графовске базе података, као и детаљније изучава сама графовска база са системом препоручивања. Такође је описан информациони систем друштвене мреже, који представља приказ случаја коришћења решења које је претходно изведено. Софтверски пакет за подршку рада друштвене мреже подржава кориснике и администраторе система да манипулишу везама са другим корисницима, да прегледају фотографије, коментаре на фотографијама као и друге информације. Информациони систем доприноси прегледности и доступности података и убрзава процес додавања и брисања фотографија, као и додавања коментара на фотографије. Проналажење нових пријатеља је омогућено помоћу система за препоручивање корисника. Могуће проширење рада представља увођење додатних типова података и типова релација, чиме би се додатно увидео значај интеграције база података. Додавање подсистема за продају артикала у виду веб продавнице је потенцијално проширење.

9. ЛИТЕРАТУРА

- [1] Документација везана за Python - <https://www.python.org/>
- [2] Документација везана за Spring Framework - https://en.wikipedia.org/wiki/Spring_Framework#Spring_Boot
- [3] Документација везана за React.js - <https://reactjs.org/>
- [4] Документација везана за MySQL базу података - <https://www.mysql.com/>
- [5] Документација везана за Neo4j базу података - <https://neo4j.com/>
- [6] Веб сајт за креирање дијаграма - <https://lucid.app/documents#/dashboard>
- [7] Justin J. Miller - "Graph Database Applications and Concepts with Neo4j", 2013 - <https://asset-pdf.scinapse.io/prod/776871969/776871969.pdf>
- [8] Chen, Yaowen - "Comparison of Graph Databases and Relational Databases When Handling Large-Scale Social Data", 1989 - <https://harvest.usask.ca/handle/10388/7434>
- [9] D. Naga Jyothi - "Book Recommendation System using Neo4j Graph Database", 2020 - <http://www.ijaema.com/gallery/55-ijaema-june-4051.pdf>

Кратка биографија:



Тамара Говорчин рођена је 1999. године у Новом Саду. Факултет техничких наука, смер Рачунарство и аутоматика, уписала је 2017. године. Дипломирала је 2021. године и исте године уписала мастер студије на Факултету техничких наука, смер Рачунарство и аутоматика.

KATEGORIZACIJA I ANALIZA SENTIMENTA DOKUMENATA UPOTREBOM VIŠEJEZIČNOG TRANSFORMER MODELA SA VIŠE IZLAZA**DOCUMENT CATEGORIZATION AND SENTIMENT ANALYSIS USING A MULTILINGUAL TRANSFORMER MODEL WITH MULTIPLE OUTPUTS**Dušan Milunović, *Fakultet tehničkih nauka, Novi Sad***Oblast – PRIMENJENE RAČUNARSKE NAUKE I INFORMATIKA**

Kratak sadržaj – U radu je predstavljen model za kategorizaciju i analizu sentimenta teksta na sto jezika upotrebom transformer neuronskih mreža. Takođe je pokazan način za optimizaciju takvog modela postupkom destilacije modela.

Ključne reči: Sistemi za analizu teksta, klasifikacija teksta, sentiment analiza teksta, višjejezični transformer modeli, modeli sa više izlaza

Abstract – This paper presents a model for categorization and sentiment analysis of texts in one hundred languages using transformer neural networks. A way to optimize that model using model distillation is also described.

Keywords: Natural language processing, text classification, sentiment analysis, multilingual transformer models, multi-output models

1. UVOD

Oblast obrade prirodnog jezika (*Natural language processing* – NLP) je podoblast lingvistike, računarske nauke i veštačke inteligencije koja se bavi analizom teksta kako bi se iz njega izvukli različiti zaključci. U ovom radu je opisan model mašinskog učenja koji iz teksta može da izvuče sentiment i odredi kategoriju teksta. Sentiment predstavlja „ton“ teksta, koji može biti negativan, neutralan ili pozitivan, a određen je brojem između -1 i 1. Skup kategorija teksta izvučen je iz GCP taksonomije [1]. Izabran je drugi nivo detaljnosti koji sadrži 216 kategorija.

Dodatni zadaci i ograničenja ovog rada su bili da model treba dobro da radi na više jezika i da jedan model određuje i sentiment i kategoriju teksta.

Kako bi se ispunio cilj rada upotrebljen je transformer model XLM-Roberta [2], koji je pretreniran na sto jezika na CommonCrawl [14] skupu podataka. Upotrebom funkcionalnih modela tensorflow [12] biblioteke, kreira se jedan model koji rešava dva problema, kategorizaciju i analizu sentimenta teksta. To je značajno jer je veličina transformer modela njihov problem.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Platon Sovilj.

Na ovaj način, „telo“ transformer neuronske mreže se kombinuje sa dve „glave“ (glava za analizu sentimenta teksta i glava za kategorizaciju) i praktično se količina potrebnih resursa za rešavanje ova dva problema prepolovljava.

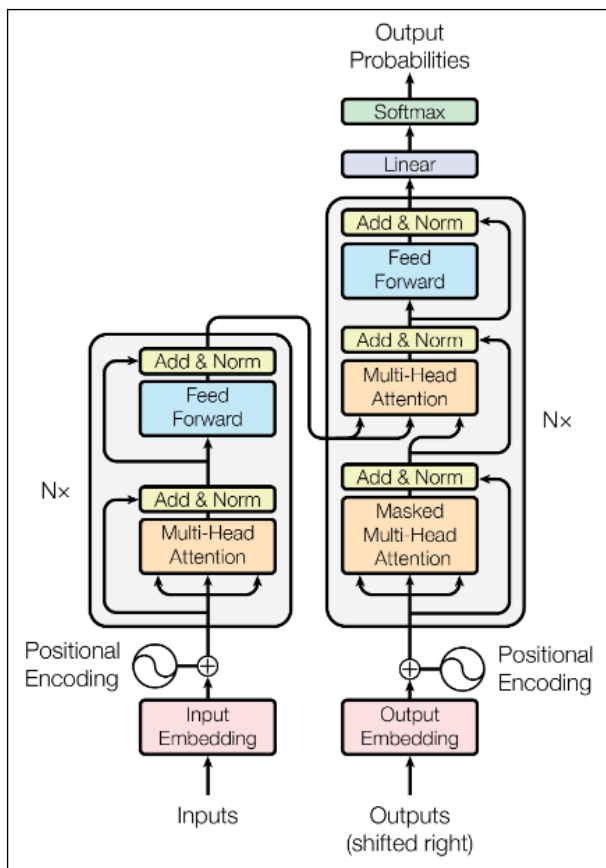
Poslednji problem kojim se bavi ovaj rad je dalja optimizacija modela postupkom destilacije, nalik na postupak predstavljen u radu istraživača huggingface-a [3].

2. TEORIJSKE OSNOVE**2.1. Transformer neuronske mreže**

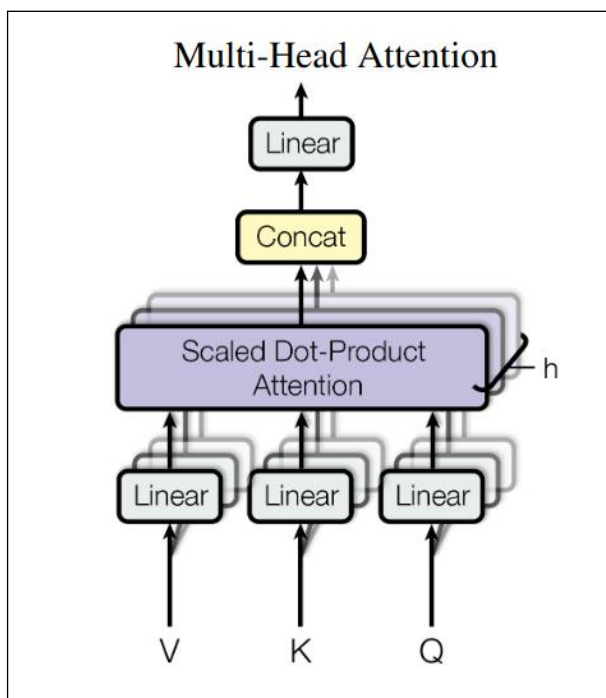
Transformer neuronske mreže [4] predstavljaju najuspešniju arhitekturu mreža za NLP u ovom momentu. Transformer neuronske mreže predstavljaju tip enkoder-dekoder mreža, čija je najčešća svrha da iz niza reči generiše drugi niz reči (npr. u slučaju prevodenja teksta). Za razliku od svojih prethodnika koji su koristili rekurentne neuronske mreže, transformeri izbacuju rekurentnu mrežu iz kombinacije i rade oslanjajući se samo na mehanizam pažnje. Rekurentne mreže imaju problem sa brzinom treniranja zbog svoje sekvencijalnosti, dok transformer mreže omogućavaju da se izračuna rezultat za čitav tekst jednim prolazom kroz mrežu. Ova činjenica omogućava brži trening transformer mreža i otvara vrata za treniranje nad većim skupovima podataka. Na slici 1 prikazana je arhitektura transformer neuronske mreže.

Sa leve strane slike nalazi se enkoder mreža, dok je sa desne strane dekodek. Za početak, vidi se da ulazni podaci i ciljani izlazni podaci prolaze kroz sloj za vektorizaciju (engl. *input/output embedding*). Taj sloj pretvara svaku reč u vektor fiksne dužine (standardne dužine su 128, 256, 512, 768 itd.). Višeglava pažnja (engl. *multi-head attention*) je glavna komponenta u ovoj mreži. Na slici 1. se vidi da se ona pojavljuje i u enkoder i u dekodek delu mreže.

Na slici 2. vidi se šematski prikaz višeglave pažnje. Ona se sastoji iz h ponovljenih komponenti, od kojih se svaka, u slučaju enkoder mreže, naziva samo-pažnja (engl. *self-attention*). Ulazi u mehanizam samo-pažnje su V , K i Q , koji redom označavaju vrednost (engl. *value*), ključ (engl. *key*) i upit (engl. *query*). Poenta mehanizma samo-pažnje je da odredi koji delovi ulaza se odnose jedni na druge. Dobar primer bi bio povezivanje zamenica sa imenicama koje ih predstavljaju, npr. u tekstu „Dečak je pomazio psa. To ga je učinilo srećnim“.



Slika 1. Arhitektura transformer mreže [4]



Slika 2. Višeglava pažnja [4]

Ne postoji jasna definicija na koga se reč „ga” odnosi. Mehanizam samo-pažnje mogao bi da pomogne tako što bi stvorio jaču vezu između reči „ga” i „dečak” ili „ga” i „psa”). Čitava poenta mehanizma samo-pažnje jeste da pametnije enkodira ulazne reči, uzimajući u obzir i ostale reči ulaza. Bitno je primetiti da se kod računanja pažnje

taj račun može izvesti potpuno nezavisno za svaku reč iz ulaza. Ta osobina omogućava paralelizaciju treninga kod transformer modela, koja nije bila moguća kod rekurentnih mreža.

Na slici 1. vidi se da ulaz, osim što ulazi u sloj višeglave pažnje, takođe obilazi taj sloj takozvanom rezidualnom (engl. *residual*) vezom, nakon čega se sabira sa izlazom iz sloja višeglave pažnje i normalizuje. Normalizacija predstavlja način za regulisanje donjih i gornjih granica vrednosti u mreži i služi da se stabilizuje i ubrza trening. Nakon toga, rezultat ovih operacija prolazi kroz potpuno povezanu mrežu. Na isti način kao i ranije, ulaz u tu mrežu takođe ima rezidualnu vezu preko koje se sabira sa izlazom iz potpuno povezane mreže i opet se vrši normalizacija dobijenog zbira.

Nakon što se završi računanje vrednosti svih ulaznih reči u enkoderu, određuje se još jedan par vektora vrednosti V i ključeva K za svaku reč, koji služe kao ulaz u sloj višeglave pažnje u dekodek mreži. Da bi se dobili upiti Q za dekodek mrežu, koristi se mehanizam maskirane višeglave pažnje (engl. *masked multi-head attention*). Jedina razlika između obične i maskirane višeglave pažnje je što se u maskiranoj verziji, povezanost sa rečima koje u rečenici slede posle trenutno obrađivane reči postavlja na 0. Na taj način, sprečava se mogućnost da mreža posmatra reči koje još uvek nije generisala. Ovo je relevantno za vreme treninga transformer mreže, jer kasnije, tokom njene upotrebe, te reči zapravo neće biti unapred poznate. Izlaz iz maskirane višeglave pažnje služi kao vektor upita Q za sloj obične višeglave pažnje u dekodeku. Ostatak mreže radi na isti način kao i kod enkoder mreže. Nakon što poslednji u nizu dekodeka izračuna svoj izlaz, taj izlaz prolazi kroz potpuno povezani sloj sa linearnom aktivacijom i na kraju kroz *softmax* sloj kako bi se odredilo koja će se reč generisati.

Poslednji koncept koji je uveden sa transformer mrežama je koncept pozicionog enkodiranja (engl. *positional encoding*). U do sad objašnjenom delu transformer mreže ne postoji način da mreža uzme u obzir pozicije reči u rečenici. Poziciono enkodiranje rešava taj problem tako što dodaje (sabira) određen vektor vektoru svake ulazne reči. Vektori koji se dodaju zavise od pozicije reči u tekstu i sačinjavaju šablon koji mreža može da nauči. Intuicija iza ove tehnike je da se dodavanjem ovih vrednosti u vektore reči dodaje dovoljna razlika između njih da mreža može da nauči da ih razlikuje po poziciji u rečenici.

2.2. BERT i XLM-Roberta

BERT [5] (*Bidirectional Encoder Representations from Transformers*) predstavlja transformer neuronske mrežu, uz nekoliko velikih promena u odnosu na originalnu transformer mrežu. Glavna je ta što je *BERT* samo enkoder deo transformer mreže. To znači da umesto dekodeka, kao poslednji sloj *BERT*-a može da se izabere neuronska mreža za rešavanje proizvoljnog zadatka. Druga značajna razlika je što je *BERT* pretreniran model, koji služi za učenje prenošenjem znanja (engl. *transfer learning*).

XLM-Roberta [2] je transformer model sličnih osobina kao *BERT*, ali sa ciljem da se koristi kao jedan model koji radi sa sto jezika. Stoga, on je pretreniran na sto jezika,

koristeći *CommonCrawl* program za prikupljanje 2.5 terabajta podataka. Još jedan od glavnih ciljeva ove mreže je postići mogućnost da se trenira za specifičan zadatak na jednom jeziku, a da se nakon toga koristi za isti zadatak na preostalim jezicima. Taj cilj se naziva međujezično razumevanje (engl. *cross-lingual understanding*) i *XLM-Roberta* trenutno predstavlja najbolje rešenje na tom polju.

2.3. Destilacija modela

Pojam destilacije modela je predstavljen u radu [6], gde je osnovna ideja bila da se znanje ansambla modela sakupi u jedan model, koji bi oponašao izlaze čitavog ansambla. Slična ideja je primenjena u polju *NLP* nad *BERT* modelom u radu [3]. Ovde, ideja je bila da se napravi model koji se sastoji od istih delova kao i *BERT*, ali sa manjim brojem slojeva. Rezultat rada je *DistilBERT* model, koji sadrži 40% parametara *BERT* modela, 60% je brži i postiže 97% performansi *BERT*-a na *GLUE Benchmark* [7] zadacima.

3. SKUP PODATAKA

Za analizu sentimenata iskorišćen je otvoren skup podataka sa projekta „*The Stanford Analysis Treebank*” [8]. On predstavlja skup ocena filmova sa označenim sentimentom.

S obzirom da otvoren skup podataka koji navodi kategoriju teksta nije pronađen, korišćena je tehnika pretraživanja interneta za tekstove koji odgovaraju kategorijama. Za početak, izabran je *GCP (Google Cloud Platform)* skup kategorija teksta. On sadrži tri nivoa kategorija po nivou opštosti. Za ovaj rad izabran je drugi nivo opštosti, koji sadrži 216 različitih kategorija. Zatim, izabrane su ključne reči za svaku od kategorija. Nakon toga, korišćenjem biblioteke *mediawiki* [9] pretražena je onlajn enciklopedija *Wikipedia* [10] za svaku ključnu reč i za svaku je pronađeno po 500 najrelevantnijih tekstova. S obzirom da je za slične ključne reči moguće dobiti iste linkove, izbačeni su duplikati. Koristeći istu biblioteku dobavljeni su sadržaji članaka sa *Wikipedia*-e. Time je dobijen skup podataka sa tekstovima na engleskom jeziku sa označenim kategorijama sačinjen od 224.879 primera. Ovaj način prikupljanja podataka je iskorišćen i opisan u radu [15].

4. IMPLEMENTACIJA REŠENJA

Za implementaciju *XLM-Roberta* neuronske mreže upotrebljena je kombinacija biblioteka *huggingface* [11] i *tensorflow* [12]. *Tensorflow* predstavlja radni okvir za kreiranje, treniranje i upotrebu neuronskih mreža u programskom jeziku *Python*. *Huggingface* je biblioteka specijalizovana za transformer neuronske mreže i sadrži veliki broj pretreniranih transformer mreža. *Huggingface* nudi implementacije modela u radnim okruženjima *tensorflow* i *pytorch* [13]. Za implementaciju *XLM-Roberta* modela kao osnova korišćena je klasa *TFXLMRobertaMode* biblioteke *huggingface*. Ona sadrži pretrenirano „telo” mreže i služi za dalje nadograđivanje za bilo koji *NLP* zadatak. Čitava mreža se može posmatrati kao jedan sloj koji služi za izvlačenje karakteristika teksta (engl. *feature extraction*).

Preostali deo mreže je deo za određivanje izlaza. Kako je potrebno odrediti i sentiment i kategoriju teksta, ovde su

kreirane dve „glave” mreže. To je moguće implementirati funkcionalnim modelima *tensorflow*-a. Jedna od njih se bavi određivanjem sentimenta, a druga određivanjem kategorije teksta. Obe glave mreže su napravljene kao potpuno povezane mreže sa dodatkom sloja ispuštanja (engl. *dropout*).

Nakon treniranja *XLM-Roberta* modela, cilj je bio istrenirati manju mrežu koja će postizati slične rezultate koristeći destilaciju modela. Za studentsku mrežu izabran je model *DistilBERT-multilingual*. Razlog za izbor već postojeće mreže je što je treniranje potpuno novog studentskog modela postupak koji zahteva značajnu hardversku moć. Pre treniranja studentskog modela, napravljen je novi skup podataka, tako što je učiteljski model odredio distribucije verovatnoća za kategorije i sentiment za svaki tekst iz prvobitnog skupa podataka. Koristeći novi skup podataka i isti postupak treniranja kao kod učiteljskog modela, uspešno je istreniran studentski model. Rezultujući model sadrži oko 2.05 puta manje parametara od *XLM-Roberta* modela. Brzina novog modela je takođe oko 2 puta veća.

5. EVALUACIJA REŠENJA

Pri treningu modela primenjena je podela skupa podataka na trening, validacioni i testni skup podataka. Za trening podatke uzeto je 70% skupa podataka, za validacione 10%, a za testne 20% podataka. Rezultati u ovom odeljku se odnose na testni skup podataka, odnosno podatke sa kojima model nije radio ranije.

5.1. Evaluacija XLM-Roberta modela

Bitne metrike za treniranje *XLM-Roberta* modela pri treniranju se odvajaju na metrike vezane za kategorizaciju i metrike vezane za analizu sentimenta.

Za analizu sentimenta praćena je prosečna apsolutna greška (engl. *Mean absolute error*), odnosno apsolutna razlika između pravih i izračunatih sentimenata. Dobijena vrednost je 0.102.

Za klasifikaciju teksta korišćene metrike su mikro F-mera, tačnost i top 3 tačnost. Dobijena vrednost mikro F-mere je 0.596, tačnosti 60.1%, a top 3 tačnosti 78%.

U svrhu testiranja kako model radi na različitim jezicima, ručno je označeno 20 primera prikupljenih iz novinskih članaka na srpskom jeziku. Ti primeri su zatim prevedeni na engleski, nemački, španski i italijanski jezik upotrebom *Google Translate* alata. Cilj ovog dela evaluacije bio je dokazati da će model za sve, ili bar većinu jezika računati iste rezultate, bili oni većinom tačni ili većinom netačni. Za analizu sentimenta, vrednosti od -1 do 1 su podeljene u 5 kategorija (vrlo negativno, negativno, neutralno, pozitivno i vrlo pozitivno). Model je u 18 od 20 rezultata dao iste rezultate na svim jezicima. U jednom primeru je dao 4 od 5 istih rezultata, i u jednom 3 od 5 rezultata. Za kategorizaciju teksta posmatrano je da li model u svoja top 3 rezultata daje tačnu kategoriju (ukoliko daje, rezultat se smatra tačnim, a u suprotnom netačnim). Za 14 od 20 primera model je dao isti rezultat na svim jezicima. Za 5 primera je dao 4 od 5 istih rezultata, i u jednom 3 od 5 istih rezultata.

5.2. Evaluacija destilovanog modela

Za destilovani model, prvo su primenjene iste dve mere kao za *XLM-Roberta* model.

Prosečna apsolutna greška pri određivanju sentimenta sa destilovanim modelom je 0.135, što je za oko 0.033 lošije od originalnog modela. Mikro prosek F-mere *DistilBERT-multilingual*-a je 0.587, što je za oko 0.01 lošije od *XLM-Roberta*. Tačnost i top 3 tačnost destilovanog modela su 59.5% i 77.5%. Obe vrednosti su manje od vrednosti originalnog modela za otprilike 0.5%. Sve ove mere ukazuju na dobru uspešnost procesa destilacije.

Za testiranje modela na različitim jezicima, korišćeni su isti primeri kao sa originalnim modelom. Za analizu sentimenta, destilovani model u 15 od 20 primera daje isti rezultat na svim jezicima. U 2 primera daje 4 od 5 istih rezultata, a u preostala 3 primera daje 3 od 5 istih rezultata. Za kategorizaciju model u 8 od 20 primera daje isti rezultat za sve jezike. U 11 primera daje 4 od 5 istih rezultata, i u jednom primeru daje 3 od 5 istih rezultata. Ovo ukazuje da destilovani model nije u mogućnosti da potpuno replicira uspeh originalnog modela u radu sa različitim jezicima.

Poslednja mera koja se posmatrala kod destilovanog modela je koliko često on daje iste kategorije za tekst kao i originalni model. Izračunato je da u 73% slučajeva, oba modela daju potpuno isti rezultat. Ukoliko se posmatra da li se rezultat originalnog modela nalazi među najverovatnija tri rezultata destilovanog modela, dobija se da se u 91.5% slučajeva nalazi. Ove metrike pokazuju da destilovani model dobro oponaša originalni.

6. ZAKLJUČAK

U ovom radu predloženo je rešenje za problem kategorizacije i analizu sentimenta tekstova na sto različitih jezika upotrebom modernih transformer arhitektura.

Model je konstruisan uz oslonac na biblioteku *huggingface* kao osnovu za transformer modele i biblioteku *tensorflow* za prilagođene slojeve koji omogućuju da model ima dve izlazne vrednosti.

Najznačajniji doprinos ovog rada predstavljaju primenjene optimizacije nad transformer modelima. Prva je ta što se “telo” transformer mreže koristi za ekstrakciju karakteristika teksta, i njen izlaz se koristi kao ulaz za dve “glave” mreže koje određuju sentiment i kategoriju teksta.

Druga optimizacija jeste destilacija originalnog modela čija je osnova *XLM-Roberta* transformer model u manji model čija je osnova *DistilBERT-multilingual* model.

Obe optimizacije predstavljaju korak ka dovođenju transformer mreža u stanje u kom su dostupnije za upotrebu u realnom svetu, gde su performanse modela (brzina i zauzeće memorije) direktno vezane za cenu upotrebe i potrošnju električne energije modela.

7. LITERATURA

- [1] GCP taksonomija
<https://cloud.google.com/natural-language/docs/categories> (pristupljeno u septembru 2022.)

- [2] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ... & Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.
- [3] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- [4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).
- [5] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- [6] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- [7] GLUE Benchmark
<https://gluebenchmark.com/>
- [8] The Stanford Analysis Treebank
<https://nlp.stanford.edu/sentiment/>
- [9] Mediawiki
<https://github.com/barrust/mediawiki>
- [10] Wikipedia
<https://www.wikipedia.org/>
- [11] Huggingface
<https://huggingface.co/>
- [12] Tensorflow
<https://www.tensorflow.org/>
- [13] Pytorch
<https://pytorch.org/>
- [14] CommonCrawl
<https://commoncrawl.org/>
- [15] Agarwal, A., Dahleh, M., Shah, D., Sleeper, D., Tsai, A., & Wong, M. (2019). Zorro: A Model Agnostic System to Price Consumer Data. *arXiv preprint arXiv:1906.02420*.

Kratka biografija:



Dušan Milunović rođen je u Novom Sadu 1997. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Primenjene računarske nauke i informatika odbranio je 2022.god.
 kontakt:
 dusanmilunovic17@gmail.com

КОМПАРАТИВНА АНАЛИЗА ПРИМЕНЕ ТРАДИЦИОНАЛНИХ МЕТОДА И ГРАФ НЕУРОНСКИХ МРЕЖА У ОБЛАСТИ СИСТЕМА ЗА ПРЕПОРУКУ**COMPARATIVE ANALYSIS OF THE APPLICATION OF TRADITIONAL METHODS AND GRAPH NEURAL NETWORKS IN THE AREA OF RECOMMENDER SYSTEMS**

Јелена Достих, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – Системи за препоруку релевантних објеката све чешће постају део наше свакодневице и утичу на наше одабیره и активности. Подаци коришћени у системима за препоруку природно могу бити представљени у виду графа у којем кориснике и објекте представљамо као чворове тог графа, а њихове интеракције као везе између њих. Овај рад пореди ефикасност традиционалних система за препоруку и система за препоруку базираних на граф неуронским мрежама. У циљу поређења, имплементирана су два система - хибридни традиционални систем и систем базиран на граф неуронским мрежама. Перформансе, односно резултати експеримената извршених над оба система евалуирани су и упоређени на основу прецизности, одзива и F1 мере, односно средњеквадратне грешке у случају континуалних вредности.

Кључне речи: системи за препоруку, граф неуронске мреже, матрица сличности, граф конволуција

Abstract – Recommendation systems have increasingly become part of our everyday life and influence our choices and activities. Data used in recommendation systems can naturally be represented as a graph in which we present users and objects as nodes of that graph and their interactions as links between them. This paper compares the effectiveness of traditional recommendation systems and recommendation systems based on graph neural networks. For comparison, two systems were implemented - a traditional hybrid system and a system based on graph neural networks. Performance and results of experiments performed on both systems were evaluated and compared based on precision, recall, and F1 measure, but also mean square error in the case of continuous values.

Keywords recommendation systems, graph neural networks, similarity matrix, graph convolution

1. УВОД

Порастом количине информација које нас окружују, појављује се потреба за издвајањем, филтрирањем и рангирањем корисних информација.

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је била др Јелена Сливка, ванр. проф.

Системи за препоруку релевантних објеката све чешће постају део наше свакодневице и утичу на наше одабیره и активности. Објекат препоруке се разликује у зависности од самог домена у којем се систем користи.

Подаци који се користе за креирање система за препоруку су подаци о корисницима, подаци о објектима који се препоручују и прикупљене оцене које су корисници доделили објектима. Овакви подаци природно могу бити представљени у виду графа у којем кориснике и објекте представљамо као чворове тог графа, а њихове интеракције као везе између њих. Граф неуронске мреже представљају један од облика дубоког машинског учења. За улазне податке користе податке структуриране у виду графа, те их можемо искористити приликом креирања система за препоруку.

Овај рад пореди ефикасност традиционалних система за препоруку и система за препоруку базираних на граф неуронским мрежама.

У циљу поређења, имплементирана су два система - традиционални и систем базиран на граф неуронским мрежама. Перформансе, односно резултати експеримената извршених над оба система приказани су у поглављу 4, где су се као основне метрике користиле прецизност, одзив, F1 мера и средњеквадратна грешка у случају континуалних вредности.

Поглавље 2 описује неке од досадашњих приступа који су увели појам граф неуронских мрежа у област система за препоруку. Поглавље 3 се фокусира на методологију и детаљније описује скуп података, а затим и начин на који су добијени и евалуирани резултати креираних система који ће затим бити продискутовани у поглављу 4. На самом крају, у поглављу 5 описани су главни закључци, као и предности и мане које се могу запазити у оба приступа.

2. ПРЕТХОДНА РЕШЕЊА

У овом поглављу ће бити представљено неколико значајних радова који су се бавили архитектуром граф неуронских мрежа, као и њиховом применом у области система за препоруку. Ова област нуди неколико праваца и другачијих начина употребе графова у овој улози.

Хронолошки гледано, међу првима се истакао рад који говори о структури и начину функционисања граф конволутивне мреже (енг. *Graph Convolutional Network, GCN*) [1]. Једно од проширења овог рада је и опис колаборативног филтрирања заснованог на граф архитектури мреже (енг. *Neural Graph Collaborative Filtering, NGCF*) [2]. У њему је објашњено како граф неуронске мреже доприносе побољшању крајњих препорука, увођењем информација о интеракцијама између корисника и објеката.

Мало другачије приступе представљају *GraphSage* [3] и његов наследник *PinSage* [4] алгоритми, који уводе унапређење тиме што у процесу проналажења препорука узимају у обзир само фиксни, мањи подскуп најближих комшија, за разлику од њихових претходника који користе целе, врло често веома велике скупове података.

Унапређење претходних приступа је алгоритам представљен у раду “*Multi-Component Graph Convolutional Collaborative Filtering*” (*MCCF*) [5] који додатно уводи и механизам пажње у процес формирања препорука циљаног чвора. Механизам пажње омогућава разликовање тежине и нивоа утицаја комшијских чворова на циљани чвор.

Поред поменутих радова, један другачији правац креирања препорука представљају системи који приликом креирања препорука уводе и информације о социјалним интеракцијама међу корисницима. У овој категорији најбољи представници су *DiffNet++* [6] и *GraphRec* [7] алгоритми. Сличност коју деле је увођење информација о међусобним интеракцијама корисника, међутим *DiffNet++* користи два одвојена бипартитивна графа за њихову репрезентацију, док *GraphRec* користи један хетерогени граф који другачијим врстама веза прави разлике у горе поменутих интеракцијама.

Још један интересантан приступ који је настао као комбинација две сложеније структуре је *Knowledge Graph Attention Network (KGAT)* [8]. Његове карактеристике су то да користи бенефите механизма пажње, али и повећава комплексност тиме што у репрезентацији графа садржи чворове који не представљају само кориснике и објекте, већ додатно и њихове атрибуте.

3. МЕТОД

Методологија, односно поступак креирања два система за препоруку приказан је у овом поглављу. Један од система заснован је на традиционалним методама, док је други заснован на граф неуронским мрежама. У поглављима 3.2 и 3.3 детаљније је описан сваки од њих. Оба система су развијена у програмском језику *Python*.

3.1. Скуп података

Ради већег фокуса на имплементацији различитих архитектура модела система за препоруку, у експерименталном кораку коришћен је једноставан и често употребљаван *MovieLens* скуп података.

MovieLens представља један од најчешће коришћених скупова података у домену система за препоруку,

углавном примењен у едукативне сврхе као и у истраживачким радовима у циљу поређења перформанси различитих система. Садржи информације о филмовима и корисницима, као и оценама, односно рејтинзима који су додељени филмовима од стране корисника.

Постоји неколико верзија и неколико различитих величина овог скупа података. У овом раду коришћен је скуп који садржи 100,000 забележених рејтинга додељених 1700 различитих филмова од стране 1000 различитих корисника. Тренинг и тест подела података је одрађена у односу 90:10, односно, 90,570 рејтинга је употребљено за обучавање модела, док је преосталих 9,430 употребљено за евалуацију.

Сваки рејтинг описан је идентификатором корисника, идентификатором филма који је оценио и оценом која може имати вредност између 1 и 5.

Корисници су поред идентификатора описани још и бројем година, полом и занимањем, док су филмови описани називом и годином објављивања. Поред тога, за сваки филм постојала је информација којим жанровима припада. У случају традиционалног система базираног на садржају, ради квалитетнијег формирања вектора филмова, додатно су коришћене и кључне речи, односно тагови преузети из новије верзије *MovieLens* скупа података.

3.2. Традиционалне методе

Ово поглавље описује кораке припреме података, креирања два подсистема која ће формирати крајњи модел као и начин на који су они агрегирани формирајући финални хибридни систем за препоруку.

3.2.1 Систем заснован на садржају

Код система заснованог на садржају потребно је користити особине објеката, односно филмова. Како почетни скуп садржи релативно мали број колона, односно особина, било је неопходно одрадити неколико трансформација над њим. У оригиналном формату података, филмови су описани кроз једну особину која је сачињена од назива и године објављивања филма и, поред тога, кроз листу жанрова којима филм припада. Оно што је додатно коришћено у приступу приказаном у овом раду јесте информација о таговима, односно кључним речима, које су поједини корисници додељивали појединим филмовима.

Измене оригиналних података резултовале су новим скупом особина, где је сваки филм садржао назив, годину објављивања, деценију објављивања и као и листу жанрова и листу агрегираних кључних речи у “*bag of words*” формату. Приликом креирања латентних вектора креиране су различите комбинације ових особина. Осим различитих особина које улазе у процес креирања вектора, оно што се разликује јесте и метода којом се они креирају. С обзиром на то да су коришћене особине у текстуалном формату, било је неопходно применити одређену трансформацију текста у нумерички формат. За ове потребе могуће је користити различите векторајзере. У овом раду приме-

њена су два једноставна модела - “Count” и “TF-IDF” векторајзери.

3.2.2 Систем заснован на колаборативном филтрирању

Приликом креирања система заснованог на колаборативном филтрирању, кораци припреме података су поједностављени с обзиром да се користе само информације о рејтинзима. Као што је случај код сваког система за препоруку базираног на колаборацији, креирана је матрица односа корисника и филмова. Након тога формирана је матрица сличности филмова коришћењем *Pearson* мере сличности, на основу рејтинга додељених од стране корисника. Креирана матрица је затим употребљена у циљу проналажења препорука за конкретног корисника на основу филмова које је он до тада оценио. Добијени вектори су затим употребљени за креирање матрице сличности, приликом чега је коришћена косинусна мера сличности. Вредности резултујуће матрице представљале су меру сличности сваког филма са преосталим филмовима из коришћеног скупа података.

3.2.2 Креирање хибридног система

Системи су агрегирани на једноставан начин у циљу смањења проблема хладног почетка (енг. “cold start”), али истовремено и креирања семантички богатијих препорука. Примењено правило агрегације је у основи врло једноставно - након провере броја филмова које је корисник до тада погледао, систем се преусмерава на одговарајући систем.

Уколико би корисник погледао мање од два филма, као резултат издвојени су тренутно најбоље ранжирани филмови. Са друге стране, уколико би корисник погледао више од десет филмова систем је преусмерен на логику колаборативног филтрирања. Коначно, у случајевима где је корисник погледао мањи број филмова, коришћена логика система базираног на садржају.

3.3. Систем заснован на граф неуронским мрежама

Други систем креиран је уз примену граф неуронских мрежа. С обзиром на специфичан формат података над којима граф неуронске мреже могу да се обуче, било је неопходно одрадити одговарајућу трансформацију улазних података, што је уједно и први корак овог приступа који је детаљније описан у поглављу 3.3.1 Припремљени подаци у форми графа се у наставку користе за тренирање модела граф неуронске мреже. Овај корак је детаљније описан у поглављу 3.3.2.

3.3.1 Припрема података

Од почетне форме података, који су описани у поглављу 3.1, на почетку је неопходно креирати њихову нумеричку репрезентацију. Подаци се примарно представљају у виду тензора. Посебно су креирани тензори који представљају *кориснике*, *филмове* и *рејтинге*, а посебно они који представљају њихове особине - *године*, *радно место* и *пол* у случају корисника, односно *жанр* у случају филмова.

Тензори, односно вишедимензионе матрице, су даље коришћени за формирање финалног облика улазних

података, односно, **графова**. Овај корак јесте и један од кључних који прави разлику између граф неуронских мрежа од осталих типова. При имплементацији овог система, коришћена је *DGL*, односно *Deep Graph Library* [9] - библиотека за рад са граф неуронским мрежама. Проблем који је решаван уз помоћ ове библиотеке јесте класификација постојећих веза унутар графа, што би доменски представљало предвиђање оцена које ће корисници доделити филмовима и препоручивање оних са највишом оценом. С обзиром на то да је било неопходно имати одвојене типове чворова за кориснике и филмове, подаци су били представљени у форми *хетерогеног графа*.

3.3.2 Архитектура модела

Први корак унутар модела јесте креирање латентних вектора на основу информација преузетих из улазног графа. Формирани су одвојено латентни вектори корисника и филмова, сабирањем вектора њихових особина. У наставку се креиран латентни вектори прослеђују првом скривеном слоју модела.

3.3.2.1 Скривени слој

Унутар сваког скривеног слоја врши се исти поступак ажурирања вредности прослеђених вектора. Вредности латентних вектора претходног слоја се прослеђују одговарајућој граф конволуцији. У овом случају, с обзиром на то да постоје два типа веза, потребно је креирати *HeteroGraphConv* инстанцу унутар *DGL* библиотеке, односно, за сваки тип везе дефинисати посебан тип конволуције. У случају модела описаног у овом раду, за оба типа веза коришћен је посебно креиран тип конволуције - *GCMCConv*. *GCMCConv*, односно “*Graph Convolutional Matrix Completion*” [10] конволуција представља посебан тип конволуције која узима у обзир постојање свих веза унутар графа, с тим да поједине иницијално немају вредност. Циљ граф неуронске мреже јесте да предвиди вредности веза које недостају.

3.3.3 Тренирање модела

Приликом тренирања модела, поред улазних података, који су трансформисани на начин описан у поглављу 3.2.1, прослеђивани су додатни параметри. За потребе креирања латентних вектора било је неопходно проследити димензију вектора у скривеном слоју *hidden_dim*, али и број корисника *num_users*, филмова *num_items* и рејтинга *num_ratings* коришћених у тренингу. Поред вектора корисника и филмова, креиран су и вектори особина, тако да је било неопходно проследити и број могућих вредности корисникових година *user_age_num*, пола *user_gender_num* и занимања *user_job_num*, као и број жанрова којима филмови могу припадати *movies_genres_num*. Поред димензија, прослеђен је и број скривених слојева *num_layers* који ће формирати финални модел.

Током пропагације у напред, латентни вектори корисника и филмова се сабирају са латентним векторима њихових особина и у таквом формату прослеђују првом слоју модела. Унутар сваког слоја иницијализује се *HeteroGraphConv* објекат, где се дефинише тип конволуције за сваки тип везе. У овом

случају је за оба типа веза искориштена GCMSCov конволуција. Током пропагације у напред, извршеном граф конволуцијом се добијају предикције које се даље користе за израчунавање грешке модела која се затим користи за ажурирање матрице тежина. У овом кораку коришћена је средњеквадратна грешка, док су оптимизационе функције варирале у зависности од експеримента. Резултати експеримената се могу видети у поглављу 4.

4. РЕЗУЛТАТИ И ДИСКУСИЈА

Ово поглавље описује конкретне перформансе модела описаних у поглављу 3 и тиме заокружује компаративну анализу ових система.

У табели 1 су приказани резултати добијени након извршених експеримената над моделима различитих параметарских конфигурација. За процену система су коришћене “Precision”, “Recall” и “F1” мере, добијене на тест скупу. Коришћена је и средњеквадратна грешка “RMSE” ради поређења континуалних вредности.

	Precision	Recall	F1	RMSE
Трад. Систем 1	0.395	0.4	0.397	0.84
Трад. Систем 2	0.395	0.4	0.397	0.84
ГНМ 1	0.257	0.257	0.257	1.262
ГНМ 2	0.337	0.337	0.337	1.262
ГНМ 3	0.351	0.351	0.351	1.256

Табела 1. Перформансе креираних система

Као што се може видети из табеле 1, традиционални системи су надмашили системе креиране употребом граф неуронских мрежа. У случају креирања једноставног система за препоруку над малом количином података, традиционалне методе су се показале као боље, што се свакако може довести у питање уколико би се комплексност података и система повећала. У циљу побољшања добијених перформанси, могуће је унети одговарајуће измене у постојеће системе.

У случају традиционалног приступа, систем је могуће унапредити развојем комплексније логике у позадини хибридног система, односно начина агрегирања једноставнијих система који се налазе у његовој основи. Базни системи су такође креирани на врло једноставан начин и нуде простор за увођење додатне комплексности која би систем учинила прецизнијим. Поред тога, подаци обogaћени додатним особинама могли би знатно да побољшају перформансе, као и сама величина скупа за обучавање.

Системи засновани на граф неуронским мрежама такође могу да пруже боље перформансе у случају богатијих података и опширнијег скупа за обучавање.

5. ЗАКЉУЧАК

С обзиром на велики успех граф неуронских мрежа у појединим областима, било је за очекивати да оне остваре боље перформансе од традиционалних метода. Међутим, у овом раду се показало да то не мора увек бити случај. Граф неуронске мреже, као веома комплексна структура, у неким случајевима уводе додатну сложеност у систем у којем то није потребно. Управо је то оно што показује и овај рад, где је над веома малом количином података једноставне структуре било довољно применити традиционалне методе система за препоруку. Наравно, како будућност доноси све веће количине података, али и захтева прецизније и унапређеније моделе, граф неуронске мреже ће наставити да се употребљавају у све већој мери, како у системима за препоруку, тако и у другим областима.

6. ЛИТЕРАТУРА

- [1] GCN: Semi-Supervised Classification with Graph Convolutional Networks <https://arxiv.org/abs/1609.02907> [2016]
- [2] Neural Graph Collaborative Filtering <https://arxiv.org/abs/1905.08108> [2019]
- [3] GraphSage: Inductive Representation Learning on Large Graphs <https://arxiv.org/abs/1706.02216> [2017]
- [4] PinSage: Graph Convolutional Neural Networks for Web-Scale Recommender Systems <https://arxiv.org/abs/1806.01973> [2018]
- [5] Multi-Component Graph Convolutional Collaborative Filtering <https://arxiv.org/abs/1911.10699> [2019]
- [6] DiffNet++: A Neural Influence and Interest Diffusion Network for Social Recommendation <https://arxiv.org/abs/2002.00844> [2020]
- [7] GraphRec: Graph Neural Networks for Social Recommendation <https://arxiv.org/abs/1902.07243> [2019]
- [8] KGAT: Knowledge Graph Attention Network for Recommendation <https://arxiv.org/abs/1905.07854> [2019]
- [9] Deep Graph Library <https://www.dgl.ai/>
- [10] Graph Convolutional Matrix Completion <https://arxiv.org/pdf/1706.02263v2.pdf>

Кратка биографија:



Јелена Достич рођена је 17.6.1997. године у Новом Саду. Основне академске студије завршила је 2020. године на Факултету техничких наука у Новом Саду, на ком 2022. године брани и мастер рад из области Електротехнике и рачунарства.

GENERISANJE TEKSTUALNOG OPISA SLIKE POMOĆU MAŠINSKOG UČENJA**IMAGE CAPTIONING USING MACHINE LEARNING**Ivan Činčurak, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – Automatsko opisivanje slike je postalo atraktivna tema u poslednjih nekoliko godina. Postoji velika potreba mašinskog opisivanja situacija u automobilske industriji. Google Image pretraga bi takođe mogla biti poboljšana. Takođe, moguće bi bilo unaprediti nadzorne kamere uklanjanjem potrebe za postojanje osobe koja bi konstantno morala da nadgleda kamere i čeka da se određena situacija desi, umesto da pogleda samo kada je opis slike na videu približan nekom unapred definisanom skupu tekstova. U ovom radu isprobana su tri načina za automatsko generisanje opisa slike. Prvi je primenom enkoder-dekoder arhitekture sa mehanizmom pažnje, drugi je bez ovog mehanizma, dok je treći upotrebom rekurentnih neuronskih mreža. Rešenje je evaluirano metrikama BLeU, ROUGE i Doc2Vec. Modeli su trenirani i testirani na MSCOCO skupu podataka. Dodatno, model je testiran podacima scrape-ovanim sa Google Images pretrage.

Ključne reči: opisivanje slika, metrike za sličnost teksta, enkoder-dekoder arhitektura, mehanizam pažnje

Abstract – Image captioning has become an attractive topic in the last few years. There is a substantial need for machine description of situations in the automotive industry. Google Image search could be improved. Also, it would be possible to enhance surveillance cameras by eliminating the need for a person to constantly monitor the cameras and wait for a certain situation to occur instead of only looking when the description of the image in the video is close to some predefined set of texts. In this paper, three methods for automatic image description were tested. The first is by applying the encoder-decoder architecture with the attention mechanism, the second is without this mechanism, and the third is by using recurrent neural networks. The solution was evaluated with BLeU, ROUGE, and Doc2Vec metrics. The models were trained and tested on the MSCOCO dataset. Additionally, the model was tested with scraped data from a Google Image search.

Keywords: image captioning, text similarity metrics, encoder-decoder architecture, attention mechanism

1. UVOD

Razvoj tehnologije svakodnevno donosi nove izazove. Svedoci smo neverovatnog razvoja automobilske

industrije koja želi da što je više moguće olakša vožnju svojim korisnicima. Jedna od stvari koja pomaže u tome je automatsko opisivanje okoline u kojoj se nalazi auto. Ovo bi moglo biti od izuzetne važnosti za vozača koji zbog smanjene vidljivosti ne može tačno da proceni šta se nalazi ispred njega na putu. Isto tako bi automobil uz pomoć dobrog mehanizma za opisivanje slika mogao da ga upozori na opasnost ukoliko se nešto neočekivano desi ispred automobila.

U ovom radu isprobane su metode predložene u radovima koji će biti predstavljeni u nastavku i koji su rešavali problem automatskog generisanja opisa slike. Isprobana je enkoder-dekoder arhitektura sa i bez mehanizma pažnje [2, 3] i rekurentna neuronska mreža čiji ulaz predstavljaju informacije dobijene upotrebom konvolutivnih neuronskih mreža koje su prethodno trenirane.

U radu je eksperimentisano sa varijacijom ovih metoda. Varijacije uključuju korišćenje raznih modela koji su predstavljali enkoder. Za dekoder su korišćene LSTM i GRU arhitekture. Pored arhitekture, eksperimentisano je sa različitim načinima pretprocesiranja teksta.

Za potrebe treniranja je korišćen MSCOCO skup podataka [7]. Za svaku sliku korišćeno je pet različitih opisa. Radi poređenja sa referentnim radovima, modeli su testirani nad MSCOCO skupom podataka i pokazalo se da su rezultati dosta slični, u proseku pet procenta razlike.

Dodatno, model je testiran podacima scrape-ovanim sa Google Images pretrage. Kao što je očekivano, model je dao lošije rezultate nad ovim skupom nego nad MSCOCO. Ovakvo ponašanje je normalno jer se prilikom evaluacije nad MSCOCO skupom podataka za svaku sliku koristilo pet različitih opisa za istu sliku što je modelu davalo veći dijapazon mogućih načina za opisivanje slike.

Prilikom testiranja modela nad scrape-ovanim slikama postojao je samo jedan mogući opis, te je logično da je model davao lošije rezultate. Prilikom testiranja su korišćene BLeU metrika, kao i ROUGE i Doc2Vec uz kosinusnu sličnost vektora.

2. PREGLED STANJA U OBLASTI

U radu [1] opisano je rešenje problema automatskog generisanja opisa slike primenom mehanizma pažnje koji je ubačen u enkoder-dekoder arhitekturu. Enkoder predstavlja CNN koja sliku reprezentuje vektorom. Ovaj vektor predstavlja ulaz u dekoder koji predstavlja RNN. Prilikom treniranja korišćen je vokabular veličine 10000 reči. Pri evaluaciji rešenja [1] korišćen je MSCOCO,

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Aleksandar Kovačević.

Flickr30k i Flickr8k skup podataka. Evaluacija rezultata je vršena upotrebom BLeU i Meteor metrika. Rezultati koji su dobijeni su 67 i 20,30 nad Flickr8k, 66,9 i 18,49 nad Flickr30k i 71,8 i 23,9 nad MSCOCO skupom podataka. Pored generisanja opisa slike, u ovom radu analizirano je koje tačno piksele model razmatrao prilikom formiranja opisa slike. Ovakva analiza je moguća zbog upotrebe mehanizma pažnje.

U radu [2] je prikazano još četiri rešenja problema automatskog generisanja opisa slike zasnovano na enkoder-dekoder arhitekturi sa i bez mehanizma pažnje, primenom *multimodal learning*-a kao i kompozicionih arhitektura. Autori su koristili skup podataka kao što je u prvom radu [1]. Ukupan skup podataka je podeljen tako da je 5000 slika korišćeno za testiranje i validaciju iz MSCOCO skupa, dok je po 1000 slika uzeto iz Flickr skupova. Korišćene metrike su: BLeU, Meteor, ROUGE-L i CIDEr. Zanimljivo je napomenuti da su rezultati dobijeni enkoder-dekoder arhitekturom sa mehanizmom pažnje identični kao u prvom radu [1]. Najbolje rešenje dalo je poslednje od četiri navedena rešenja. Dobijenu su sledeći rezultati: 0,72, 0,248, 0,53, 0,95 gde brojevi odgovaraju redom navedenim metrikama.

U radu [3] je rešavan problem automatskog generisanja slike bez mehanizma pažnje. Cilj je bio napraviti LSTM model koji prima sliku i niz reči i vrši predikciju reči koja ima najveću verovatnoću da se pojavi sledeća u rečenici. Za evaluaciju su korišćene BLeU, Meteor, CIDEr, SBU i Pascal VOC. Rezultati dobijeni u ovom radu su dosta slični onima dobijenim u prethodna dva [1, 2].

Rad [4] predstavlja rešavanje ovog problema upotrebom GAN mreže. Arhitektura se sastojala iz dva dela, generatora i evaluatora. Prvi od njih je služio kako bi se napravio opis slike, dok je drugi deo tu kako bi odredio koliko dobro generisana rečenica opisuje ulaznu sliku. Prilikom treniranja korišćeni su MSCOCO i Flickr30k skupovi podataka. Za evaluaciju su korišćene sledeće metrike: BLeU-3, BLeU-4, Meteor, ROUGE-L, CIDEr i SPICE kao i dve dodatne metrike E-GAN i E-NGAN koje su detaljno opisane u ovom radu. Dobijeni rezultati su redom: 0,39, 0,19, 0,24, 0,52, 1, 0,2, 0,52, 0,62.

U radu [5] pokušano je rešavanje problema upotrebom samo konvolutivnih neuronskih mreža bez dekodera. Kao ulaz model je primao feature iz slike upotrebom pretrenirane VGG16 mreže i vektor dobijen od rečenice. Model za pretvaranje reči u vektore je treniran od početka. Skup podataka koji je korišćen je MSCOCO a metrike koje su korišćene su BLeU, Meteor, ROUGE-L, CIDEr i SPICE dok su rezultati redom: 0,713, 0,247, 0,525, 0,912, 0,172. Može se primetiti da su autori uspeli ovim pristupom da dobiju rezultate kao i u prethodnim radovima.

Radovi [1, 2, 3] predstavljaju osnovu za ovaj rad i daju uvid u to kakvim rezultatima treba da težimo ukoliko bismo želeli da automatizujemo proces opisivanja slike.

Ostali radovi [4, 5] nisu implementirani u ovom radu jer je akcenat bio na istraživanju mehanizma pažnje u enkoder-dekoder arhitekturama i koliko on zapravo unapređuje rešavanje problema kojim se bavi ovaj rad.

3. METODOLOGIJA

U ovom poglavlju opisani su implementacioni detalji rešenja za automatsko opisivanje slika. Na početku poglavlja je opisan skup podataka, kao i prikupljanje podataka za test skup. Potom sledi deo koji se odnosi na obradu podataka. Poglavlje se završava navođenjem arhitekture rešenja.

3.1. Skup podataka

Za rešavanje problema, prilikom treniranja korišćen je MSCOCO skup podataka. Sastoji se od 83000 slika za treniranje i 41000 za validaciju. Ukupna količina podataka je 19GB. Svaka slika ima 5 različitih opisa koji joj odgovaraju. Za dodatno testiranje modela je korišćen ručno napravljen skup podataka. U svrhu pravljenja skupa za testiranje koristila se tehnika *scrape*-ovanja. Sa sajta <http://www.google.com/imghp> su skidane slike po sledećem principu. Kod je uzimao rečenicu iz unapred definisanog skupa rečenica od 100 primeraka i išao na navedeni sajt. Nakon učitavanja rezultata skidana je prva slika koju je Google izbacio i to je skladišteno u direktorijum gde je naziv svake slike odgovarao rečenici koja je uneta u cilju pretrage na navedeni sajt.

3.2. Obrada podataka

U radu su isprobana dva različita pristupa obradi podataka pre treniranja. Prvi pristup podrazumeva uklanjanje znakova interpunkcije. Druga obrada je bila nešto složenija. Pored osnovne obrade korišćene u prvom pristupu, dodato je uklanjanje stop reči i svodenje reči na njihov koren.

Obrada podataka je vršena i nad slikama čiji se opis generisao. Svaka slika se najpre smanjivala na dimenzije 299×299 , a zatim se vršila normalizacija slike tako da svaki piksel ima vrednost od -1 do 1.

3.3. Arhitektura rešenja

U cilju generisanja opisa slike korišćena su tri različita pristupa. Prvi pristup koji je isproban je korišćenje enkoder-dekoder arhitekture podržane mehanizmom pažnje. Prvi deo predstavlja CNN koja kao ulaz prima sliku, dok kao izlaz daje *feature* koji je dobijen na osnovu te slike. Drugi deo predstavlja RNN koja prima izlaz iz prethodnog sloja, odnosno vektor i daje na izlazu rečenicu koja opisuje sliku. Za CNN korišćeni su pretrenirani modeli kao što su: *InceptionV3*, *EfficientNet*, *Retina* i *VGG*. Ovi modeli su morali biti donekle izmenjeni u odnosu na originalne verzije. Potrebno je bilo skinuti poslednji sloj koji kao ulaz prima vektor, a kao izlaz vrši kvalifikaciju. Mreže nije bilo potrebno dodatno trenirati.

Kada su u pitanju RNN, u našem slučaju je korišćen model koji ima više ulaza i jedan izlaz (ulaz je sekvenca reči, a izlaz jedna reč). Za ovaj pristup je korišćen GRU i LSTM model. Rezultati ova dva modela su bili skoro identični, pa će zbog toga u daljem radu biti prikazani samo rezultati GRU modela koji se pokazao bolje.

U drugom pristupu pravljen je arhitektura koja ima CNN i RNN samo bez mehanizma pažnje. Cilj ovog pristupa je bila provera koliko stvarno ovaj mehanizam pomaže modelu u opisivanju slika.

Treći pristup je opisivanje slike upotrebom LSTM-a. Slika se kao i u prethodnim primerima pušta kroz CNN, koja nije deo celog modela. Ona nam služi samo kako bi dobili vektore od slika za potrebe treniranja. Ono što je zanimljivo na ulaz LSTM modela pored ovog vektora, dovodi se niz reči. Ako zamislimo početak predikcije, model će na ulaz dobiti vektor slike i reč koja predstavlja početak rečenice. U našem slučaju je to bila reč START.

Model vrši predikciju i kao izlaz daje reč koja je najverovatnija da se nađe nakon prosleđene reči. Sada se u naredni ulaz stavlja ponovo vektor slike ali se sada se stavlja i reč START i ona koja je dobijena od strane modela. Postupak se ponavlja dokle god ne dodemo do oznake koja predstavlja kraj rečenice, END, ili dok model ne izbaci broj reči koji smo odredili da je maksimalan mogući broj reči u opisu.

4. EKSPERIMENTI I REZULTATI

Prilikom evaluacije su korišćene sledeće metrike: BLeU, ROUGE-1, ROUGE-2, ROUGE-L, Doc2Vec + kosinusna sličnost.

4.1. BLeU metrika

U ovom radu je korišćen sledeći način računanja sličnosti uz pomoć BLeU metrike. Računate su metrike gde je n pripadalo skupu brojeva 1, 2, 3, 4 i zatim je uzimana srednja vrednost tih metrika. Ovaj način računanja preuzet je iz rada [4]. Naravno, n odgovara dužini sekvenci reči (n-gram).

Tabela 1. Rezultati korišćenjem BLeU metrike

Mehanizam pažnje	Obrada podataka	Model	Skor
DA	Osnovno	Inception + GRU	0,44
DA	Osnovno	EffNet + GRU	0,36
DA	Napredno	Inception + GRU	0,39
DA	Napredno	EffNet + GRU	0,3
NE	Osnovno	Inception + GRU	0,27
NE	Osnovno	EffNet + GRU	0,25
NE	Osnovno	Retina + GRU	0,26
NE	Osnovno	VGG + LSTM	0,4
NE	Osnovno	Inception + LSTM	0,4

Tabela 1. sastoji se od četiri kolone. Prva kolona govori o tome da li je u eksperimentu korišćen mehanizam pažnje ili ne. Druga kolona odgovara obradi podataka. Pod osnovnom obradom se smatra uklanjanje znakova interpunkcije dok se naprednom smatra uklanjanje stop reči i svođenje reči na njihov koren.

Treća kolona služi za opisivanje modela, odnosno kaže koji se model koristio za enkoder a koji za dekoader.

Četvrta predstavlja BLeU skor. Bitno je napomenuti da poslednja dva reda u tabeli odgovaraju eksperimentima gde pored slike u model ulazi i sekvenca reči.

Ono što se može primetiti je da napredna obrada podataka samo pogoršala rezultate. Još jedna zanimljiva činjenica koja se može shvatiti na osnovu ovih rezultata je da se model bez mehanizma pažnje najgore ponaša. Naravno, kao što je i očekivano, kombinacija Inception + RNN + mehanizam pažnje se pokazala kao najbolja kombinacija i postignu su rezultati slični kao u relevantnom radu.

4.2. Doc2Vec + kosinusna sličnost

Nedostatak BLeU i ROUGE metrika je da ne mogu verodostojno izračunati sličnost između rečenica ako su reči drugačije ili sinonimi ili ako je redosled reči drugačiji i time promenjen smisao rečenice. Pristup predstavljen u radu [6] je najbliži realnoj proceni kvaliteta.

Rezultati dobijeni nad MSCOCO skupu je 84% što govori da je nad tim slikama model dosta uspešno radio i davao dosta dobre opise.

Tabela 2. Rezultati nastali primenom Doc2Vec metrike

Mehanizam pažnje	Obrada podataka	Model	Skor
DA	Osnovno	Inception + GRU	0,84
DA	Osnovno	EffNet + GRU	0,82
DA	Napredno	Inception + GRU	0,78
DA	Napredno	EffNet + GRU	0,72
NE	Osnovno	Inception + GRU	0,75
NE	Osnovno	EffNet + GRU	0,68
NE	Osnovno	Retina + GRU	0,65
NE	Osnovno	VGG + LSTM	0,83
NE	Osnovno	Inception + LSTM	0,8

U tabeli 2. su prikazani rezultati dobijeni istim eksperimentom kao u prethodnoj tabeli, ali primenom Doc2Vec metrike. Ono što je i očekivano je da je ova metrika dala mnogo bolje rezultate nego prethodne dve.

4.3. ROUGE metrika

ROUGE je još jedna poznata metrika za proveru kvaliteta mašinski proizvedenih rečenica [5]. Postoje razne varijacije ove metrike od kojih su u ovom radu korišćene ROUGE-1, ROUGE-2 i ROUGE-L. ROUGE-N, gde u našem slučaju N predstavlja jedna i dva, predstavlja preklapanje n-grama između prave rečenice i one koja je proizvod modela. ROUGE-L predstavlja izračunavanje najduže sekvence reči koja se nalazi u obe rečenice. Ono po čemu se ova metrika razlikuje od BLeU je to što je bazirana na *recall*-u a ne na *precision*-u.

Rezultati dobijeni nad MSCOCO skupom su 0,7, 0,35 i 0,5. Poslednji broj se odnosi na metriku koja je korišćena i u nekim navedenim radovima [2, 4, 5]. Rezultat koji smo dobili je samo za manje od par procenata lošiji nego rezultati iz tih radova.

U tabeli 3. prikazani su rezultati dobijeni istim eksperimentom kao u prethodne dve tabele, ali primenom tri prethodno navedenih metrika.

Tabela 3. Rezultati nastali primenom ROUGE metrike

Mehanizam pažnje	Obrada podataka	Model	ROUGE-1	ROUGE-2	ROUGE-L
DA	Osnovno	Inception + GRU	0,62	0,26	0,58
DA	Osnovno	EffNet + GRU	0,57	0,25	0,58
DA	Napredno	Inception + GRU	0,58	0,22	0,54
DA	Napredno	EffNet + GRU	0,51	0,2	0,51
NE	Osnovno	Inception + GRU	0,53	0,2	0,53
NE	Osnovno	EffNet + GRU	0,49	0,15	0,49
NE	Osnovno	Retina + GRU	0,5	0,18	0,48
NE	Osnovno	VGG + LSTM	0,55	0,23	0,55
NE	Osnovno	Inception + LSTM	0,51	0,2	0,51

Prva stvar koja se zaključuje iz tabele 3. je ta da je međusobno poređenje modela paralelno donekle poređenju BLEU metrikom. Vidimo da se i ovde može doći do istog zaključka koji se model najbolje a koji nagore ponašao. Zanimljiva stvar koja se može primetiti je da ROUGE-2 metrika davala dosta niske rezultate, što znači da se sekvence reči dužine dva iz tačnih rečenica nisu pojavljivale mnogo u kandidatskim rečenicama. Na osnovu toga se može zaključiti da bi se za ROUGE-3, 4 i svaki sledeći dobila još manja tačnost.

5. ZAKLJUČAK

U ovom radu su opisana tri načina rešavanja *Image captioning* problema. Ispostavilo se da je pristup sa enkoder-dekoder arhitekturom podržanom mehanizmom pažnje dao najbolje rezultate. Prikazani su različiti načini pripreme podataka za treniranje modela kao i različiti modeli koji su se trenirali. Iz svega navedenog u radu je jasno da je ovaj problem rešiv i da je moguće istrenirati modele koji će davati opise koji stvarno odgovaraju slikama. Možemo primetiti da je u ovom radu dostignuta tačnost od 84% iako model nije treniran na više od 20 epoha, što daje osnovu za dalje istraživanje.

U cilju popravljivanja rezultata moguće se fokusirati na neku određenu oblast odnosno kategoriju slika. Recimo mogao bi da se odvoji takav skup podataka koji bi sadržao samo slike ljudi i njihovih aktivnosti. Ovakav pristup bi sigurno doveo do značajnog povećanja tačnosti modela, ali bi isto tako izgubio na svojoj generičnosti. Naravno, pored svega navedenog moguće je pokušati sa drugačijim arhitekturama koje mogu rešiti ovaj problem.

Kao dalje razvijanje moguće je ugraditi ovaj model u softver koji prati dešavanja koja se dešavaju na nekom snimku tako što bi vršio opisivanje *frame*-a na svakih nekoliko sekundi. Jasno je da su mogućnosti neograničene.

6. LITERATURA

- [1] Xu, Kelvin, et al. "Show, attend and tell: Neural image caption generation with visual attention." International conference on machine learning. PMLR, 2015.
- [2] Bai, S., & An, S. (2018). A survey on automatic image caption generation. *Neurocomputing*, 311, 291-304

- [3] Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2016). Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4), 652-663.
- [4] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. *International journal of computer vision*, 115(3), 211-252.
- [5] Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (pp. 311-318)
- [6] Xia, P., Zhang, L., & Li, F. (2015). Learning similarity with cosine similarity ensemble. *Information Sciences*, 307, 39-52
- [7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ra-manan, P. Doll ar, and C. L. Zitnick. Microsoft coco: Com-mon objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

Kratka biografija:



Ivan Činčurak rođen je 8. januara 1998. godine u Novom Sadu, Srbija. Fakultet tehničkih nauka upisao je 2016. godine na studijskom programu Računarstvo i automatika. Diplomski rad odbranio je 2020. godine. Master akademske studije upisao je iste godine na usmerenju Inteligentni sistemi. Kontakt: ivan.cincurak98@gmail.com

CHATBOT NAMENJEN ZA POMOĆ STUDENTIMA PRILIKOM UPISA NA FAKULTET

CHATBOT INTENDED TO HELP STUDENTS WHEN ENROLLING IN FACULTY

Tanja Drčelić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu je vršena evaluacija chatbot-a namenjenog za pomoć studentima prilikom upisa na fakultet. Evaluacija je vršena pomoću neuronske mreže, algoritma mašinskog učenja. Skup podataka je kreiran za dati problem od strane autora rada. Vršeno je pretprocesiranje upita radi bolje predikcije, a rezultati su obrađivani i dopunjavani informacijama iz baze podataka. Izmerene su preciznost, pouzdanost i F1 mera.

Ključne reči: Chatbot, virtualni asistent, mašinsko učenje, neuronska mreža

Abstract – In this paper, the chatbot was evaluated for helping students when enrolling in faculty. The evaluation was performed using a neural network, an algorithm of machine learning. The data set was created by the author. The query was preprocessed for better prediction, and the results were processed and supplemented with information from the database. Accuracy, reliability, and F1 measurements were measured.

Keywords: Chatbot, virtual assistant, machine learning, neural network

1. UVOD

Chatbot je softverska aplikacija za komunikaciju sa korisnikom. Naziva se i virtualni asistent. Virtualni asistent pomaže korisnicima da lakše dođu do željenih informacija. Studenti uvek radije potraže pomoć referenta studentske službe ili rukovodioca departmana umesto da informacije potraže na zvaničnom sajtu fakulteta.

Virtualni asistent se može kreirati na mnogo načina. Rad je zasnovan na evaluaciji pomoću neuronske mreže. Neuronska mreža je modelovana pomoću Keras [1] biblioteke. Korisnik može da komunicira sa softverom tekstualnim ili govornim putem, a odgovor koji korisnik dobija je isključivo tekst. Korišćena je F1 mera za evaluaciju.

Problem je sagledan sa strane korisnika, odnosno studenta koji želi da se upiše na fakultet. Skup podataka je napisan u JSON formatu i čini ga ukupno 161 podatak. Skup podataka je kreiran od strane autora rada.

Ovaj rad se sastoji od 7 poglavlja. U poglavlju 2 je opisan razvoj chatbot-a kroz istoriju i aktuelna rešenja na osnovu kojih je napisan rad. U poglavlju 3 su opisani pojmovi neophodni za razumevanje evaluacije neuronske mreže koja je glavni deo softvera napravljenom za chatbot-a.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kovačević, red. prof.

Poglavlje 4 prolazi kroz metode evaluacije sistema, a poglavlje 5 govori o eksperimentalnoj evaluaciji. U poglavlju 6 se nalaze informacije o rezultatima i diskusija. Poglavlje 7 je zaključak.

2. PREGLED STANJA U OBLASTI

Postoji više pristupa za kreiranje dobrog chatbota. Kroz radove [2] i [3] se može zaključiti da su najbitnije stavke prilikom planiranja softvera skup podataka i evaluacija. Tokom samog modelovanja treba pažljivo odabrati model koji će se koristiti.

ELIZA [4,7] koji simulira psihoterapiju. Nastao je 1960 te godine. ELIZA od korisnika prima jednu ili više rečenica koje razdvaja na reči sa razmacima. Zatim posmatra sve reči zajedno, a zatim od njih pravi spisak ključnih reči. Za svaku ključnu reč traži rečenicu koja se podudara. Ukoliko ne pronade onda nju zanemari i posmatra ostale ključne reči. Na osnovu njih posmatra obrasce i na kraju obavlja skup zamen. Pomenuti rad je zasnovan na skripti. Posmatrajući novije radove, zaključak je da rad isključivo sa skriptama nije najbolja praksa iz razloga što je za unapređenje konverzacije neophodan stalan rad sa skriptom.

A.L.I.C.E. (Artificial Linguistic Internet Computer Entity) [5,7] je besplatan softver chatbot-a kreiran 1995te godine. Kreirana je u AIML (Artificial Intelligence Markup Language). U pitanju je markap jezik koji predstavlja derivat XML-a. AIML je jednostavan. Podržava reči, razmake, _ i *.

Virtualni asistent Alexa [6,7] je implementirana 2014 te godine. Namena je pomoć korisnicima u obavljanju zahteva glasovnim putem. Primer naredbe:

- "Alexa, pusti muziku"

RASA [8] je virtualni asistent za obradu prirodnog jezika koji je zasnovan na DIET modelu. RASA NLU i RASA Core su dva modula koja ga čine. Ulazna rečenica se pretprocesira i zaključuje se namera korisnika i entitet. Zatim se ulazna rečenica, namera i podatak šalju u jezgro radi daljeg procesa.

Rad [9] opisuje kreiranje skupa podataka na način da je najlakše kreirati model neuronske mreže, kao i obraditi podatke. Za skup podataka, istraživanjem se najbolje pokazala pouzdana neuronska mreža.

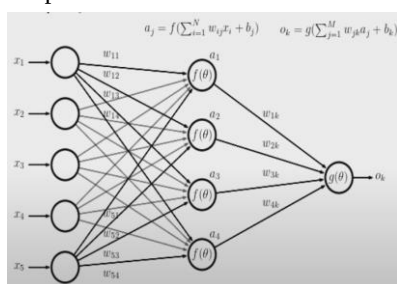
3. TEORIJSKI POJMOVI I DEFINICIJE

U ovom poglavlju objašnjeni su osnovni pojmovi neophodni za razumevanje procesa evaluacije chatbot-a.

U prvom poglavlju je objašnjenje neuronske mreže, a u drugom opis Keras biblioteke.

3.1. Neuronska mreža

Umesto neurona u računarstvu se spominju perceptroni. Prvi radovi na ovu tematiku su nastali 1943 godine. Arhitektura perceptrona koristi prostiranje signala u jednom smeru („feedforward”), a funkcioniše na bazi supervizorskog učenja. Na slici 1. je predstavljena šema perceptrona. Svaka neuronska mreža ima jedan ili više ulaza i izlaza. U ovom slučaju ulaza u neuronsku mrežu ima 5 i to su h_1, h_2, h_3, h_4 i h_5 , a izlaz iz neuronske mreže je samo jedan o_k . Postoje ulazni sloj, skriveni sloj i izlazni sloj. Skrivenih slojeva može biti više i u zavisnosti od broja skrivenih slojeva se meri složenost neuronske mreže. U nastavku teksta će se podjednako koristiti izrazi neuron i perceptron.



Slika 1. Neuronska mreža [3]

Jačina veza između ulaznog sloja neurona i skrivenog sloja neurona treba slučajno izabrati po nekom zakonu verovatnoće, a njihove vrednosti treba fiksirati tokom celog procesa učenja. Zatim se težinski odnosi između skrivenog sloja neurona i jedinog izlaznog neurona podešavaju tokom algoritma učenja.

3.2. Keras biblioteka

Keras [1] je otvorena biblioteka napisana u programskom jeziku Python i namenjena je pravljenju neuronske mreže.

Neke od stavki koje čine biblioteku posebnom su [14]:

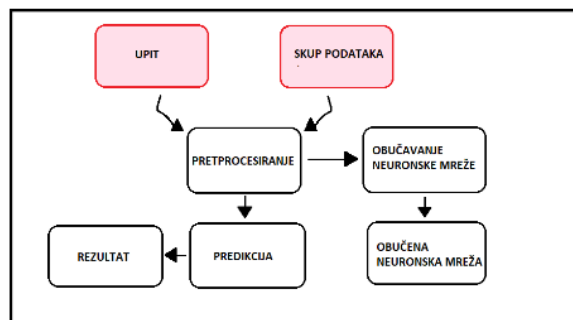
- fokus na korisničko iskustvo,
- brzo usvajanje u industriji,
- podržava više-platformsko programiranje,
- lako razumljiva,
- podržava brz prototipiranje,
- neprimetno radi na centralnom procesoru i grafickom procesoru,
- pruža slobodu dizajniranja bilo koje arhitekture,
- laka proizvodnja modela.

4. METODOLOGIJA I ALATI

Softver namenjen za pomoć studentima prilikom upisa na fakultet zahteva veliku složenost. U narednom poglavlju su opisani sistemi od kojih je softver sačinjen.

U poglavlju će biti reč o strukturi sistema, u drugom o strukturi podsistema neophodnog za obradu upita od strane korisnika ili skupa podataka, u trećem o neuronskoj mreži i u četvrtom o alatima nad kojima je rad baziran.

4.1. Struktura sistema



Slika 2. Šematski prikaz strukture sistema

Skup podataka sadrži uređene rečenice kao odgovore u zavisnosti od pitanja. Podatke nije potrebno dodatno preuređivati, ali je neophodno razvrstati ih tako da se omogući obučavanje neuronske mreže. Obučena neuronska mreža se čuva i dalje se koristi za predikciju. Upit koji korisnik uputi mora se preprocesirati kako bi predikcija bila tačnija. Na osnovu predikcije određuje se rezultat koji se vraća korisniku.

Na slici 3. prikazana je konverzacija korisnika sa virtuelnim asistentom. Korisnik je odabrao da će se razgovor obavljati putem teksta. Zelenom bojom je označen upit korisnika, a svaki par upit-odgovor je razdvojen sa '--'. Svaki put kada se upit unese aktivira se podsistem za preprocesiranje. Ulaz u sistem je upit korisnika. Vršiti se preprocesiranje upita, odnosno priprema teksta za predikciju.

Nakon podsistema za preprocesiranje prelazi se u sistem za neuronsku mrežu i obavlja se predikcija. Obraduje se rezultat predikcije i korisniku se postavljaju dodatna pitanja u zavisnosti od odgovora koji je dobijen na osnovu predikcije. Izlazi se iz podsistema za obradu rezultata i čeka novi upit. Svaka putanja do informacije na sajtu fakulteta je dobavljena iz baze podataka u procesu obrade. Odgovor korisniku je izlaz iz podsistema za obradu rezultata.

```

Moje ime je Četko. Ja sam virtuelni asistent spreman da Vam pomognem.
--
Korisnik:
Dobro sam. Hvala na pitanju. Kako mogu da Vam pomognem?
--
Virtuelni asistent:
Da li Vas zanima način polaganja prijemnog ispita?
http://www.ftn.uns.ac.rs/3492633526/dokumentacija-za-upis
--
Korisnik:
Šta Vas zanima?
--
Virtuelni asistent:
Koji departman Vas zanima?
--
Virtuelni asistent:
Departmani fakulteta se nalaze na sledećem linku:
http://www.ftn.uns.ac.rs/316113793/departmani-fakulteta
--
Korisnik:
Nadam se da sam Vam pomogao.
--
Virtuelni asistent:
Da li Vas zanimaju akademske ili strukovne studije?
Virtuelni asistent:
http://www.ftn.uns.ac.rs/n216788275/fakultet-tehnickih-nauka
--
Virtuelni asistent:
Dokumentacija neophodna za prijemni ispit se nalazi na sledećem linku
http://www.ftn.uns.ac.rs/3492633526/dokumentacija-za-upis
--
Korisnik:
Ugodan ostatak dana Vam želim.
--

```

Slika 3. Kompleksnija primena chatbot-a

4.2. Podsistem za pretprocesiranje

Podsistem za procesiranje je sastavljen iz dva dela. Deo sistema koji se bavi pretprocesiranjem upita od strane korisnika je jedan deo sistema, dok je drugi deo sistema namenjen pretprocesiranju skupa podataka.

4.3. Podsistem za neuronsku mrežu

Podsistem za neuronsku mrežu je sastavljen iz dva dela. Prvi deo predstavlja obučavanje neuronske mreže, a drugi deo predikciju. Ulaz u sistem za obučavanje sistema je trening skup podataka koji predstavlja listu parova (upiti, tagovi) koje imaju vrednost 0 ili 1.

4.4. Podsistem za izbor rezultata

Podsistem za obradu rezultata obrađuje rezultat predikcije na osnovu određene logike i za rezultat šalje tekst korisniku. Na taj način je simulirana smisljena konverzacija između dvoje ljudi. Odgovori su generički. Na osnovu predikcije se određuje i uzima konkretan odgovor iz skupa podataka, a na kraju obrade se dopunjuje u zavisnosti od potrebe dobavljanja podataka koji se nalaze u bazi podataka, a ne u skupu podataka.

4.5. Korišćeni alati

Chatbot je kreiran pomoću Python programskog jezika. Verzija je 3.7. U tabeli 1. je prikazana lista biblioteka koje se koriste i njihovih verzija.

Tabela 1. Korišćene biblioteke i njihove verzije

BIBLIOTEKA	VERZIJA
<i>json</i>	2.0.9
<i>numpy</i>	1.21.6
<i>nlTK</i>	3.7
<i>keras</i>	2.10.0
<i>speech_recognition</i>	3.8.1

5. EKSPERIMENT

Chatbot namenjen za pomoć studentima prilikom upisa je evaluiran u nekoliko koraka. Prvo je osmišljen skup podataka, a zatim i kreiran za konkretan problem. Nakon toga je evaluirana neuronska mreža nad kojom se vrši predikcija. Sistem je razvijen da generički daje odgovore na osnovu predikcije pomoću neuronske mreže. Predikcija nije izlaz iz celog sistema, već iz dela sistema o kojima je u prethodnom poglavlju bila reč. U ovom poglavlju se kroz prvo poglavlje opisuje evaluacija skupa podataka, a kroz drugi evaluacija neuronske mreže i računanje preciznosti, povrata i F1 mere.

5.1. Skup podataka

Skup podataka koji je korišćen u ovom radu kreiran je od strane autora konkretno za dati problem upisa studenata na fakultet. Na osnovu posmatranja studenta koji želi da se upiše na fakultet osmišljeno je nekoliko dijaloga na osnovu kojih je nastao skup podataka. Posmatrane su i skraćenice i žargoni. Nije se posmatrao neobavezan

razgovor koji ne služi za dobijanje konkretnih informacija, ali su upotrebljene fraze za pozdrave. Podaci koji čine skup podataka su sastavljeni uz pomoć reči za upite, a rečenica za odgovore.

Skup podataka je predstavljen u JSON formatu. Organizacija je preuzeta iz rada [9]. Na taj način je skup podataka pregledniji i njegovo korišćenje u neuronskoj mreži je lakše. Skup podataka je na srpskom jeziku i ukupan broj podataka je 161.

Intents predstavlja listu očekivanih pitanja i odgovora za određene tagove. Tagovi su tekstualni zapisi koji opisuju pitanja i odgovore. Patterns je lista pitanja za određeni tag, a responses je lista odgovora u okviru određenog taga. Pitanja nisu gramatički ispravna pitanja poput "Da li mi možete reći više o upisu na fakultet?", već su korišćene ključne reči i fraze, poput "upis" i "upis na fakultet".

```
{
  "intents": [
    {
      "tag": "pozdrav",
      "patterns": [
        "zdravo", "cao", "ćao",
        "dobro dan", "dan",
        "dobro jutro", "jutro",
        "dobro veče", "veče", "dobro veče",
        "veče", "pozdrav", "poz", "hi", "haj",
        "kako se zoveš", "kako se zoveš", "ko si ti"
      ],
      "responses": [
        "Moje ime je Četko. Ja sam virtuelni asisten  
spreman da Vam pomogne."
      ]
    }
  ]
}
```

Slika 3. Deo skupa podataka

5.2. Evaluacija neuronske mreže

Korišćen je Sequential model kod kojeg se slojevi pričvršćuju u sekvencijalnom redosledu za razliku od Functional modela koji nudi veću fleksibilnost po pitanju organizacije slojeva. Postoji pet slojeva u neuronskoj mreži, a broj perceptrona se smanjuje po slojevima. Za minimalni broj perceptrona je uzet broj tagova, jer se na taj način ne gubi na evaluaciji neuronske mreže. Ukoliko se Dropout slojevi izbace dolazi do overfitting-a. Iz tog razloga su Dropout slojevi ubačeni između Dense slojeva.

Prilikom treniranja neuronske mreže, skup podataka koji je korišćen se nasumično "promeša" i nakon toga je spreman za upotrebu. Ova akcija nije neophodna, ali se pokazala kao veoma korisna. Nakon predikcije se uzimaju samo verovatnoće veće od 0.3.

U ovom sistemu, preciznost predstavlja udeo pronađenih relevantnih odgovora u pronađenim odgovorima, a povrat je udeo pronađenih relevantnih odgovora u svim mogućim odgovorima. Uzimaju se u obzir odgovori koji imaju verovatnoću veću od 0.3, nezavisno od procenta.

Tabela 2. Statistika rezultata trening skupa

BROJ TAČNIH ODGOVORA	BROJ NETAČNIH ODGOVORA	UČESTALOST
1	0	53 puta
1	1	11 puta
1	2	10 puta

Na osnovu tabele 2. izračunate su preciznost, povrat i F1 mera. U tabeli 3. su prikazani rezultati. Preciznost i povrat su dosta dobri. Preciznost je velika, jer od mogućih odgovora vraća najviše dva netačna od tri, a povrat je mali iz razloga što nad velikim skupom on uvek vraća jedno tačno rešenje.

Tabela 3. *Rezultati F1 mere*

T,N	PRECIZNOST	POVRAT	F1 MERA
1,0	1	0,0312	0.061
1,1	0,5	0,0156	0.03
1,2	0,33	0,01	0.019

Dobijeni rezultati su dobri iz razloga što za vrednosti verovatnoća većih od 0.3 uvek vraća jedan tačan rezultat. Problem nastaje da li tačan rezultat ima najveći procenat. Takav slučaj može da se desi u primeru gde je moguć isti upit za različite tagove, a od konteksta zavisi koji rezultat je tačan.

6. REZULTATI I DISKUSIJA

Nije neophodno uneti identične upite kao što su očekivani u skupu podataka. Neuronska mreža je dovoljno dobro obučena da u većini slučajeva vraća smislen odgovor.

Chatbot ima usku primenu i očekivane upite, a rezultati su odlični. Problem može nastati ukoliko korisnik unosi upite koji nisu predviđeni projektovanjem. U tom slučaju bi odlično rešenje bilo uvođenje odgovora u slučaju da je unet nerazumljiv upit. Takvo rešenje daje ELIZA [4] i omogućava nastavak interakcije. Takođe, dobro rešenje je i uvođenje opcija korisniku da bira između odgovora i na taj način se sprečava unošenje nerazumljivog upita.

U slučaju traženja preciznosti od korisnika, takav vid zaštite je primenjen, ali nije pokriven slučaj za potpuno nerazumljive upite. RASA [7] daje odličan primer preusmeravanja korisnika. Za većinu upita automatski daje mogućnost korisniku da bira između nekoliko odgovora.

Alexa [7] omogućava komunikaciju na više jezika. Ovaj rad je fokusiran na fakultet u Srbiji i iz tog razloga je moguće prepoznavati samo srpske reči. U Srbiji postoje fakulteti koji primaju strane studente i omogućavaju im studiranje na engleskom jeziku. Iz tog razloga bi sistem bio bolji ukoliko bi se dodalo i prepoznavanje engleskog jezika.

8. ZAKLJUČAK

Virtuelni asistent kreiran pomoću neuronske mreže predstavlja moćan alat za dobavljanje preciznih informacija. Univerzalnost primene se ogleda u tome što je modelovana neuronska mreža za jedan fakultet primenjiva i za podatke bilo kog drugog fakulteta. Fokus je na tipu korisnika.

Komunikacija sa korisnikom može biti narušena sa lošim skupom podataka ili parametara prilikom modelovanja neuronske mreže. Dobar skup podataka je temelj dobro obučene neuronske mreže.

Svaki softver je uvek moguće unaprediti u nekom pravcu, pa tako i ovaj. Jedan od načina unapređenja softvera jeste uvođenje još nekog algoritma. Moguće je modelovati Bajesovu mrežu koja će se primeniti pre predikcije i na taj način će se predvideti šta je korisnik hteo za rezultat iako nije jasno definisao zahtev. Takođe, moguće je dodati i logiku koja će u zavisnosti od doba dana korisniku poželeći laku noć ili dobro jutro bez obzira šta je korisnik naveo kao upit.

Odlična baza podataka za ovaj vid softvera jeste ElasticSearch. Osim što su podaci indeksirani, lako se može pristupiti i pretražiti fajl u njima i na taj način se može dobiti konkretna informacija, a ne samo putanja do nje.

9. LITERATURA

- [1] Keras biblioteka (pristupljeno u oktobru 2022.)
- [2] Adamopoulou E. Moussiades L. , “An Overview of Chatbot Technology“, 2020.
- [3] Nimavat K., Champaneria T., “Chatbots: An overview. Types, Architecture, Tools and Future Possibilities“, 2017.
- [4] J. Weizenbaum, “ELIZA--A Computer Program For the Study of Natural Language Communication Between Man and Machine,” Commun. ACM, vol. 9, no. 1, pp. 36–45, 1966.
- [5] B. A. Shawar and E. Atwell, “A comparison between Alice and Elizabeth Chatbot Systems,” Univ. Leeds, 2002.
- [6] Amazon Developer, “Alexa.”, <https://developer.amazon.com/alexa> (pristupljeno u septembru 2022.)
- [7] “The History of Chatbots – From ELIZA to ALEXA“, https://onlim.com/en/the-history-of-chatbots/?fbclid=IwAR0Stk0fQ4754cPNep16Z5Q_PAUwo93Bm1zYhRnFq1QGR-GuSE9-I9bPrM (pristupljeno u septembru 2022.)
- [8] RASA: Open source conversational AI, <https://rasa.com/> (pristupljeno u septembru 2022.)
- [9] NeuralNine, “Intelligent AI Chatbot in Python” <https://www.youtube.com/watch?v=1lwddP0KUEg&t=1247s> (pristupljeno u septembru 2022.)

Kratka biografija:



Tanja Drčelić rođena je u Novom Sadu 1998. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika odbranio je 2022.god.

Kontakt: ta.drcelic@gmail.com

**АНАЛИЗА ПОДАТАКА СА ПЛАТФОРМЕ JOBERTY.RS ПОМОЋУ ТЕХНИКА
МАШИНСКОГ УЧЕЊА****DATA ANALYSIS OF JOBERTY.RS DATA USING MACHINE LEARNING
TECHNIQUES**

Митар Перовић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – Од великог значаја за компаније јесте да имају увид у ставове и мишљења како свог колектива, тако и оних из других фирми. Са друге стране, онима који траже посао овакве информације помажу да открију предности и слабости компанија које их интересују. У овом раду креиран је скуп података о српским ИТ фирмама користећи рецензије запослених, бивших запослених и кандидата, као и осталим информацијама са сајта Joberty.rs. Извршене анализе над овим скупом подељене су у три задатка. У првом задатку су одређени најзначајнији аспекти компаније уз посматрање овог задатка као регресионог проблема са просечном оценом компаније као циљном лабелом. Резултати указују на то да су аспекти као што су флексибилност и однос послодавца према запосленима значајнији у односу на коришћене технологије и плату. Извршена је анализа сентимента коментара који се тичу процеса селекције употребом БЕРТ модела. Закључено је да HR и технички интервју имају подједнак утицај на формирање утиска о процесу селекције. Коришћењем модела насумичних шума истрениран је модел који врши предикцију процента запослених који сматрају да је плата фер, на основу оцена бенефита које компанија нуди. Постигнут је резултат од 0.1 у контексту метрике MSE. Над истренираним моделом је одрађена анализа важности улазних обележја.

Кључне речи: joberty.rs, анализа задовољства запослених, машинско учење, ит фирме

Abstract – Employee satisfaction is essential for any organization to progress successfully and achieve its goals. Companies need to have insight into the attitudes and opinions of both their team and those from other companies. On the other hand, such information helps job seekers to discover the strengths and weaknesses of the companies they are interested in. In this paper, a data set of Serbian IT companies was created using reviews of employees, former employees, candidates, and other information from the Joberty.rs website. The analyzes performed on this set are divided into three tasks.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је била др Јелена Сливка, ванр. проф.

In the first task, the most significant aspects of the company were determined while observing this task as a regression problem with the average rating of the company as the target label. The results indicate that the employer's flexibility and attitude towards the employees are more significant than used technologies and salary. Sentiment analysis of comments regarding the selection process was performed using the BERT model. It was concluded that HR and the technical interview have an equal influence on forming the impression about the selection process. Using the RandomForest model, a model was trained that performs prediction of the percentage of employees who think that the salary is fair, based on the evaluation of the benefits offered by the company. A score of 0.1 MSE was achieved. An analysis of the importance of the input features was performed on the trained model.

Keywords joberty.rs, data analysis, machine learning, it companies

1. УВОД

Анализа задовољства запослених на основу рецензија са интернета постало је могуће захваљујући платформама за рецензирање. Као најпопуларнији репрезент ових платформи, истиче се Америчка компанија Glassdoor.com. На овој платформи је од 2008. године објављено преко 45 милиона рецензија за око 830 хиљада фирми.

Овај рад се бави анализом рецензија запослених и извлачења различитих увида о важности компанијских бенефита, задовољству процесом селекције, као и груписању фирми на основу оцена. За потребе рада креиран је скуп података о српским ИТ фирмама користећи се рецензијама запослених, бивших запослених и кандидата, као и осталим информацијама са сајта Joberty.rs. Извршене анализе над овим скупом подељене су у три задатка.

У првом задатку су одређени најзначајнији аспекти компаније, уз посматрање овог задатка као регресионог проблема, где су улазна обележја оценом компаније, а просечна оцена компаније представља циљну лабелу. Резултати указују на то да су аспекти као што су флексибилност радног места и однос са колегама значајнији за задовољство запослених у односу на коришћене технологије и плату.

Други задатак је предикција процента запослених који сматрају да је плата фер на основу оцена бенефита

које компанија нуди. Искоришћен је *RandomForest* модел и постигнут је резултат од 0.1 просечне суме квадрата грешке (енг. *Mean Squared Error*, MSE). Након евалуирања модела, анализирани су бенефити компаније на основу важности улазних обележја.

У трећем задатку, извршена је анализа сентимента коментара који се тичу процеса селекције употребом БЕРТ модела. Закључено је да *HR* и технички интервју имају подједнак утицај на формирање утиска о процесу селекције.

У следећем поглављу описана је релевантна литература и еволуција техника за анализу задовољства запослених и битних аспеката компаније. Методологија је описана у поглављу 3 – описано је како су коришћени алгоритми и на који начин су евалуирани експерименти. Експерименти и постигнути резултати су описани у поглављу 4. На самом крају, у поглављу 5, укратко је описана методологија рада са најзначајнијим резултатима и потенцијалним побољшањима.

2. ПРЕТХОДНА РЕШЕЊА

У овом поглављу бавићемо се радовима сличне тематике, као и самом еволуцијом области и значајним резултатима и увидима. Нажалост, како су експерименти у овом раду издељени на више мањих задатака везаних за специфичан скуп података на српском језику, за већину задатака не постоји адекватна литература.

Као најважнији задатак, за који постоји релевантна литература, издвојио се задатак рангирања аспеката компаније по важности. Релевантни радови су тражени на основу тематике (анализа рецензија запослених) и издвојени су на основу године објаве и броја цитата. Радови који су узети у разматрање имају макар 10 цитата и објављени су 2016. године или касније.

Већина наведених радова се ослања на рецензије са сајта *Glassdoor.com*, као најсличнијој платформи са које су преузети подаци за овај рад. Превасходно питање које се поставља је да ли су ове рецензије валидне за озбиљнију анализу задовољства запослених. У раду [1], аутори су показали да су оцене са овог сајта валидно средство процене задовољства запослених. Јавно доступне податке о анализи задовољства запослених за оквирно 40 Америчких државних организација, аутори су упоредили са оценама на *Glassdoor-у*. Крајњи резултат показује да су рецензије добар, али не савршен, медијатор за анкете о задовољству запослених. Дошли су до закључка да нема индиција да корисници користе онлине рецензије као издвни вентил, већ да се држе што ближег искреног рецензирања.

Један од првих битнијих радова који је заснован на рецензијама са *Glassdoor-а* јесте [2]. Аутори су испитивали повезаност задовољства запослених са пословним резултатима компаније. У раду је пре тога показано да се мишљења потрошача и анализа сентимента онлине рецензија, твитова и блогова, могу

користити за предикције обима продаје, као и цене деоница неке фирме [2]. Преузимањем 257,454 рецензија са *Glassdoor-а*, прикупили су скуп података. Након прикупљања, извршили су пречишћавање података, уклањање мање битних речи (енг. *stopwords*), као и стеминг (енг. *stemming*). Из рецензија су издвојили компанијске вредности које се највише спомињу, попут: интегритета, тимског рада, иновација, квалитета, заједнице, комуникације и награде. Коришћењем *bag-of-words* приступа, индексирани су све рецензије и извукли број појављивања речи за сваку категорију и кључне речи. На основу кључних речи су рецензије додељене некој од категорија (компанијске вредности). На основу броја појављивања ових вредности у рецензијама су направили регресиони модел. За сваку од фирми су прикупили податке о тржишној вредности и укупној вредности имовине компаније. На основу тих података су израчунали *Q* фактор (тржишна вредност / укупна вредност имовине) и употребили као зависну променљиву. Након уклопљеног регресионог модела, извршили су анализу важности фактора и дошли до закључка да су компанијске вредности тимски рад, поштовање и иновација у позитивној корелацији са циљном променљивом. Мана овог приступа јесте начин додељивања категорија свакој од рецензија и што нису коришћене софистициране технике анализе текста. Међутим, овај рад је отворио врата многим будућим радовима који користе јавно доступне податке са сајтова за рецензирање, наспрам класичних анкета о задовољству запослених.

У каснијем раду, аутори су издвојили десет најбоље оцењених компанија, као и три најгоре оцењене компаније из ИТ сектора на основу текстуалних рецензија са *Glassdoor.com-а* [3]. За анализу текста коришћен је *IBM Watson* [4], који на основу улазне реченице издваја пар кључних фраза. Затим су намапирани фразе на вредности компаније. Најчешће помињани аспекти су: друштвена вредност (задовољство запослених у раду са другима), интересантност (колико су изазовни задаци у оквиру посла, колико се користе нова решења, итд.), апликативна вредност (у којој мери су њихова знања и вештине примењене).

3. МЕТОД

У наредним поглављима изложени су скуп података и начин на који су спроведени експерименти за три експеримента.

3.1. Скуп података

Подаци су прикупљени са повлачењем (eng. *scraping*) са сајта *Joberty.rs*. Прикупљени су подаци о 389 компаније које послују у Србији. Скуп података је подељен у више подскупова, зависно са које странице су прикупљени подаци. Прикупљени су следећи подаци:

- Оцене запослених различитих аспеката компаније попут: услова рада, радне атмосфере, односа послодавца према запосленима, плата, пројекти, итд.

- Коментари о фирми - оцена и текстуални коментар запослених о самој компанији.
- Присутност бенефита које компанија нуди - рад од куће, плаћени курсеви, храна, итд. Поред бенефита, са исте странице прикупљен је и проценат запослених који мисле да је плата фер.
- Минималне, просечне и максималне плате за одређене позиције у оквиру фирме.
- Оцена процеса селекције, оцена тежине интервјуа, текстуални коментари о техничком и HR делу интервјуа.

3.2. Одређивање важности аспеката компаније

Проблем је моделован као регресиони проблем, а не класификациони, из разлога што бисмо дискретизацијом података потенцијално изгубили део информација. Циљна лабела је просечна оцена компаније, док су улаз у модел оцене аспеката компаније попут: оцене плате, флексибилности (баланс између приватног и пословног дела живота), однос послодавца према запосленима и остали. За регресиони модел су трениране насумичне шуме. Критеријум поделе података у стаблима је гини нечистоћа (eng. *gini impurity*). Хиперпараметри су оптимизовани помоћу мрежасте претраге (eng. *grid search*), и то следећи:

- ***n_estimators*** (број стабала, у распону од 5 до 20, са кораком пет)
- ***max_depth*** (максимална дубина стабла, у распону од два до девет)

Коришћена је мрежаста претрага са унакрсном валидацијом. Унакрсна валидација је рађена над десет делова (eng. *folds*), и на крају је издвојен само најбољи модел. Метрика евалуација је *MSE*. Над најбољим моделом је извршена анализа важности обележја (аспеката компаније).

3.3. Предикција процента запослених који мисле да је плата фер

Предикција процента запослених који сматрају да је плата фер је вршена над скупом података о бенефитима сваке компаније. Први корак је био претворити број корисника који потврђују постојање одређеног бенефита за неку компанију у процентуалну вредност. Ово је урађено дељењем те вредности са колоном "*review_count*", која представља укупан број корисника који су оцењивали постојање бенефита за ту компанију. Након тога је испробано седам различитих комбинација колона које представљају улаз у модел. Колоне које имају велики број недостајућих вредности (више од две трећине) су избачене. Комбинације колона су прављене насумично.

Задатак модела је да предиктује проценат корисника који сматрају да је плата фер. Тренирани су модели *Support Vector Regressor*, *XGBoostRegressor*, линеарна регресија и насумичне шуме. Коришћена је унакрсна валидација над 10 делова док су хиперпараметри оптимизовани помоћу мрежасте претраге. Модели су евалуирани и поређени коришћењем *MSE* метрике. Најбољи резултат је остварен коришћењем модела

насумичних шума. За модел насумичних шума је извршена претрага следећих хиперпараметара:

- ***n_estimators*** (број стабала, у распону од 4 до 128, са кораком степена двојке)
- ***max_depth*** (максимална дубина стабла, у распону од два до 32, са кораком степена двојке)

3.4. Анализа сентимента коментара о процесу селекције

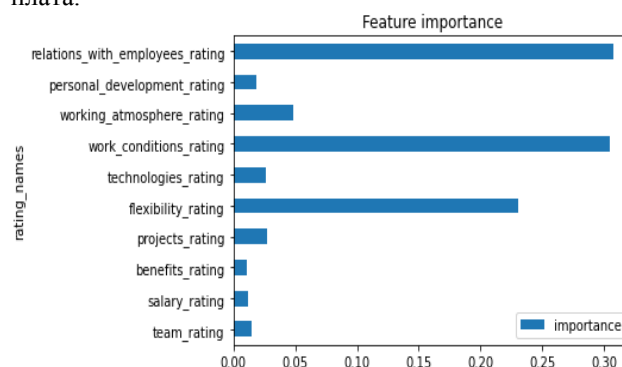
За овај задатак је искоришћен подскуп података који се тиче коментара на сам процес селекције у компанијама. Сваки коментар садржи оцену интервјуа коју је дао кандидат, као и текстуалне коментаре о HR и техничком делу интервјуа одвојено. Идеја задатка је установљење који од два дела интервјуа има већи утицај на свеукупан утисак о процесу селекције. Ради реализације задатка рачуната је оцена сентимента коментара за оба дела интервјуа и поређене су са коначном оценом целокупног процеса селекције. За ове потребе коришћен је трансформер модел из *HuggingFace* библиотеке [5]. Конкретније, коришћен је модел "*BERT Multilingual Uncased Sentiment*" из организације "*nlptown*". Након провлачења коментара кроз BERT модел, као предикција је узета оцена од један до пет највеће вероватноће. Разлика је рачуната помоћу *MSE* метрике. Коментар о делу интервјуа који би мање одступао од циљне лабеле, би указивао на потенцијално већи утицај на свеукупан утисак о процесу селекције.

4. РЕЗУЛТАТИ И ДИСКУСИЈА

У овом поглављу су приказани добијени резултати за сва три експеримента и предочене потенцијалне мане и предложена побољшања.

4.1. Одређивање важности аспеката компаније

Након претраге хиперпараметара, најбољи модел постиже резултат од **0.0022 MSE**. На слици 1 су приказани аспекти компаније и њихова важност. Види се да су три најважнија аспекта управо она која имају најјачу позитивну корелацију са оценом компаније. Резултат овог задатка потенцијално указује на приоритет запослених при вредновању свог радног места. Неки од неочекиваних резултата су места приоритета пројеката, коришћених технологија, као и плата.



Слика 1 Значај различитих аспеката компаније при предикцији оцене

Најважнији аспект које компаније треба да негују јесте управо однос према запосленима. Услови рада

су доста широк појам, и запосленима то може бити примарна асоцијација на оцену компаније свеукупно. Добијени резултати могу се додатно поредити са резултатима линеарне регресије и других модела.

4.2. Предикција процента запослених који мисле да је плата фер

Најбољи резултат од **0.1 MSE** постигнут је помоћу *RandomForestRegressor* модела са хиперпараметрима *max_depth=4* и *n_estimators=128*. Колоне подскупа података које дају најбољи резултат су „флексибилно радно време”, „рад на даљину”, „плаћени курсеви”, „добиање бонуса” и „осигурање”. Закључак је да су дати бенефити најбољи показатељи да ли запослени мисле да је плата фер. Како се најбоље показао модел насумичних шума, над тиме истренираним моделом је одрађена и анализа важности улазних обележја. Као најзначајнији бенефит се показало флексибилно радно време. Након њега следи „рад на даљину”, који има сличну важност са бенефитима „плаћени курсеви и обуке”, „приватно осигурање” и „добиања бонуса”.

4.3. Анализа сентимента коментара о процесу селекције

Након проласка кроз свих 1856 коментара, добијени су резултати да мера *MSE* коментара *HR* дела износи 2.7384, док *MSE* коментара техничког дела интервјуа износи 2.4461. Из приложеног се може већ закључити да је сентимент мање-више конзистентан за оба дела интервјуа, уз незнатну предност *HR* дела као већег фактора на свеукупну оцену.

Недостатак овог приступа лежи у чињеници да не постоји пре-тренирани трансформер модел за овај задатак на српском језику, те у случају појављивања таквог модела, експеримент би требало поновити.

5. ЗАКЉУЧАК

У овом раду описана је примена разних техника машинског учења над подацима прикупљених са сајта *Joberty.rs*. Креиран је скуп података о рецензијама ИТ фирми у Србији. Екстракција знања из рецензија први пут урађена над подацима фирми у Србији, по узору на најпознатију алтернативу - *Glassdoor.com*.

У првом задатку је вршена анализа важности аспеката компанија. Проблем је моделован као регресиони, помоћу модела насумичних шума. За циљну лателу је коришћена оцена компаније, а улаз у модел су оцене аспеката компаније попут флексибилности, односа послодавца према запосленима, занимљивост пројеката, итд. Резултати указују да је запосленима најважнији аспект односа послодавца према запосленима. Уз овај аспект, најбитнији су општи услови рада и флексибилност, односно баланс приватног и пословног живота. Добијени резултати су доста слични са радовима који су анализе вршили над подацима са *Glassdoor.com* сајта. Проблем представља другачије фразирање сличних аспеката и недовољно објашњење шта тачно представља оцењивани аспект. Резултате експеримента би потенцијално требало поредити са резултатима линеарни регресије или других модела.

Други задатак моделује предикцију проценту запослених који мисле да је плата фер. Улаз у модел

су оцене бенефита које компанија нуди попут плаћених курсева, осигурања, итд. Најбољи резултат постигнут је тренирањем *RandomForestRegressor* модела од 0.10 *MSE*. На основу истренираног модела, извршена је анализа важности бенефита компаније. Флексибилно радно време је убедљиво најважнији бенефит, а за њим следе рад на даљину, добијање бонуса, плаћени курсеви и осигурање, наведени сходно важности.

У следећем задатку је спроведена анализа сентимента коментара. Наиме помоћу пре-тренираног БЕРТ модела је извршена анализа коментара о техничком делу процеса селекције, и *HR* делу процеса селекције. Добијена оцена сентимента је поређена са свеукупном оценом процеса селекције коју је кандидат доделио. Поређење је вршено помоћу *MSE* метрике. Резултати указују да не постоји значајна разлика у оценама једног дела интервјуа у односу на оцену целог процеса селекције. Непостојање трансформер модела истренираног на корпусу српског језика, за задатак анализе сентимента, представља проблем и доводи у питање валидност резултата.

Ове студије могу послужити као користан алат за компаније како би повећали задовољство запослених. Такође, креирани скуп података могао би да послужи у неким другим истраживањима која се тичу ИТ тржишта у Србији.

6. ЛИТЕРАТУРА

- [1] Landers, Richard N., Robert C. Brusso, and Elena M. Auer. "Crowdsourcing job satisfaction data: Examining the construct validity of Glassdoor. com ratings." *Personnel Assessment and Decisions* 5, no. 3 (2019): 6.
- [2] Luo, Ning, Yilu Zhou, and John Shon. "Employee satisfaction and corporate performance: Mining employee reviews on glassdoor. com." (2016).
- [3] Jung, Yeonjae, and Yongmoo Suh. "Mining the voice of employees: A text mining approach to identifying and analyzing job satisfaction factors from online employee reviews." *Decision Support Systems* 123 (2019): 113074.
- [4] High, Rob. "The era of cognitive systems: An inside look at IBM Watson and how it works." *IBM Corporation, Redbooks* 1 (2012): 16.
- [5] <https://huggingface.co/docs/transformers/index> [приступљено: 11.10.2022.]

Кратка биографија:



Митар Перовић рођен је 1998. године у Подгорици. Основне академске студије завршио је 2021. године на Факултету техничких наука у Новом Саду, на ком брани и мастер рад 2022. године из области **Електротехнике и рачунарства** контакт: perovicmitar@uns.ac.rs

RAZVOJ VS CODE EKSTENZIJE ZA VIZUELIZACIJU I MANIPULACIJU GRAPHQL ŠEMOM**DEVELOPMENT OF VS CODE EXTENSION FOR VISUALIZATION AND MANIPULATION OF GRAPHQL SCHEMA**

Tanja Vukmirović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu opisana je implementacija ekstenzija za Visual Studio Code koja treba da omogućiti dodavanje novih elemenata u GraphQL šemu, kao i vizuelizaciju već postojeće šeme

Ključne reči: *vscode ekstenzija, graphql, šema, vizuelizacija*

Abstract – *This paper describes implementation of an extension for Visual Studio Code that enables users to add new elements to their GraphQL schema, and to visualize already existing schemas*

Keywords: *vscode extension, graphql, schema, visualization*

1. UVOD

Naše današnje društvo svedok je eksplozije informacija, znanja i sve veće složenosti sistema u raznim domenima. Imati informacije „na dohvat ruke“ postalo je ključno pitanje. Da bi se obezbedilo da velike količine informacija budu lako dostupne korisnicima, one moraju biti strukturirane, a sama struktura poznata korisnicima.

Vizuelizacije struktura svojstvenih velikim količinama informacija mogu pomoći u razumevanju odnosa između informacionih elemenata i vizuelnom pretraživanju relevantnih informacija. Stoga, postoji potreba za tehnikama vizuelizacije kako bi strukture informacija u velikim repozitorijumima bile očigledne i od pomoći korisnicima u efikasnom pretraživanju i lociranju elemenata informacija relevantnih za zadatak s kojim se suočavaju [1].

Kako bi se uspostavila komunikacija između dva sistema potrebno je definisati API (Application Programming Interface). API opisuje način na koji dva servisa komuniciraju i koje podatke tom prilikom razmenjuju. GraphQL je upitni jezik sa rad sa API-jima.

Za specifikaciju podataka koji se razmenjuju prilikom komunikacije koristi se šema. U slučaju sistema gde kompleksnost raste, raste i kompleksnost šeme, te razumevanje šeme, kao i pretraga i izmena podataka postaju otežani.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Miroslav Zarić, red. prof.

2. PREGLED SLIČNIH SISTEMA

Kako bi se bolje razumeli i definisali funkcionalni zahtevi poželjno je imati uvid u već postojeća rešenja.

2.1. GraphQL

GraphQL [2] je razvojno okruženje integrisano u pretraživač. Podržava kompletnu GraphQL specifikaciju. Pruža isticanje grešaka u realnom vremenu, automatsko kompletiranje koda, a takođe čuva i istoriju pokrenutih upita.

2.2. GraphQL Playground

GraphQL Playground [3] zasnovan je na najboljim delovima GraphQL-a, uz dodatak novih funkcionalnosti i poboljšanje postojećih, kao što je bolji prikaz istorije pokrenutih upita.

2.3. GraphQL Editor

GraphQL Editor [4] se fokusira na vizuelizaciji šeme. Korisnici su u mogućnosti da definišu šemu i pregledaju blokove dok pišu ili mogu da kreiraju šemu pomoću blokova, a potom da na osnovu njih generišu kod.

2.4. GraphQL Voyager

GraphQL Voyager [5] je alat koji se koristi za vizuelizaciju šeme. Neke od osnovnih funkcionalnosti su brzo pozicioniranje na graf, kao i detaljne informacije o svakom tipu.

2.5. OpenAPI (Swagger) Editor

OpenAPI (Swagger) Editor [6] je VS Code ekstenzija koja daje podršku OpenAPI specifikaciji u JSON i YAML format. Neke od funkcionalnosti koje uključuje su pregled SwaggerUI-a, IntelliSense, navigacija koda, statička sigurnosna analiza i druge.

3. KORIŠĆENE SOFTVERSKJE TEHNOLOGIJE**3.1. Visual Studio Code, JavaScript/TypeScript, Node.js**

Visual Studio Code editor dolazi sa ugrađenom podrškom za JavaScript, TypeScript i Node.js, kao i bogatim ekosistemom ekstenzija. Za implementaciju rešenja koristio se TypeScript programski jezik koji je tipizirani superset JavaScript-a i koji se kompajlira u čist JavaScript. Node.js je platforma za lako pokretanje brzih, skalabilnih i laganih aplikacija. Može se smatrati serverskim JavaScript-om. Za vizuelizaciju šeme, korišćen je npm paket *graphqlvisualizer*. Ovaj alat omogućuje vizuelizaciju postojećih GraphQL šema. Nudi

dve opcije – vizuelizaciju šeme iz lokalnih fajlova ili vizuelizaciju šeme koja se prosleđuje u tekstualnom formatu.

3.2. Visual Studio Code ekstenzija

Visual Studio Code kreiran je sa idejom da bude proširivo radno okruženje. Od korisničkog interfejsa do funkcija za obradu koda, skoro svaki deo Visual Studio Code-a može biti prilagođen i poboljšan korišćenjem Extension API-ja. Postoje različiti tipovi ekstenzija koje se mogu kreirati. Da bi se kreirala ekstenzija potrebno je razumeti sledeća tri koncepta:

- **Aktivacioni događaji** [7], događaji na koje se ekstenzija aktivira
- **Tačke doprinosa** [8], funkcionalnosti koje se proširuju
- **VS Code API** [9], skup JavaScript API-ja koji mogu da se pozivaju iz koda ekstentije

Svaka ekstenzija treba da sadrži package.json fajl koji predstavlja manifest ekstenzije u kom su definisani aktivacioni događaji i tačke doprinosa, kao i extension.ts koji je ulazni fajl ekstenzije i koji definiše dve funkcije. Funkcija activate() se poziva kada se desi aktivacioni događaj koji je specificiran, a druga nije obavezna i poziva se kad se ekstenzija deaktivira.

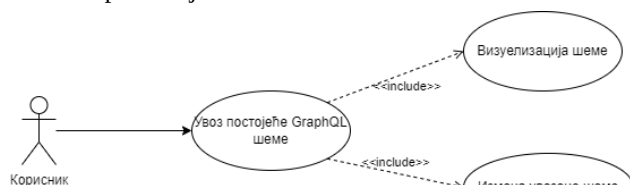
4. SPECIFIKACIJA

4.1. Specifikacija zahteva

U ovom odeljku specificirani su funkcionalni i nefunkcionalni zahtevi.

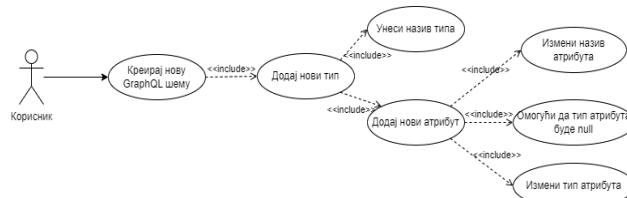
4.1.1. Specifikacija funkcionalnih zahteva

Nakon što se ekstenzija aktivira, korisnik pred sobom ima dve opcije, da uveze već postojeću GraphQL šemu ili da kreira novu šemu ispočetka. Ukoliko se korisnik odluči da uveze postojeću šemu, potrebno je da izabere fajl koji ima ekstenziju .gpl ili .graphql, kako su to jedini podržani formati. Nakon što korisnik uveze šemu, može ručno da pravi promene nad šemom. Ako se desi da korisnik svojim promenama dovede šemu u nevalidno stanje, nakon iniciranja pregleda šeme biće prikazana poruka o grešci. Ukoliko je šema validna, klikom na dugme za pregled šeme prikazaće se vizuelizovana šema. Ovaj scenario prikaza je na Slika 1.



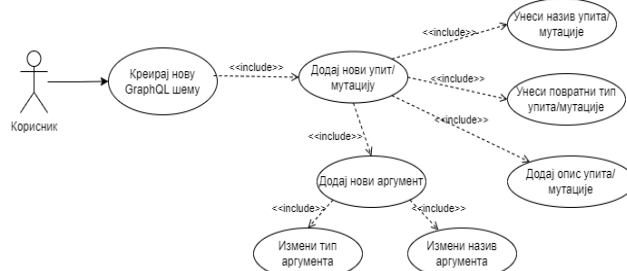
Slika 1. Dijagram slučaja korišćenja postojeće šeme

Korisnik, takođe, može da se odluči da kreira šemu ispočetka i u tom slučaju otvara se prazan tekstualni dokument. Korisnik ima opciju da doda novi tip, enumeraciju, upit ili mutaciju. Na Slika 2. prikazan je dijagram slučaja korišćenja kada korisnik odabere opciju dodavanja novog tipa. Tada korisnik može da menja naziv tog tipa, kao i da dodaje nove attribute, gde takođe može da menja naziv i tip atributa, kao i da specificira da li se dozvoljava da vrednost atributa može biti null.



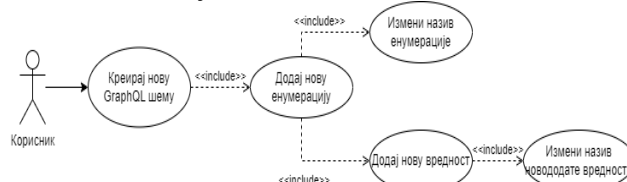
Slika 2. Dijagram slučaja korišćenja kada je odabrano dodavanje novog tipa

Korisnik može da odabere da doda upit ili mutaciju. Ove dve opcije pružaju iste mogućnosti pa su prikazane u sklopu jednog dijagrama slučaja korišćenja prikazanog na Slika 3. Nakon što korisnik izabere da doda novi upit ili mutaciju, može da promeni naziv upita ili mutacije, kao i povratni tip. Ono što korisnik još može da uradi jeste da dodaje argumente upitima i mutacijama, kao i opise.



Slika 3. Dijagram slučaja korišćenja kada je odabrano dodavanje upita ili mutacije

Preostala opcija, koja se pruža korisniku jeste i dodavanje enumeracije, prikazano na Slika 4. Nakon što korisnik doda enumeraciju, može da izmeni naziv enumeracije ili da doda novu vrednost, čiju vrednost takođe može naknadno da menja.



Slika 4. Dijagram slučaja korišćenja kada je odabrano dodavanje enumeracije

4.1.2. Specifikacija nefunkcionalnih zahteva

Upotrebljivost je mera kojom se često iskazuje koliko su korisnički interfejsi laki za korišćenje.

Reč „upotrebljivost“ se, takođe, odnosi i na metode za poboljšanje lakoće korišćenja tokom procesa dizajniranja. Upotrebljivost je definisana sa pet komponenti kvaliteta:

- **Mogućnost učenja** – koliko je korisnicima lako da ostvare osnovne zadatke kada prvi put naiđu na dizajn
- **Efikasnost** – koliko korisnici brzo mogu da izvršavaju zadatke kada nauče dizajn
- **Pamtljivost** – koliko je lako ostvariti efikasnost kada se vratite dizajnu nakon određenog vremena
- **Greške** – koliko često korisnici mogu da naprave greške
- **Zadovoljstvo** – koliko je ugodno koristiti dizajn

Postoji još bitnih atributa kvaliteta, a onaj koje se ističe je korisnost, koji se odnosi na to da li određena funkcionalnost radi ono što korisniku treba. Upotrebljivost i korisnost ne idu jedno bez drugoga. Može se desiti da postoji funkcionalnost koja je jednostavna za korišćenje, ali nije ono što je korisniku potrebno. Isto tako, nije dobro ako sistem radi ono što je korisniku potrebno, ali korisnik ne može da to efikasno iskoristi jer je korisnički interfejs prekomplikovan.

4.2. Specifikacija sistema

U ovom poglavlju opisani su model podataka i struktura rešenja.

4.2.1. Model podataka

Kako bi se podržao rad sa osnovnim konceptima GraphQL-a, definisane su klase koje odgovaraju konceptima koji su podržani u sklopu ekstenzije. Koncepti koji su podržani su tipovi, enumeracije, upiti i mutacije.

Bazični koncept GraphQL šeme je objektni tip koji se u ekstenziji modeluje *TypeNode* klasom. Može da sadrži više polja koja se modeluju klasom *PropertyNode*. Upiti su modelovani klasom *QueryNode*. Svaki upit može da ima više argumenata, koji su takođe modelovani *PropertyNode* klasom. Isti slučaj je i sa mutacijama koje se modeluju *MutationNode* klasom. Mutacijama i upitima je moguće dodavati opise, a to je modelovano poljem *description* u klasama *QueryNode* i *MutationNode*. Poslednji koncept koji je modelovan jeste enumeracija. Enumeracije su modelovane *EnumerationNode* klasom.

4.2.2. Struktura rešenja

Kako se radi o VS Code ekstenziji, struktura rešenja je jednostavna. Sastoji se iz *TypeScript* fajlova (modula) koji sadrže funkcije koje implementiraju određenu poslovnu logiku i koji su organizovani u direktorijume prema celini koju zajedno čine.

5. IMPLEMENTACIJA

5.1. Aktivacija ekstenzije

Ulazna tačka u ekstenziju jeste *extension.ts* fajl, u kome je definisana funkcija *activate()* koja se aktivira kada se desi neki od aktivacionih događaja. Da bi se aktivacioni događaj desio, potrebno ga je prvo definisati u manifestu ekstenzije u *activationEvents* polju. Zadatak funkcije *activate()* jeste da registruje komande i inicijalizuje događaje koji mogu da se dese prilikom obrađivanja komandi.

U ovoj funkciji se definišu i obrađivači događaja

- *onDidChangeActiveTextEditor* i
- *onDidChangeTextDocument*.

Pored aktivacionih događaja u manifestu ekstenzije potrebno je definisati i komande koje su dostupne iz korisničkog interfejsa. Komande se definišu u objektu *contributes.commands*. Osnovno prilikom definicije komande jeste navesti identifikator komande, pošto se on kasnije koristi za registraciju komande.

5.2. Korisnički interfejs

Korisnički interfejs sastoji se iz četiri osnovna dela, a to su traka aktivnosti, kontejner pogleda, editor i aktivnoti

editora. Traka sa aktivnostima služi za navigaciju u okviru VS Code-a.

Kako bi se dodala nova stavka u traku sa aktivnostima potrebno ju je definisati u manifestu ekstenzije u okviru objekta *contributes.viewsContainer.activitybar*. Na ovaj način se stavka aktivnosti povezuje sa kontejnerom pogleda. Kontejner pogleda je roditeljski kontejner u koji je moguće dodati poglede. Pogledi mogu biti tipa *TreeView*, *WelcomeView* i *Webview*.

WelcomeView je inicijalni pogled i definiše se u sklopu objekta *contributes.viewsWelcome*, a sadrži dve komande, za kreiranje nove šeme i uvoz već postojeće.

Kako bi se kreirao *TreeView* potrebno je u objekat *contributes.views* dodati objekat koji predstavlja listu čvorova u stablu. Naziv ovog objekta treba da odgovara identifikatoru nekog od objekata iz liste *contributes.viewsContainer.activitybar*. Kako bi se prikazali podaci u sklopu ovih pogleda, za svaki pogled potrebno je definisati klasu koja će implementirati klasu *TreeDataProvider* i tom prilikom implementirati sve metode: *getChildren* i *getTreeView*. Klasa *getChildren* treba da vrati listu elemenata u slučaju korenskog čvora ili listu pripadajućih elemenata ukoliko se prosledi čvor. Tako, na primer, ukoliko smo pozicionirani u pogledu *Types*, trebali bi dobiti listu svih tipova ili listu svih atributa ako prosledimo određeni tip. Metoda *getTreeItem* treba da vrati UI reprezentaciju čvora, odnosno objekat klase koja nasleđuje *TreeItem*.

5.3. Manipulacija šemom

Manipulacija šemom omogućava izmene tipova, enumeracija, upita, mutacija i drugih elemenata. Kako bi se ovo uspešno ostvarilo, potrebno je komande koje su definisane u manifestu ekstenzije definisati i nad objektom *contributes.menus* manifesta ekstenzije. Ukoliko se radi o komandama koje se izvršavaju nad pogledom koji je definisan u objektu *contributes.views*, komandu je potrebno definisati u objektu *contributes.menus.view/title* i potrebno je dodati atribut koji govori kada komanda treba da se prikaže. Ukoliko je komandu potrebno dodati nekom od elemenata koji pripadaju definisanim pogledima, onda se to radi u objektu *contributes.menus.view/item/context*, gde je takođe potrebno navesti uslov pod kojim se prikazuje komanda, a koji treba proveriti da li se radi o željenom pogledu i da li kontekstna vrednost odgovara željenoj.

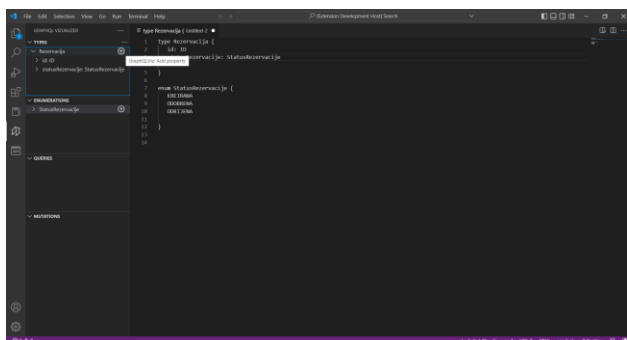
Nakon što se komande dodaju u manifest ekstenzije, za svaku komandu potrebno je obezbediti da postoji odgovarajući obrađivač te komande.

5.4. Vizuelizacija šeme

Za prikaz vizuelizacije šeme koristi se klasa *SchemaPreviewPanel* čija odgovornost je da kreira novi *webview* panel za prikaz šeme ili da ažurira postojeći, kao i da prilagodi HTML sadržaj koji je potrebno prikazati u panelu. Kako bi se dobio sadržaj koji je potrebno prikazati u panelu koristi se funkcija *Visualize*, *graphqlvisualizer* npm paketa. Ova funkcija kao parametar prima šemu u tekstualnom formatu, a kao rezultat vraća objekat koji sadrži sam html sadržaj koji je potrebno prikazati, stilove koje je potrebno primeniti na taj sadržaj kao i skripte koje omogućavaju manipulaciju sadržajem.

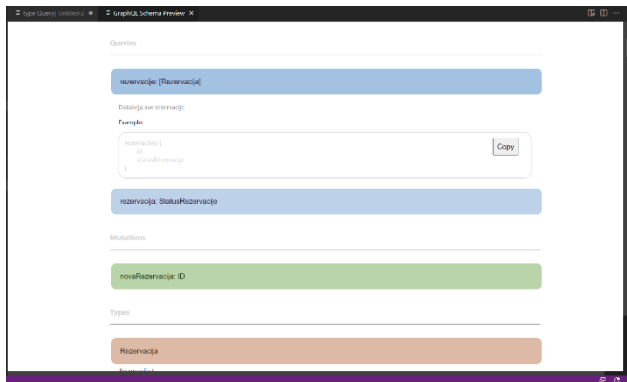
6. DEMONSTRACIJA

U ovom poglavlju biće demonstriran scenario prikazan na Slika 2. Nakon što je korisnik odabrao da kreira šemu ispočetka, može da odabere opciju da doda novi tip, enumeraciju, upit ili mutaciju. U nastavku fokus će biti na dodavanju novog tipa. Da bi korisnik dodao novi tip potrebno je da klikom na „tri tačkice“ u sklopu pogleda *Types* odabere opciju dodavanja novog tipa. Tada se dodaje novi tip sa predefinisanim nazivom, koji je moguće izmeniti kao na slici ispod. Nakon što doda novi tip korisnik može da doda i nove attribute, klikom na plus pored naziva odgovarajućeg tipa kome želi da doda novi atribut. Pored ovoga, korisnik dodatom atributu može da postavi da tip bude ili ne bude null vrednost, što čini klikom na olovkicu pored naziva atributa.



Slika 5. Izgled korisničkog interfejsa nakon što se doda tip i enumeracija

Kada korisnik odluči da želi da pregleda šemu, može to učiniti klikom na komandu u editorskom meniju. Tada se korisniku prikazuje panel prikazan na Slika 6.



Slika 6. Prikaz vizuelizovane šeme

7. ZAKLJUČAK

Predmet ovog rada bilo je istraživanje postojećih rešenja kada je u pitanju vizuelizacija GraphQL šema i implementacija novog rešenja koje će omogućiti kako vizuelizaciju šeme, tako i manipulaciju istom u cilju lakšeg rukovanja šemom kada sistemi koji se modeluju naglo porastu, a rukovanje podacima postane otežano.

Ono što ovo rešenje čini različitim od ostalih rešenja koja se bave vizuelizacijom GraphQL šema jeste što se šema vizuelizuje na način sličan SwaggerUI-u koji je poznat kao jedan od popularnijih alata za vizuelizaciju i dokumentaciju OpenAPI specifikacije, čineći je jednostavnom za čitanje i intuitivnom.

Kako je implementacija tek započeta, mesta za napredak ima mnogo. Korisno bi bilo omogućiti povezivanje sa serverom i na taj način omogućiti izvršavanje testnih upita i mutacija i prikaz vrednosti koje se dobijaju kao rezultat izvršavanja API poziva.

4. LITERATURA

[1] Keller, T., Tergan, SO. 2005. Visualizing Knowledge and Information: An Introduction.

[2] GraphQL <https://github.com/graphql/graphql>

[3] GraphQL Playground <https://github.com/graphql/graphql-playground>

[4] GraphQL Editor <https://github.com/graphql-editor/graphql-editor>

[5] GraphQL Voyager <https://github.com/IvanGoncharov/graphql-voyager>

[6] OpenApi (Swagger) Editor <https://marketplace.visualstudio.com/items?itemName=42Crunch.vscode-openapi>

[7] Aktivacioni događaji <https://code.visualstudio.com/api/references/activation-events>

[8] Tačke doprinosa <https://code.visualstudio.com/api/references/contribution-points>

[9] VS Code API <https://code.visualstudio.com/api/references/vscode-api>

Kratka biografija:



Tanja Vukmirović rođena je 07.09.1997. u Novom Sadu. Srednju školu završila je u Novom Sadu, nakon čega je upisala Fakultet tehničkih nauka u Novom Sadu, studijski program Primenjeno softversko inženjerstvo. Diplomirala je 2020. godine, a iste godine je upisala i master studije. Položila je sve ispite predviđene planom i programom.

HTML VIZUELIZACIJA GRAPHQL SCHEMA-E U VS CODE EDITORU**HTML VISUALIZATION OF GRAPHQL SCHEMA IN VS CODE EDITOR**Nikolina Šarenac, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – Kroz ovaj rad se razmatra mogućnost vizuelnog dokumentovanja interfejsa web servisa definisanog pomoću GraphQL-a. Implementirano je i predstavljeno jedno takvo rešenje, počevši od njegove strukture, načina implementacije, do demonstracije implementiranih funkcionalnosti.

Ključne reči: GraphQL šema, SDL, vizuelizacija, API dokumentacija

Abstract – This paper contemplates the possibilities of implementing visual documentation of web service APIs that are defined using GraphQL. Alongside, one such solution is implemented and will be presented here, starting with its structure, implementation details, and up to the demonstration of containing functionalities.

Keywords: GraphQL schema, SDL, visualization, API documentation

1. UVOD

U poslovnom svetu, svakodnevno se generišu velike količine podataka u raznim sferama delovanja, što podstiče sve veću zainteresovanost za analizu efikasnosti načina upotrebe i upravljanja tim podacima. Kako su podaci često velike kompleksnosti i međusobno povezani na različite načine, potrebno je da njihova razmena između različitih sistema bude što bolje definisana i optimizovana. Optimalan prenos podataka obuhvata razmenu najužeg potrebnog skupa podataka, obavljenu sa što manje komunikacije između različitih entiteta.

Komunikacija između različitih sistema definisana je specifikacijom aplikativnih interfejsa, odnosno API-ja (eng. *Application Programming Interface*), a među specifikacijama, imajući u vidu pomenute prednosti, trenutno najveći rast u popularnosti ima GraphQL, koji je detaljnije opisan u odeljku 1.1.

Pored specifikacije API-ja veoma je bitan i način na koji je dokumentovan. Na osnovu dokumentacije korisnici, bilo da su klijenti koji koriste funkcionalnosti nekog servisa ili programeri koji takav servis implementiraju, treba da budu u potpunosti upućeni u način na koji se interfejs može koristiti. Velika kompleksnost se često može uprostiti upotrebom vizuelnih prikaza, a više reči o mogućnosti konkretne primene vizuelizacije za dokumentovanje GraphQL interfejsa biće u odeljku 1.2.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Miroslav Zarić, red. prof.

1.1 GraphQL

GraphQL opisuje strukture upita koji neki servis pruža korisnicima, kao i tipove podataka koji se koriste. Nastao je kao alternativa REST (eng. *Representational State Transfer*) web servisima, a jedna od najvećih prednosti je fleksibilnost u pogledu izbora konkretnih polja povratnih entiteta, gde u okviru upita korisnik dobija tačno one podatke koje traži, ni manje ni više od toga. Fleksibilnost se ogleda i u nezavisnosti od platforme kao i programskog jezika za implementaciju servisa.

GraphQL specifikacija definiše i jezik za opis šeme (eng. *Schema Definition Language - SDL*) koja predstavlja datoteku u kojoj su opisani svi postojeći tipovi, kao i upiti koji su na raspolaganju klijentu (koji se koriste za dobavljanje podataka) i mutacije (koje se koriste za modifikaciju podataka npr.: kreiranje, izmena i brisanje). GraphQL šema je pisana jezikom za definiciju šeme i skladištena u datoteci sa .graphql ekstenzijom. Na slici 1. je prikazan izgled jednostavne šeme sa definisanim tipovima *User* i *Address* kao i primerima upita i mutacija koji te tipove koriste.

```
type User {
  id: ID!
  firstName: String!
  lastName: String!
  address: Address!
}

type Address {
  city: String!
  postalCode: Int!
}

type Query {
  user(id: ID!): User!
  userOnAddress(address: Address!): [User!]
}

type Mutation {
  createAddress(city: String!, postalCode: Int!): Address!
  deleteUser(id: ID!): Boolean!
}
```

Slika 1. Primer GraphQL šeme

Ovako definisana šema predstavlja i sveobuhvatnu dokumentaciju interfejsa, ona u potpunosti opisuje kompleksne tipove koji se koriste, kao i upite nad podacima koji su korisniku dostupni. Takođe podržava i dodatak opisa za svaki podržani upit. Neki od nedostataka upotrebe šeme kao dokumentacije mogu biti nepreglednost u slučaju kompleksnog interfejsa kao i slabija čitljivost velike količine upita i tipova.

1.2 Vizuelizacija dokumentacije GraphQL servisa

Vizuelizacija kao metoda za pojednostavljenje kompleksnih prikaza se dugo koristi za različite tehnologije. Različiti vizuelni prikazi mogu pružiti više pogleda na iste podatke, pa se stoga razlikuju po svojoj svrsi.

Jezici bazirani na *json* notaciji, kao što je i prethodno pomenuti *GraphQL* jezik za definisanje šeme, prednjače po pitanju čitljivosti u odnosu na *xml* bazirane notacije.

Međutim, to ne znači da bi implementacija neke vrste vizuelne reprezentacije za takve dokumente ne bi imala ne prednosti. Iako čitljiviji, ovakvi dokumenti zbog količine podataka lako mogu postati nepregledni.

Upravo u takvim situacijama dolazi do izražaja način prezentovanja podataka korisniku. Grafička predstava ovakvih jezika se uglavnom svodi na upotrebu grafova [1]. Vizuelna reprezentacija po kojoj bi korisnik imao jednostavniji grubi pregled sadržaja, kao i prikaz detalja po potrebi, mogla bi da ima značajan uticaj na efikasnost u radu. Posebno za rad korisnika koji nisu do detalja upoznati sa sintaksama sličnih jezika.

2. PREGLED SLIČNIH SISTEMA

Često je upravo sama definisana šema sve što je potrebno korisniku za uvid u način funkcionisanja servisa, što može biti jedan od razloga za postojanje malog broja ovakvih rešenja. Prilikom istraživanja ove teme pronađeno je nekoliko onih koja pružaju srodne mogućnosti i koja će biti opisana u narednim odeljcima. Uz to će u odeljku 2.4. posebno biti istaknuto rešenje koje je steklo veliku popularnost za dokumentovanje REST servisa i koje je bilo glavna inspiracija za ovaj rad.

2.2. GraphiQL

GraphiQL je interaktivno integrisano razvojno okruženje (eng. *IDE - Integrated Development Environment*) za *GraphQL* koje se pokreće u internet pretraživaču.

Prikaz je podeljen na četiri kolone, gde je dokumentacija API-a prikazana u poslednjoj koloni, sa funkcionalnošću pretraživanja, gde se za odabrane tipove mogu izlistati sva njihova polja [2].

Jedna od prednosti *GraphiQL*-a jeste puna podrška za *GraphQL SDL* kao i ispisivanje predloga u vidu dostupnih polja pri pisanju upita.

Mana ovakvog prikaza dokumentacije je slabija preglednost – kolona u kojoj se nalazi dokumentacija može se uvećati samo na jednu trećinu prozora pretraživača, uz to su sve informacije ispisane u jednom odeljku, što otežava pregled.

2.2. GraphQL playground

GraphQL playground takođe predstavlja jedno integrisano razvojno okruženje koje je izgrađeno na osnovama rešenja opisanog u prethodnom odeljku. Vizuelni prikaz čine tri kolone, od kojih poslednja prikazuje dokumentaciju [3].

Prikaz dokumentacije je znatno poboljšan u odnosu na *GraphiQL*, na raspolaganju su dve kartice: za prikaz šeme i za interaktivno istraživanje upita kao i tipova koji se koriste.

GraphQL playground svoj razvojni put neće nastavljati samostalno jer je u planu njegovo integrisanje sa ranije pomenutim *GraphiQL* razvojnim okruženjem.

2.3. GraphQL Editor

Za razliku od prethodno pomenutih rešenja, *GraphQL Editor* je najviše fokusiran na vizuelizaciju šeme. Moguće je istovremeno pisati šemu i posmatrati njenu vizuelizaciju u vidu povezanih blokova koji grafički predstavljaju elemente šeme. Podržano je i kreiranje šeme u suprotnom smeru, gde se dodavanjem i povezivanjem grafičkih elemenata generiše odgovarajući kod.

Jedan od nedostataka se uočava kod kompleksnijih šema, mapiranje elemenata iz tekstualnog dela na grafički prikaz se radi ručno, i za svaki element neophodno je tražiti njegovu reprezentaciju i obrnuto.

2.3. Swagger (OpenAPI)

Swagger predstavlja kolekciju alata za pomoć pri dizajnu, implementaciji, dokumentovanju i upotrebi REST API-a. Od posebnog značaja za temu ovog rada je *Swagger UI* koji pruža interaktivnu vizuelizaciju resursa web servisa bez implementacionih pojedinosti [5]. Jednostavan vizuelni prikaz čini ga veoma korisnim za različite tipove korisnika.

Operacije koje se mogu pozivati vizuelno su prikazane u vidu kartica, grupisanih prema entitetu na koje se odnose, a na kojima su ispisane osnovne informacije o pozivu. Karakteristična je i upotreba boja za određene vrste operacija što ih vizuelno razdvaja i olakšava pronalazak željene operacije.

3. TEHNOLOGIJE I ALATI

Za razvoj ovog rešenja korišten je *Visual Studio Code* editor, čije jezgro se zasniva na *JavaScript* okruženju, odnosno pokreće ga *Node.js* server.

3.1. TypeScript

Za implementaciju je korišten *TypeScript* programski jezik, strog nadskup *JavaScript*-a, koji dodaje jeziku opcionu statičku tipizaciju i objektnu orijentisanost. Dizajniran je za razvoj velikih aplikacija i prevodi se pre izvršavanja u *JavaScript*.

3.2. Node.js

Node.js je višepatformsko *JavaScript* radno okruženje otvorenog koda za izvršavanje *JavaScript*-a na serverskoj strani. Iskorišćen je za pokretanje implementiranog programa, kao i za njegovo objavljivanje u vidu NPM (eng. *Node Package Manager*) paketa.

3.3. GraphQL Tools

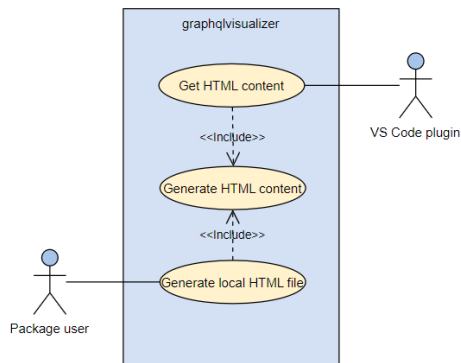
Alat koji predstavlja skup NPM paketa i omogućava lakše učitavanje šema iz raznih izvora, kao i izgradnju servisa definisanih šemom [6].

4. SPECIFIKACIJA

Analizom sličnih sistema, njihovih prednosti i nedostataka kao i uočenih potreba korisnika, definisana je specifikacija ovog rešenja čiji će funkcionalni i nefunkcionalni zahtevi biti obrađeni u nastavku.

4.1. Funkcionalni zahtevi

Osnovna funkcionalnost potrebna za realizovanje programa za vizuelizaciju `graphql` šema prikazana je na dijagramu slučaja korištenja na slici 3.



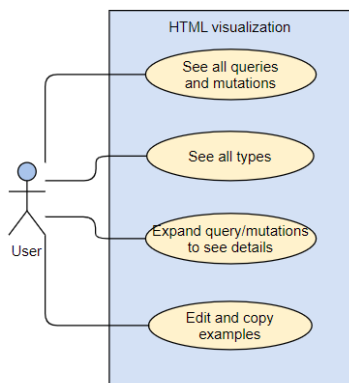
Slika 3. Dijagram slučaja korištenja `graphqlvisualizer`-a

Rešenje treba da pruži dve osnovne funkcionalnosti, prva se odnosi na običnog korisnika, dok je druga implementirana specifično za ekstenziju `VS Code` editora.

Funkcionalnost koja se odnosi na običnog korisnika treba da omogući da se po potrebi može izvršiti generisanje lokalnog fajla koji sadrži vizuelni reprezentaciju šeme. Stoga, preduslov je da korisnik poseduje definisanu šemu napisanu prema gramatici jezika za definisanje šema. Šema se prosleđuje programu na obradu na dva načina, kao jedan string parametar, ili prosleđivanjem putanje do šeme sačuvane lokalno u okviru direktorijuma projekta koji koristi ovu funkcionalnost. Obrada šeme rezultuje generisanjem `HTML` fajla koji korisnik zatim može koristiti za pregled interfejsa.

Druga funkcionalnost, koja se tiče konkretne ekstenzije `VS Code` editor-a, odnosi se na generisanje delova sadržaja `HTML` stranice, koji se zatim integrišu u samoj ekstenziji. Preduslov za izvršavanje ove funkcionalnosti jeste prosleđivanje šeme u vidu stringa. Izvršavanje rezultuje generisanjem stila, zatim tela tj. komponenti samog prikaza, kao i skriptnog dela stranice koji sadrži `JavaScript` kod potreban za interaktivnost prikaza.

Interakcije koje postoje između korisnika i generisanog vizuelnog prikaza predstavljene su na dijagramu slučaja korištenja na slici 4.



Slika 4. Dijagram slučaja korištenja vizuelnog prikaza

Korisniku treba izlistati sve postojeće tipove, kao i upite i mutacije. Za svaki od njih, proširenjem prikaza, treba da se prikažu detalji, opis ili primeri. Za upite i mutacije za koje je generisan početni primer, treba da se omogući

njihova izmena kao i lako i jednostavno kopiranje sadržaja.

4.2. Nefunkcionalni zahtevi

Pored osnovnih funkcionalnosti definisanih specifikacijom, potrebno je zadovoljiti i nefunkcionalne kriterijume.

Potrebno je sažeto predstaviti sadržaj cele šeme tako da ga korisnik može lako sagledati. Zatim treba da se omogući što lakše snalaženje među elementima, što njihovim raspoređivanjem i grupisanjem, što upotrebom boja za označavanje različitih tipova.

Uz to je neophodno obezbediti i interaktivnost prikaza, kroz proširenje detalja elemenata i njihovo sakrivanje ili uređivanje generisanih primera.

5. IMPLEMENTACIJA I DEMONSTRACIJA

U ovom poglavlju će biti ukratko opisana implementacija sistema specificiranog u prethodnom poglavlju.

Program je osmišljen kao NPM paket i objavljen je pod nazivom `graphqlvisualizer`. Kao takav može biti instaliran i korišten iz CLI-a (eng. *Command Line Interface*), gde je već instaliran `node`. Instalacija se vrši jednostavnim pozivom `> npm i graphqlvisualizer`.

Rešenje je implementirano kao `Node.js` projekat u kombinaciji sa `TypeScript` programskim jezikom.

Rešenje se sastoji od pet komponenti: generator tela prikaza (eng. *body*), generator stila, generator skripti, generator `HTML` stranice i *Visualizer* komponenta koja izvozi funkcionalnosti dostupne krajnjem korisniku. Konkretno komponente će biti objašnjene u nastavku.

5.1. Visualizer komponenta

Ova komponenta korisnicima daje na raspolaganje tri metode, a to su: `GenerateHtmlFromString`, `GenerateHtmlFromLocalFile` i `Visualize`.

Od navedenih funkcionalnosti plugin implementiran za `VS Code` editor koristi metodu `Visualize` tako što joj prosleđuje šemu u vidu stringa. Preostale dve funkcionalnosti namenjene su korisnicima koji ne koriste ekstenziju editora, nego žele lokalno generisan vizuelni prikaz. `GenerateHtmlFromString` metodi se prosleđuje šema kao string parametar, dok postoji mogućnost učitavanja fajla šeme sačuvane lokalno u okviru projekta, i to upotrebom metode `GenerateHtmlFromLocalFile`.

Pre prosleđivanja šeme na obradu, upotrebom *GraphQL Tools* alata, radi se učitavanje šeme iz konkretnog izvora, zavisno od metode koja se koristi. Transformisana šema se dalje prosleđuje generatoru `HTML` sadržaja.

5.2. Generator tela sadržaja, stila i skripti

Generator `HTML` stranice kombinuje funkcionalnosti ovih komponenti sačinjavajući funkcionalnu stranicu.

Generator stila gradi sadržaj koju definiše izgled stranice, statički je i ne zavisi od prosledene šeme.

Generator sadržaja izgrađuje telo stranice na osnovu prosledene šeme. Prolaskom kroz njen sastav obrađuju se sadržani elementi i generišu prikazi. Za upite i mutacije ispisuje se njihov potpis. Proširenjem prikaza ispisuje se opis pročitanih iz šeme, kao i primer poziva. Za korisnički definisane tipove i enumeracije generišu se elementi slični prethodno opisanim, s tim da se proširenjem prikaza ispisuje struktura tipa, njegova polja i tipovi polja, kao i informacija da li su polja obavezna.

Generator skripti omogućava dinamičko ponašanje stranice, i kroz to interakciju korisnika sa prikazom. Izgenerisan je *JavaScript* kod koji se izvršava u pretraživaču. Konkretno se odnosi na mogućnosti interakcije sa karticama koje predstavljaju tipove, proširenje i skrivanje detalja, podrška kopiranja primera klikom na dugme.

Program implementiran na način objašnjen u prethodnom poglavlju rezultuje prikazom na slici 5.

The screenshot shows a GraphQL IDE interface with three main sections: Queries, Mutations, and Types. The Queries section contains two items: 'user: User' and 'usersOnAddress: [User]'. The Mutations section contains two items: 'createAddress: Address' and 'deleteUser: Boolean'. The Types section contains two items: 'User' and 'Address'.

Slika 5. Primer generisanog prikaza

Nakon učitavanja generisanog prikaza na prvi pogled uočljiva je podela sadržaja na upite, mutacije i korisničke tipove. Na svakoj kartici se nalazi ili naziv upita i povratni tip ili naziv korisnički definisanog tipa u zavisnosti od toga na šta se kartica odnosi. Oni sačinjavaju osnovu strukture šeme. Za željenu karticu, klikom, otvara se širi prikaz detalja, kao što je prikazano na slici 6.

The screenshot shows a detailed view of a query in a GraphQL IDE. The query is 'usersOnAddress(address: Address)'. Below the query, there is a description: 'Returns a list of users that have the specified address'. An 'Example:' section shows the JSON result of the query, which is a list of users with their details. A 'Copy' button is visible next to the JSON.

Slika 6. Prikaz detalja upita

Za odabrani upit se ispisuje opis, ukoliko je definisan, a ispod njega primer poziva. Polje koje sadrži primer korisnik može uređivati i transformisati u oblik koji se zatim može izvršiti na serveru koji podržava datu šemu. Takođe je neophodno parametre date u obliku 'naziv: tip'

izmeniti tako da se prosleđuje odgovarajuća vrednost. Obavezni parametri su naglašeni podebljanim slovima i ne bi smeli biti izostavljeni.

6. ZAKLJUČAK

Razmatranjem potreba korisnika koji često ulažu vreme u razumevanje i analizu *GraphQL* šema, kao i postojećih sistema koji im taj posao olakšavaju, razvijen je *graphqlvisualizer* koji predstavlja spoj dobrih strana postojećih rešenja i omogućava istraživanje API-a na jednostavan i intuitivan način. Rešenje koje je u te svrhe steklo najveću popularnost kada su u pitanju REST servisi jeste *Swagger UI* koji je imao veliki uticaj na konačni izgled ove vizuelizacije.

Fleksibilnost ovog rešenja se ogleda u dva načina upotrebe, kroz instalirani plugin za *VS Code* editor, kao i direktnom upotrebom NPM paketa gde je omogućeno učitavanje šeme iz stringa, ukoliko je pisana u samom kodu, ili iz posebnog fajla. Funkcionalnosti generisanog prikaza su svedene na jednostavan prikaz sadržaja šeme, što određuje i njegovu osnovnu namenu, a to je upoznavanje API-a na što lakši način.

Ovakva osnova pruža mnogo mogućnosti i za dalji razvoj i proširenja. Koristan dodatak rešenju bila bi i implementirana pretraga kao i filtriranje sadržaja što bi bilo izvodljivo generisanjem dodatnih skripti koje bi se izvršavale nad stranicom.

Nešto drugačiji pristup proširenju bi mogla biti i implementacija funkcionalnog klijenta, gde bi bilo moguće sa iste stranice poslati kreirani upit i prikazati njegov odgovor.

7. LITERATURA

- [1] Frisendal, T. "Visual Design of GraphQL Data", Apress, Berkeley, CA, 2018.
- [2] <https://www.gatsbyjs.com/docs/how-to/querying-data/running-queries-with-graphql/> (pristupljeno u oktobru 2022.)
- [3] <https://github.com/graphql/graphql-playground> (pristupljeno u oktobru 2022.)
- [4] <https://graphqleditor.com/features/visual/> (pristupljeno u oktobru 2022.)
- [5] De, B. "API Documentation. In: API Management", Apress, Berkeley, CA, 2017.
- [6] <https://www.theguild.dev/graphql/tools/docs/introduction#/> (pristupljeno u oktobru 2022.)

Kratka biografija:



Nikolina Šarenac rođena je u Zvorniku 1997. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika odbranila je 2022.god.

kontakt: nina.sarenac@hotmail.com

OBRADA I VIZUELIZACIJA PODATAKA UZ OSLOMAC NA AWS

DATA PROCESSING AND VISUALIZATION USING AWS SERVICES

Nikolina Stamenić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – U ovom radu opisana je implementacija i specifikacija arhitekture sistema za obradu i vizuelizaciju podataka koji se oslanja na AWS servise.

Ključne reči: *Obrada podataka, AWS, StepFunctions, DynamoDB, S3, API Gateway, Lambda*

Abstract – *This paper describes the implementation and specification of the architecture of the data processing and visualization system that relies on AWS services.*

Keywords: *Data processing, AWS, StepFunctions, DynamoDB, S3, API Gateway, Lambda*

1. UVOD

Upravljanje podacima podrazumeva proces praćenja toka podataka od trenutka kreiranja do trenutka kada oni pronalaze svoju upotrebu. Često je potrebna dodatna obrada i izmena podataka pre nego što se dovedu u stanje potrebno za dalju upotrebu.

Obrada podataka predstavlja težak i vremenski zahtevan zadatak ukoliko se radi ručno. Tehnološki razvoj je doveo do pojave alata koji poseduju mogućnost lake i znatno brže obrade podataka. Pomenuti alati zahtevaju da korisnik poseduje određeno znanje o njihovoj upotrebi. Takođe, potrebno je da korisnik bude upućen u to na koji način podaci treba da budu obrađeni i koje izmene je potrebno napraviti.

Sistem koji se opisuje u ovom radu omogućava laku obradu podataka, gde je potreba za korisnikovo angažovanje svedena na minimum. Ideja je da korisnik obezbedi ulazne podatke, a da se sva dalja obrada obavi automatski i da korisnik kao odgovor dobije gotov rezultat sa primenjenim izmenama.

Sistem je razvijen za obradu podataka prikupljenih sa „Portala otvorenih podataka“ [1]. Radi se o podacima o zagađenjima vazduha u Republici Srbiji. Vrš se obrada ovih podataka tako da krajnji rezultat obezbeđuje informacije o zagađenjima po regijama u državi, koje delatnosti ispuštaju najviše zagađujućih materija u atmosferu, kao i koja postrojenja fabrika su najveći zagađivači.

2. KORIŠĆENE SOFTVERKE TEHNOLOGIJE

Za razumevanje rada potrebno je poznavanje osnova o AWS i njegovim servisima korišćenim za izradu rešenja.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Miroslav Zarić, vanr. prof.

2.1. AWS

AWS [2] je najsveobuhvatnija i najprihvaćenija platforma za računarstvo u oblaku (engl. *cloud computing*) na svetu koja korisnicima obezbeđuje preko 200 različitih servisa dostupnih u svakom trenutku. Dostupni su servisi IaaS tipa, koji obezbeđuju samo infrastrukturu, servisi tipa PaaS koji nude platformu, pa pored hardverskog dela obezbeđuju i konfigurisan paket softverskih podsistema i servisi SaaS tipa koji predstavljaju već implementirana softverska rešenja spremna za korišćenje.

2.2. AWS StepFunctions

AWS *StepFunctions* [3] je servis za orkestraciju procesa koji omogućava korisnicima kreiranje i upravljanje aplikacijama čiji je radni tok sačinjen od više koraka. Pri svakom koraku radnog procesa AWS vodi računa o ulaznim i izlaznim parametrima, o rukovanju greškama koje mogu da se dese, kao i o ponavljanjima izvršavanja koraka ukoliko je to potrebno.

StepFunctions su izgrađene po principu konačnog automata - mašine prelaza stanja (engl. *state machine*). Rade na principu podele posla u manje zadatke gde jedno stanje predstavlja jedan zadatak. Izvršavanje zadataka može da se obavlja na preko 200 AWS servisa. Svako stanje ima indikator koje je sledeće stanje u koje se prelazi nakon njegovog izvršavanja, pa se tako vrši orkestracija procesa i kontrola redosleda izvršavanja zadataka.

2.3. AWS DynamoDB

DynamoDB [4] je nerelaciona baza podataka zasnovana na ključ-vrednost i dokument strukturama podataka. Omogućava fleksibilnost šema podataka, pa svaki red u tabeli može da sadrži drugačiji broj i vrstu kolona. Podaci iz svake tabele su smešteni u celine koje se nazivaju particije, a jedna tabela u zavisnosti od količine podataka može da poseduje jednu ili više particija.

2.4. AWS S3

Amazon Simple Storage Service [5], skraćeno S3, je servis koji služi za virtuelno skladištenje objekata. S3 čuva objekte u kontejnerima koji se nazivaju korpe (engl. *buckets*). Na jednom AWS nalogu korisnik može imati jednu ili više „korpi“, a za svaku od njih pojedinačno mogu se postaviti prava pristupa, čitanja i kreiranja novih objekata.

Objekti se čuvaju po principu ključ-vrednost, gde je ključ naziv fajla koji se čuva, a vrednost sam fajl. Dozvoljeno je postavljanje svih tipova fajlova, a maksimalna veličina fajla koja je moguća za postavljanje je 5 terabajta.

2.5. AWS Lambda

AWS Lambda [6] je računarski servis koji ima mogućnost izvršavanja koda. Lambda funkcije su samostalne aplikacije napisane u jednom od podržanih jezika u izabranom radnom okruženju.

Funkcije mogu da obavljaju bilo koju vrstu računarskih zadataka, od opsluživanja veb stranica i obrade tokova podataka, do pozivanja API-a i integracije sa drugim AWS servisima.

Upotreba lambda otklanja potrebu za održavanjem servera radi pokretanja koda. AWS u pozadini vodi računa o serverima na kojima se lambda pokreću, operativnim sistemima, o mrežnom nivou i ostatku infrastrukture, pa korisnik može da se fokusira na kreiranje koda aplikacije i ne mora da obraća pažnju na konfiguraciju i održavanje infrastrukture.

2.5. AWS API Gateway

AWS API Gateway [7] je servis koji omogućava lako kreiranje, objavljivanje, praćenje i održavanje API-a na siguran način. Ovaj servis daje mogućnost kreiranja RESTful i WebSocket API-a uz pomoć kojih dobijamo aplikacije za dvosmernu komunikaciju u realnom vremenu.

API Gateway obavlja poslove koji uključuju prihvatanje i obradu velikog broja konkurentnih zahteva i upravljanje mrežnim saobraćajem. Vodi računa o zahtevima čija obrada u datom trenutku nije moguća, održava ih u stanju čekanja i vrši njihovu obradu kada se ona omogući. *API Gateway* ima mogućnost vršenja autorizacije, kao i provere validnosti poslatog zahteva.

2.6. Serverless framework

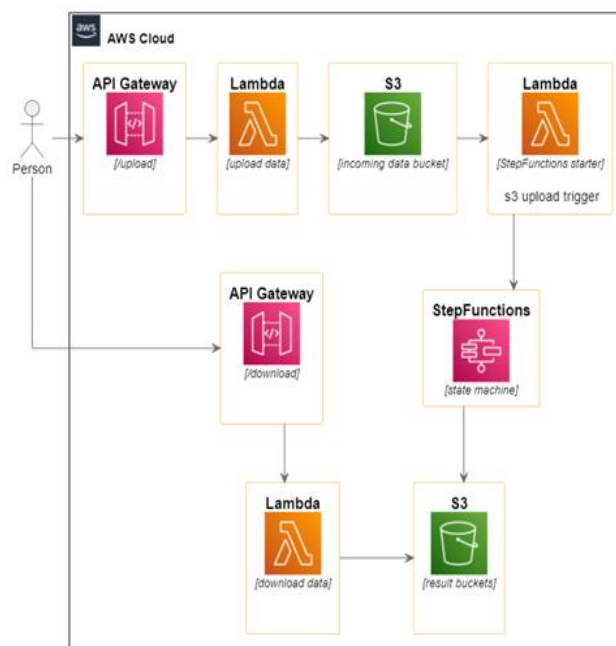
Serverless framework [8] – u prevodu okvir za razvoj aplikacija bez serverskih instanci, predstavlja rešenje za brzo kreiranje i otpremanje aplikacija koje koriste usluge direktno ponuđene od strane AWS i za čiju upotrebu nije neophodno konfigurisati serverske instance. Ovaj okvir uz upotrebu jedne komande izvršava pripremu, pakovanje i otpremanje koda. Omogućava otpremanje više AWS servisa u isto vreme, korišćenjem konfiguracionog fajla u kojem su navedeni potrebni servisi i parametri koje oni zahtevaju, kao i međusobne veze između servisa.

2.7. Python boto3

Boto3 [9] je biblioteka kreirana za *python* programski jezik. Predstavlja komplet za razvoj softvera za AWS servise. Podržava rad sa velikim brojem servisa i za svaki od njih postoji odgovarajući boto3 klijent. Ona omogućava veoma lako slanje zahteva za kreiranje, konfiguraciju, dobijanje ili izmenu podataka u sklopu AWS servisa.

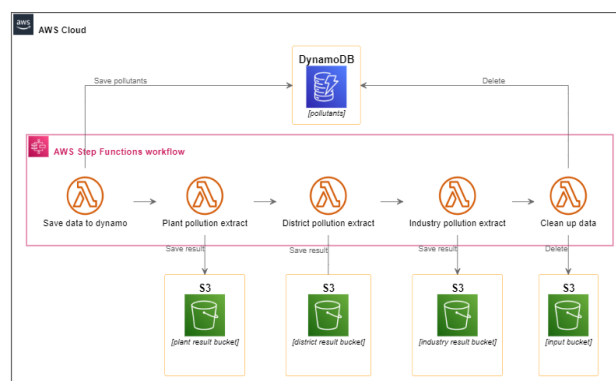
3. SPECIFIKACIJA

Na slici 1 prikazan je dijagram komponenti koje se nalaze u sistemu. Proces obrade podataka se pokreće slanjem zahteva za postavljanje ulaznog fajla. API Gateway prihvata dospeli zahtev i prosleđuje ga lambda koja obavlja čuvanje zahteva u odgovarajuću S3 korpu.



Slika 1. Dijagram komponenti

Na svako postavljanje objekta u S3 korpu podešeno je da se izvrši okidanje lambda koja ima funkciju pokretanja StepFunctions procesa čiji detalji su prikazani na slici 2.



Slika 2. StepFunctions radni proces

Prvi korak u *StepFunctions* procesu jeste prikupljanje podataka o zagađujućim materijama i njihovo čuvanje u DynamoDB bazi podataka.

Nakon što su potrebni podaci obezbeđeni i sačuvani u bazu pokreću se tri uzastopne lambda koje vrše generisanje rezultujućih fajlova. Svaka od ovih lambda vrši odgovarajuću obradu nad podacima, tako da se dobiju informacije o količinama zagađujućih materija po okrugu, delatnosti ili postrojenju fabrike. Generisane rezultate lambda čuvaju u odgovarajuće S3 korpe.

Nakon što su rezultati generisani, pokreće se poslednji korak u radnom procesu koji je čišćenje podataka. Vršiti se brisanje ulaznog fajla kao i entiteta koji su uneseni u DynamoDB bazu, jer je njihova primena bila isključivo u procesu obrade podataka.

Po završetku izvršavanja radnog procesa korisniku se otvara mogućnost pregleda generisanih rezultata i njihovo preuzimanje u formi csv fajla. Za svaki od tipova rezultata

korisnik ima mogućnost pregleda u sklopu tabele koja je prikazana na veb stranici.

4. IMPLEMENTACIJA

Sve AWS komponente korišćene za kreiranje sistema napravljene su uz pomoć *Serverless Framework*-a. Fajl je podeljen u konfiguracione celine, a na slici 3 prikazana je celina pod nazivom *provider*. U njoj se podešava tip radnog okruženja, navodi se ograničenje trajanja lambda funkcija i dozvole koje su potrebne AWS servisima.

```
provider:
  name: aws
  runtime: python3.8
  timeout: 15
  iam:
    role:
      statements:
        - Effect: 'Allow'
          Action:
            - 'states:*'
          Resource: "*"

```

Slika 3. *Provider deo u serverless.yaml fajlu*

U *functions* delu se navode definicije i konfiguracije svih lambda funkcija. Navodi se referenca na fajl u kojem se nalazi kod funkcije.

Svakoj lambda se dodeljuju potrebne dozvole za komunikaciju sa drugim AWS servisima. Primer konfiguracije jedne od lambda funkcija je prikazan na slici 4.

```
industryPollutionExtract:
  handler: functions/industry-pollution-extract.handle
  iamRoleStatements:
    - Effect: 'Allow'
      Action:
        - 's3:GetObject'
      Resource: "arn:aws:s3:::incoming-data-values/*"
    - Effect: 'Allow'
      Action:
        - 's3:PutObject'
      Resource: "arn:aws:s3:::industry-pollutants/*"
    - Effect: 'Allow'
      Action:
        - 'dynamodb:Query'
      Resource: !Sub
        "arn:aws:dynamodb:${AWS::Region}:${AWS::AccountId}:table
        /pollutions/index/input_id_gsi"

```

Slika 4. *Konfiguracija lambda funkcije u serverless.yaml fajlu*

Kao okidač funkcija koje rade otpremanje fajlova ili preuzimanje gotovih rešenja postavljen je API Gateway. Njegova konfiguracija se unosi u sklopu podešavanja lambda funkcije i prikazana je na slici 5.

```
events:
  - http:
      path: download/{type}/{year}
      cors: true
      method: get
      request:
        parameters:
          paths:
            type: true
            year: true
      response:
        headers:
          Access-Control-Allow-Origin: "*"

```

Slika 5. *Konfiguracija API Gateway lambda okidača*

Podešavanje ostalih servisa se obavlja u *resources* delu. Za *StepFunctions* je potrebno navesti dozvole za pokretanje svih lambda koje su potrebne za izvršavanje procesa.

Stanja u mašini stanja navode se željenim redosledom, gde se za svako od njih naznačava koju akciju će obavljati i koje stanje sledi nakon njegovog uspešnog izvršavanja. Način na koji su kreirana stanja u mašini stanja je prikazan na slici 6.

```
"StartAt": "SaveDataToDynamo",
"States": {
  "SaveDataToDynamo": {
    "Type": "Task",
    "Resource": "${saveDataLambdaArn}",
    "Next": "WaitTenSeconds"
  },
  "WaitTenSeconds": {
    "Type": "Wait",
    "Seconds": 10,
    "Next": "PlantPollutionExtract"
  },
  "PlantPollutionExtract": {
    "Type": "Task",
    "Resource": "${plantPollutionLambdaArn}",
    "Next": "DistrictPollutionExtract"
  },
  "DistrictPollutionExtract": {
    "Type": "Task",
    "Resource": "${districtPollutionLambdaArn}",
    "Next": "IndustryPollutionExtract"
  },
  "IndustryPollutionExtract": {
    "Type": "Task",
    "Resource": "${industryPollutionLambdaArn}",
    "Next": "CleanupData"
  },
  "CleanupData": {
    "Type": "Task",
    "Resource": "${cleanupDataLambdaArn}",
    "End": true
  }
}
```

Slika 6. *Kreiranje stanja u state mašini*

Sve lambda funkcije za obradu podataka funkcionišu na sličan način. Vršiti se pristup DynamoDB bazi i ulaznom fajlu iz kojih se učitavaju potrebni podaci. Slika 7 prikazuje učitavanje podataka iz baze uz korišćenje boto3 biblioteke. Primenjuje se potrebna logika i vrši se generisanje rezultata i njihovo čuvanje u S3 korpi.

```
response = dynamodb_client.query(
    TableName=TABLE_NAME,
    KeyConditionExpression='input_id = :input_id',
    IndexName='input_id_gsi',
    ExpressionAttributeValues={
        ':input_id': {'S': input_id}
    }
)
```

Slika 7. *Čitanje podataka iz DynamoDB baze*

Preuzimanje podataka je implementirano uz korišćenje prethodno potpisanog URL-a koji se dostavlja na *frontend* stranu aplikacije. Generisanje URL-a je predstavljeno na slici 8. Aplikacija vrši preuzimanje podataka sa pomenutog URL-a i njihov prikaz u tabeli.

```
s3_client = boto3.client('s3')
presigned_url =
s3_client.generate_presigned_url('get_object',
    Params={'Bucket': bucket, 'Key': file_name},
    ExpiresIn=60)
return
{
    "body": presigned_url,
    "statusCode": 200
}
```

Slika 8. *Generisanje ranije potpisanog URL-a*

5. ZAKLJUČAK

Motiv za implementaciju ovog rada bilo je kreiranje rešenja koje će korisniku obezbediti lak pristup željenim podacima. Ideja je da se otkloni potreba poznavanja formata ulaznih podataka od strane korisnika, kao i potreba za poznavanjem i umećem rada u alatima koji mogu da obezbede slična rešenja. Korisničke akcije su svedene na minimum, a fokus je stavljen na izvršavanje svih transformacija podataka u pozadini i dostavljanje korisniku gotovog rešenja.

Rad bi se mogao unaprediti postavljanjem veb stranice na domen koji se nalazi na *cloud* infrastrukturi. Dalje proširivanje je moguće uz definisanje novih načina obrade podataka i njihovu implementaciju i prikaz na korisničkoj aplikaciji.

6. LITERATURA

- [1] Portal otvorenih podataka
<https://data.gov.rs/sr/> (pristupljeno u oktobru 2022.)
- [2] What is AWS,
<https://aws.amazon.com/what-is-aws/> (pristupljeno u oktobru 2022.)
- [3] AWS STep functions,
<https://aws.amazon.com/step-functions/> (pristupljeno u oktobru 2022.)
- [4] AWS Dynamo DB,
<https://aws.amazon.com/dynamodb/> (pristupljeno u oktobru 2022.)
- [5] AWS Simple Storage Service,
<https://aws.amazon.com/s3/> (pristupljeno u oktobru 2022.)
- [6] AWS Lambda functions,
<https://aws.amazon.com/lambda/> (pristupljeno u oktobru 2022.)
- [7] AWS API Gateway,
<https://aws.amazon.com/api-gateway/> (pristupljeno u oktobru 2022.)
- [8] AWS Serverless framework,
<https://www.serverless.com/> (pristupljeno u oktobru 2022.)
- [9] Boto3,
<https://boto3.amazonaws.com/v1/documentation/api/latest/index.html> (pristupljeno u oktobru 2022.)

Kratka biografija:



Nikolina Stamenić rođena je u Sarajevu 1996. god. Godine 2015 upisala je Fakultet tehničkih nauka u Novom Sadu, smer Računarstvo i automatika. 2019 godine diplomirala je na osnovnim studijama na Fakultetu tehničkih nauka i iste godine upisuje master studije na smeru Računarstvo i automatika.

VIŠE-PERIODNA REKONFIGURACIJA DISTRIBUTIVNE MREŽE PRIMENOM DINAMIČKOG PROGRAMIRANJA

MULTI-PERIOD RECONFIGURATION OF THE DISTRIBUTION NETWORK USING DYNAMIC PROGRAMMING

Olivera Kojić, Neven Kovački, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu predstavljen je algoritam zasnovan na dinamičkom programiranju za određivanje optimalnog više-periodnog plana rekonfiguracije distributivne mreže. Kao kriterijumska funkcija postavljena je minimizacija ukupnih troškova gubitaka električne energije i prekidačkih akcija tokom razmatranog perioda planiranja. Predstavljena metoda testirana je na primeru realne distributivne mreže za različite troškove gubitaka električne energije i prekidačkih akcija, a rezultati su prikazani i komentarisani u radu.

Ključne reči: distributivna mreža, rekonfiguracija, metod izmene grana, dinamičko programiranje

Abstract – This paper presents the algorithm based on dynamic programming to determine the optimal multi-period reconfiguration plan for the considered distribution network. The objective is to minimize total costs of energy losses and switching manipulations. Presented method is verified on real-life distribution network for different costs of energy losses and switching manipulations and the results are presented and discussed in the paper.

Keywords: distribution network, reconfiguration, branch exchange method, dynamic programming

1. UVOD

Rekonfiguracija distributivnih mreža (DM) predstavlja proces izmene topološke strukture (konfiguracije), koji se postiže promenom uklopnog stanja prekidačke opreme. Rekonfiguracija se može posmatrati kao statički, ili kao dinamički (više-periodni) problem, u zavisnosti od dinamike promene konfiguracije DM [1]. Ako se posmatra kao statički problem tada se određuje jedna optimalna konfiguracija za jedan dati nivo potrošnje. S druge strane, ako se posmatra kao dinamički problem, onda se određuje skup konfiguracija koje je potrebno sukcesivno primeniti tokom jednog vremenskog perioda, u zavisnosti od promene potrošnje u DM.

Većina literature o rekonfiguraciji DM zanemaruje promene potrošnje u toku vremena. Pristup problemu rekonfiguracije DM kao statičkom problemu je mnogo jednostavniji sa matematičkog i računskog aspekta, dok je dinamički problem rekonfiguracije mnogo kompleksniji ali i pruža konzistentnije rezultate proračuna [2].

U ovom radu je razmatrana promena potrošnje u DM u toku jednog dana i određen je optimalan dnevni plan

rekonfiguracije. Određivanje optimalnog plana rekonfiguracije sastoji se iz tri koraka. U prvom koraku se pomoću metode izmena grana određuju optimalne konfiguracije DM za svaki sat u toku dana. U sledećem koraku vrši se proračun tokova snaga kojim se određuju gubici električne energije u svakom vremenskom periodu za sve konfiguracije DM dobijene u prethodnom koraku.

Takođe, proračunavaju se i troškovi prekidačkih akcija potrebnih za prelazak iz jedne konfiguracije u drugu.

Zatim se u trećem koraku primenjuje algoritam dinamičkog programiranja, pri čemu se koriste proračunate konfiguracije iz prethodna dva koraka. Kao rezultat dobija se optimalan plan rekonfiguracije DM za period od jednog dana kojim se minimizuju ukupni troškovi gubitaka električne energije i prekidačkih manipulacija.

2. OPŠTI OPTIMIZACIONI MODEL ZA REKONFIGURACIJU DISTRIBUTIVNE MREŽE

U ovoj glavi prikazan je optimizacioni model za rekonfiguraciju DM koji je zasnovan na mešovito-celobrojnom programiranju. Statusi grana (l) (otvorena/zatvorena) su predstavljeni binarnim (0/1) promenljivama x_l , dok je režim DM predstavljen modulima i faznim stavovima (V_i, θ_i) fazora napona u čvorovima (i) DM [1].

Kao funkcija cilja postavljena je minimizacija gubitaka aktivne snage P_{Loss} , što se zapisuje na sledeći način:

$$\min_{x_l, V_i, \theta_i} \left(P_{Loss} = \sum_{l=1}^{N_L} P_{Loss, l} = \sum_{l=1}^{N_L} (P_{ij} + P_{ji}) \right), \quad (1)$$

gde su sa P_{Loss} označeni ukupni gubici aktivne snage, sa $P_{Loss, l}$ gubici aktivne snage u grani l , N_L predstavlja ukupan broj grana u DM, dok $P_{ij}(P_{ji})$ predstavlja tok aktivne snage u grani l od čvora i (j) ka čvoru j (i).

U datom optimizacionom modelu važe sledeća ograničenja:

1. Zavisnost tokova snaga (P_{ij}, Q_{ij}) u granama DM $l=1, 2, \dots, N_L$ od promenljivih odlučivanja (x_l, V_i, θ_i)

$$P_{ij} = x_l \{ G_l (V_i)^2 - V_i V_j [G_l \cos(\theta_i - \theta_j) + B_l \sin(\theta_i - \theta_j)] \} \quad (2)$$

$$Q_{ij} = x_l \{ -B_l (V_i)^2 - V_i V_j [B_l \cos(\theta_i - \theta_j) - G_l \sin(\theta_i - \theta_j)] \} \quad (3)$$

gde su G_l i B_l konduktansa i susceptansa grane l , respektivno.

2. Bilansi aktivnih i reaktivnih snaga u čvorovima distributivne mreže $i=1, 2, \dots, N_N$:

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio docent dr. Neven Kovački.

$$P_{G,i} - P_{L,i} - \sum_{j \in \Omega_i} P_{ij} = 0, \quad (4)$$

$$Q_{G,i} - Q_{L,i} - \sum_{j \in \Omega_i} Q_{ij} = 0. \quad (5)$$

gde su sa $P_{G,i}$, $P_{L,i}$ označeni proizvodnja i potrošnja aktivne snage u čvoru i , sa $Q_{G,i}$, $Q_{L,i}$ označeni su proizvodnja i potrošnja reaktivne snage u čvoru i , a sa Ω_i skup čvorova j koji su direktno povezani sa čvorom i ,

3. Specificirani moduo i fazni stav napona u napojnom čvoru ($i=0$):

$$V_0 = V_S, \quad (6)$$

$$\theta_0 = 0, \quad (7)$$

gde je V_S specificirana vrednost modula fazora napona napojnog čvora.

4. Ograničenje modula napona u čvorovima DM $i=1,2,\dots,N_N$:

$$V_i^{\min} \leq V_i \leq V_i^{\max}, \quad (8)$$

gde su $V_i^{\min}$ i $V_i^{\max}$ minimalni i maksimalni dozvoljeni napon u čvoru i , respektivno.

5. Ograničenje tokova snaga u granama distributivne mreže $l=1,2,\dots,N_L$:

$$(P_{ij})^2 + (Q_{ij})^2 \leq (S_l^{\max})^2, \quad (9)$$

gde je $S_l^{\max}$ maksimalna dozvoljena snaga grane l ,

6. Radijalna konfiguracija razmatrane DM:

$$\sum_{i=1}^{N_i} x_i = N_N - N_{SP}, \quad (10)$$

gde je N_{SP} broj napojnih čvorova u DM.

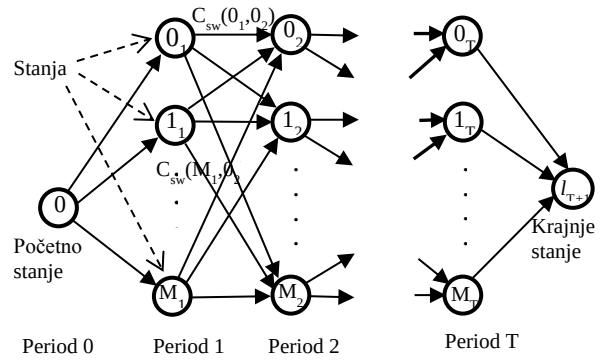
3. METODA IZMENE GRANA

Metod izmene grana (eng. *Branch Exchange*) jeste heuristički pristup koji se koristi da se na brz i jednostavan način ispitaju promene vrednosti zadate kriterijumske funkcije usled promene mesta normalno otvorenog (NO) rasklopnog uređaja u mreži.

Cilj heurističkih metoda jeste da se poznajući fiziku određenog problema u što kraćem roku dođe do kvalitetnih praktičnih, ali suboptimalnih (lokalnih) rešenja [1]. Izmena grana (IG) zapravo predstavlja izmenu mesta NO rasklopnog uređaja sa njime spregnutim normalno zatvorenim (NZ) rasklopnim uređajem. Pod spregnutim rasklopnim uređajima podrazumeva se par NO/NZ čijom se izmenom mesta održava napajanje svih čvorova u mreži i radijalnost mreže.

Velika prednost ovog metoda je da se promena mesta NO i NZ rasklopnih uređaja proračunava na bazi relacije u kojoj figurišu samo vrednosti iz baznog režima mreže. Na taj način, izbegnuta je potreba za proračunom tokova snaga (proračunom kompletnog režima) za svaku novu konfiguraciju mreže. Dakle, ispituje se vrednost kriterijumske funkcije za NO uređaj i svaki njemu spregnuti NZ rasklopnim uređaj, nakon čega se prihvata onaj slučaj gde je ostvareno poboljšanje kriterijumske funkcije.

Tek nakon tog koraka, vrši se proračun tokova snaga koji predstavlja bazni režim za sledeću iteraciju [3].



Slika 1. Grafovska predstava problema više-periodne rekonfiguracije [2]

4. PRIMENA DINAMIČKOG PROGRAMIRANJA ZA ODREĐIVANJE VIŠE-PERIODNOG PLANA REKONFIGURACIJE

Dinamičko programiranje je naziv popularne tehnike u matematičkoj optimizaciji kojom se značajno smanjuje složenost algoritama. Rešavanje jednog problema svodi na rešavanje potproblema koji su međusobno zavisni. Svaki potproblem se rešava najviše jednom, čime se izbegava ponovno računanje numeričkih karakteristika istog stanja.

Problem planiranja više-periodne rekonfiguracije prikazan je grafom na slici 1. Čvorovi grafa predstavljaju konfiguracije (stanja) DM tokom razmatranih perioda u planiranju. Za svaki čvor je poznato uklopno stanje, a samim tim i troškove gubitaka električne energije u datom periodu.

Troškovi tranzicije iz stanja (i), u periodu (t), u stanje (l), u periodu ($t+1$), jesu jednaki troškovima manipulacije prekidačkim uređajima potrebnih za prelazak iz jednog stanja u drugo [2].

Svaka putanja u grafu koja počinje iz početnog stanja (čvora) i završava u krajnjem stanju (čvoru) predstavlja jedan više-periodni plan rekonfiguracije. Cena plana više-periodne rekonfiguracije predstavlja sumu troškova u svakom periodu, što znači da ukupan trošak odgovara dužini putanje od početnog do krajnjeg čvora grafa. Posledično, optimalan plan više-periodne rekonfiguracije DM odgovara najkraćoj putanji u grafu, a putanje se dobijaju primenom algoritma dinamičkog programiranja opisanog jednačinama (3.2.1) – (3.2.6):

$$C(i, t) = \min \{C(j, t-1) + C_{sw}(j, i) + W(i, t) * C_{loss}\}, \quad (11)$$

$$(i, j) \{1, 2, \dots, N\}, t \in \{0, 1, 2, \dots, T\}$$

$$C_{sw}(i, j) = \sum_{b=1}^{N_b} \text{abs}(x(i, b) - x(j, b)) * C_{sw}, \quad (12)$$

$$(i, j) \{0, 1, 2, \dots, N\}$$

$$C(i, 0) = C_{sw}(i, 0) + W(i, 0) * C_{loss} \quad (13)$$

$$(i) \{0, 1, 2, \dots, N\},$$

gde su:

T – ukupan broj razmatranih perioda,

N – broj različitih konfiguracija,

$C(i, t)$ – trošak optimalnog plana rekonfiguracije za prvih (t) perioda koji se završava konfiguracijom (i),

C_{loss} – jedinična cena gubitaka električne energije USD/kWh,

$W(i, t)$ – gubici električne energije za konfiguraciju (i) i u vremenskom trenutku (t),
 $C_{sw}(j, i)$ – trošak prekidačkih manipulacija potrebnih za prelazak iz konfiguracije i u konfiguraciju j ,
 C_{sw} – cena manipulacije prekidačkim uređajem,
 Nb – ukupan broj grana u distributivnoj mreži,
 $x(i, b)$ – stanje prekidača u grani (b) za konfiguraciju (i).
 Algoritam dinamičkog programiranja za rešavanje problema više-periodne rekonfiguracije DM može se predstaviti sledećim pseudo kodom:

Ulaz:

$brKonfig$ – broj različitih konfiguracija

$switchCost$ – matrica troškova prekidačkih manipulacija između rekonfiguracija, dimenzija [$brKonfig \times brKonfig$]

$lossesCost$ – matrica troškova gubitaka električne energije, dimenzija [$brKonfig \times T$]

Izlaz: $optimum$ - niz rednih brojeva konfiguracija koje čine optimalan plan

Za sve vrednosti $i = 1, brKonfig$

$Cost[i][1] = lossesCost[i][1] + switchCost[i][1];$

$prior[i][1] = -1;$

Kraj

Za sve vrednosti $t = 2, T$

Za sve vrednosti $m = 1, brKonfig$

$Cost[m][t] = \infty;$

$prior[m][t] = -1;$

Za sve vrednosti $j = 1, brKonfig$

$currCost = Cost[j][t - 1] +$
 $switchCost[j][m] + lossesCost[m][t];$

Ako je $currCost < Cost[m][t]$ onda

$Cost[m][t] = currCost;$

$prior[m][t] = j;$

Kraj

Kraj

Kraj

Kraj

$minCost = \infty;$

$opt = -1;$

Za sve vrednosti $m = 1, brKonfig$

Ako je $Cost[m][T] < minCost$ onda

$minCost = Cost[m][T];$

$opt = m;$

Kraj

Kraj

$optimum = \emptyset;$

Za sve vrednosti $t = T, 1$

dodaj opt u $optimum$;

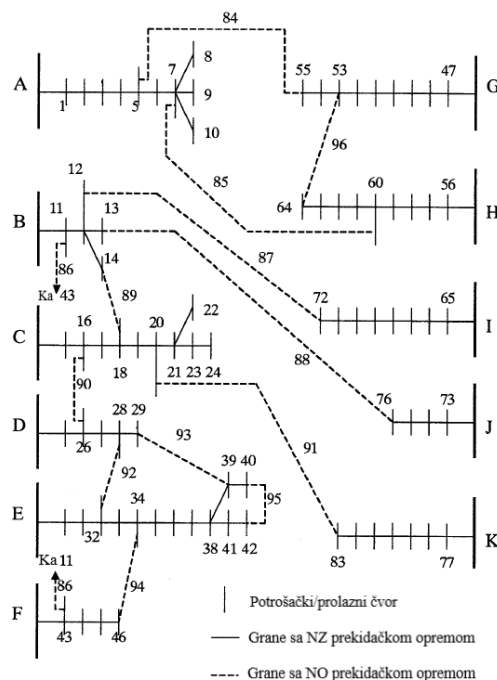
$opt = prior[opt][t];$

Kraj

Vrati $optimum$;

5. VERIFIKACIJA ALGORITMA

Proračun se vrši na distributivnoj mreži prikazanoj na slici 2. DM se sastoji od 11 srednjenaponskih izvoda označenih slovima od A do K. Nominalni napon DM mreže iznosi 11.4 kV. Parametri svih vodova i kablova dati su u [4]. Ukupan broj zatvorenih grana u mreži iznosi 83, dok je broj otvorenih grana 13. Svaka grana DM poseduje rasklopni uređaj a čvorovi su modelovani kao potrošači



Slika 2. Prikaz realne DM [4]

konstantne snage i pripada jednoj od tri grupe potrošača: stambena, komercijalna i industrijska potrošnja.

Potrošači različitih tipova se razlikuju po dinamici promene opterećenja u toku dana, vršnim vrednostima i periodu kada se dostiže maksimum opterećenja. Raspodela potrošača po čvorovima mreže, dnevni dijagrami potrošnje u toku 24h i vršne vrednosti su date u [5].

Najpre je određen skup konfiguracija DM koje mogu da se nađu u optimalnom dinamičkom planu. Date konfiguracije se dobijaju metodom izmena grana za svaki period. Dati skup je prikazan u tabeli 1, kao i sati u toku dana za koje je data konfiguracija optimalna. Na ovaj način dobijeno je 6 konfiguracija distributivne mreže, a kao kandidat uzima se u obzir i inicijalna konfiguracija označena indeksom 0.

Tabela 1. Konfiguracije dobijene metodom izmena grana

Konfiguracija	Redni brojevi otvorenih grana	Period (sat)
K0	83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95	-
K1	6, 12, 31, 34, 40, 60, 71, 83, 85, 88, 89, 90, 91	1, 2, 3, 7, 19, 20, 21, 22, 23, 24
K2	6, 12, 31, 34, 40, 60, 71, 82, 83, 85, 88, 89, 91	4, 6, 15, 16, 17
K3	6, 31, 34, 40, 60, 71, 81, 83, 85, 87, 88, 89, 91	5
K4	6, 12, 31, 34, 40, 60, 83, 85, 86, 88, 89, 90, 91	8, 9, 11, 12, 13, 14
K5	6, 31, 34, 40, 60, 81, 83, 85, 86, 87, 88, 89, 91	9
K6	6, 12, 31, 34, 40, 60, 71, 81, 83, 85, 88, 89, 91	18

Prethodni korak predstavlja ulaz za algoritma dinamičkog programiranja. Koje konfiguracije i tačan raspored datih rekonfiguracija u optimalnom dinamičkom planu zavisi od jedinične cene gubitaka električne energije i cene manipulacija prekidačkim uređajem. Proračun je izvršen za dva para troškova prikazanih u tabeli 2.

Tabela 2. Opis razmatranih planova

Slučaj	C_{sw} [USD]	C_{loss} [USD]
A	0.05	0.20
B	1.00	0.20

U tabelama 3 i 4 dati su rezultati dinamičkog programiranja za planove A i B. Prikazane su konfiguracije za svaki sat u toku dana, takođe i ukupni gubici električne energije, troškovi gubitaka električne energije, troškovi prekidačkih manipulacija i ukupni troškovi za dobijene planove.

Tabela 3. Optimalan plan rekonfiguracije distributivne mreže – slučaj A

Sat	Konfiguracija	Ukupni gubici el. energije [kWh]	Troškovi gubitaka el. energije [USD]	Troškovi prekidačkih manipulacija [USD]	Ukupni troškovi [USD]
1-8	K1	3564,25	712,85	1,3	714,15
9	K6				
10-14	K5				
15-17	K3				
18	K6				
19-24	K1				

Tabela 4. Optimalan plan rekonfiguracije distributivne mreže – slučaj B

Sat	Konfiguracija	Ukupni gubici el. energije [kWh]	Troškovi gubitaka el. energije [USD]	Troškovi prekidačkih manipulacija [USD]	Ukupni troškovi [USD]
1-8	K1	3569,25	713,92	16	729,92
9-24	K2				

U tabeli 5 je prikazano poređenje ukupnih gubitaka električne energije za oba više-periodna plana rekonfiguracija i gubitaka kada bi DM ostala u inicijalnoj rekonfiguraciji tokom celog dana.

Može se uočiti značajno procentualno smanjenje ukupnih gubitaka električne energije i ušteda u ukupnim troškovima. Za plan A, ostvareno je smanjenje ukupnih gubitaka aktivne snage od 9,98%, dok je za plan B ostvareno smanjenje od 9.77%.

Tabela 5. Poređenje uštede i ukupnih gubitaka električne energije

	Gubici el. energije [kWh]	Aps. smanjenje gubitaka [kWh]	Rel. smanjenje gubitaka [%]	Ušteda [USD]
Inic. konf.	3955.66	–	–	–
Plan A	3564.25	391.41	9.98	76.98
Plan B	3569.25	386.41	9.77	61.21

6. ZAKLJUČAK

U ovom radu je određen skup konfiguracija koje je potrebno primeniti tokom posmatranog vremenskog perioda, dok se opterećenje menja. Izložen je jednostavan metod za određivanje plana više-periodne rekonfiguracije distributivne mreže, primenom optimizacionog algoritma dinamičkog programiranja. Značajni faktori prilikom određivanja dinamičkog plana rekonfiguracije DM jesu troškovi manipulacije prekidačke opreme i jedinični trošak gubitaka

električne energije, koji mogu značajno da promene rezultat proračuna više-periodnog plana rekonfiguracije DM. U ovom radu su posmatrana dva para troškova. Prvi kojeg čini nešto manja cena prekidačkih manipulacija i drugi gde se posmatra značajno viša cena prekidačkih manipulacija. Jedinični trošak gubitaka električne energije je jednak u oba posmatrana slučaja.

Na osnovu dobijenih rezultata, može se zaključiti da smanjenje troškova prekidačkih manipulacija rezultira u povećanju broja prekidačkih manipulacija i smanjenju gubitaka električne energije.

S druge strane, ako je cena prekidačkih manipulacija viša, njih će biti manje jer ušteda u smanjenju gubitaka električne energije nije dovoljna da opravda rekonfiguraciju distributivne mreže. Svrha određivanja više-periodnog plana rekonfiguracije jeste ekonomičnije upravljanje, smanjenje ukupnih gubitaka energije i smanjenje troškova eksploatacije DM.

7. LITERATURA

- [1] N.Kovački: *Operativno planiranje rekonfiguracije distributivnih mreža primenom višekriterijumske optimizacije*; Doktorska disertacija, Fakultet tehničkih nauka, Novi Sad, 2017.
- [2] Ž.Popović, N.Kovački: *Multi-period reconfiguration planning considering distribution network automation*, International Journal of Electrical Power and Energy Systems 139 (2022) 107967
- [3] D.Popović, D.Bekut, V.Dabić: *Specijalizovani DMS Algoritmi*; Prosveta, Novi Sad, 2011.
- [4] Ching-Tyong Su, Chu-Sheng Lee: *Network reconfiguration of distribution systems using improved mixed-integer hybrid differential evolution*; Power Delivery, IEEE Transactions on. 18. 2003; 1022 - 1027. 10.1109/TPWRD.2003.813641.
- [5] Z.Ghofrani-Jahromi, M.Kazemi, M.Ehsan: *Distribution switches upgrade for loss reduction and reliability improvement*; IEEE Transactions on Power Delivery, 10.1109/TPWRD.2014.2334645

Kratka biografija



Olivera Kojić rođena je u Sremskoj Mitrovici 1996. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi odbranila je 2019. godine.



Neven Kovački rođen je u Zrenjaninu, 1987. god. Diplomirao je, masterirao i doktorirao na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi redom 2010, 2011. i 2018. godine.

JEDNOSTEPENI I DVOSTEPENI TREKERI VIŠE OBJEKATA

MULTIPLE OBJECT TRACKING WITH END-TO-END TRACKERS AND TRACKING-BY-DETECTION

Katarina Tomić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIČKO I RAČUNARSKO INŽENJERSTVO

Kratak sadržaj – Opisan je problem praćenja objekata sa dvostepenim i jednostepenim trekerima. Performanse trekeru merene su sa više različitih metrika: MOTA, IDF1, HOTA i pod-metrika HOTA. Parametri sistema su varirani u cilju izbora optimalnog rešenja.

Ključne reči: Praćenje objekata, YOLOv5, DeepSORT, FairMOT

Abstract – Multiple object tracking using tracking-by-detection and end-to end trackers is described. Tracker performances were evaluated by MOTA, IDF1, HOTA scores and the submetrics of HOTA. System parameters were varied with the goal of optimal solution choice.

Keywords: Object tracking, YOLOv5, DeepSORT, FairMOT

1. UVOD

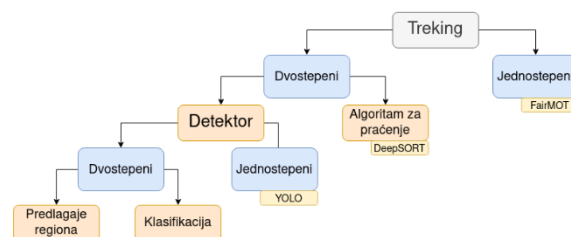
Praćenje objekata (treking) u video zapisima je jedan od klasičnih problema u kompjuterskoj viziji. Predstavlja problem estimacije trajektorije jednog ili više objekata koji se kreću u sceni.

Treking može da se podeli u dva koraka – detekcija i asocijacija. Prvo, algoritam za detekciju određuje lokaciju objekata u frejmovima, a zatim algoritam za praćenje vrši asocijaciju objekata u susednim frejmovima. U klasičnoj kompjuterskoj viziji, detekcija i treking su odvojene, kaskadne operacije, gde asocijacija frejmova zavisi od dobre detekcije objekata. Zato se veliki napor ulaže u unapređivanje algoritama za detekciju.

Trekeri koji imaju odvojene module za detekciju i asocijaciju nazivamo dvostepenim. Sa druge strane, u polju dubokog učenja, ovu zavisnost je moguće prevazići. Neuralne mreže mogu istovremeno da rešavaju oba problema i dele naučene informacije.

Kada postoji jedinstven algoritam (model) koji na ulazu prima sekvencu slika, a na izlazu pruža informaciju o lokaciji objekta i njegov identitet kroz frejmove, taj model nazivamo jednostepenim *end-to-end* trekerom.

Cilj ovog rada jeste razmatranje klasičnog, dvostepenog treking rešenja i novijeg, jednostepenog rešenja, slika 1.



Slika 1. Podela trekera

2. ALGORITMI ZA PRAĆENJE OBJEKATA

Inicijalno, metode praćenja bile su fokusirane na vizuelna obeležja u pokušaju da reše globalni optimizacioni problem pronalaženja trajektorije objekta. Međutim, za rad ovih algoritama potrebna je obrada grupe podataka (eng. *batch*) odjednom, zbog čega nisu adekvatan izbor za *real-time* korišćenje. Neke klasične metode za asocijaciju objekata nemaju potrebu da obrađuju grupe podataka odjednom već mogu da vrše asocijaciju frejm po frejm. Zbog toga su se istraživači okrenuli ka korišćenju jednostavnijih algoritama za asocijaciju, a fokus je na učenju obeležja na kojima se zasniva asocijacija.

2.1. Dvostepeni trekeri

Detekcija i praćenje su odvojeni moduli, gde se za svaki frejm prvenstveno radi detekcija pravougaonih regiona od interesa, a zatim se rezultati prosleđuju na ulaz modula za asocijaciju koji povezuje pravougaone regione kroz frejmove. U literaturi ovaj tip trekera naziva se eng. *tracking-by-detection*, i za potrebe ovog rada nazivamo ih dvostepenim trekerima.

Pod pretpostavkom da (i) detektor koji se koristi ima veoma dobre performanse, odnosno da će tačno proizvesti detekciju za svaki objekat u svakom frejmu, (ii) i da video sekvence imaju veliki broj kadrova u sekundi (eng. *frame rate*), odnosno da je razlika između dva susedna frejma dovoljno mala, treking postaje jednostavan problem koji se može rešiti IOU metrikom [1]. IOU metrika (eng. *intersection over union score*) meri količinu poklapanja dve površine, i daje vrednost $IOU = 1$ ukoliko se dve površine sasvim poklapaju, a $IOU = 0$ ako nemaju dodirne tačke. Algoritam inicijalno bira region sa najvećom verovatnoćom detekcije (a) i računa IOU vrednost za sve regione iz prethodnog frejma (b). Ukoliko nema preklapanja, onda se region (a) zapisuje kao novi objekat:

$$IOU_{(a,b)} = \frac{Area(a) \cap Area(b)}{Area(a) \cup Area(b)} \quad (1)$$

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Branko Brkljač, docent.

Napomenimo da se IOU metrika koristi i u kontekstu samih detektora. Naime, NMS (eng. *non maximum suppression*) algoritam računa IOU metriku za sve predložene regione i odbacuju one koji imaju višu vrednost od izabranog praga (eng. *threshold*). Odbacivanjem regiona koji imaju veoma visoku IOU vrednost (visoko preklapanje) omogućava se eliminacija višestrukih detekcija.

Za ranije pomenuti IOU algoritam usvojena je pretpostavka o kvalitetu detektora i frekvenciji frejmova. Međutim, u realnom sistemu ne možemo da računamo na savršen detektor. U koraku asocijacije teži se kompenzaciji grešaka koje pravi detektor, ali ne samo grešaka već i drugih artefakata u videu, poput okluzija objekta. Na primer SORT algoritam [2], asocira detekcije proizvedene od strane detektora pomoću Hungarian algoritma i Kalman filtera. Ovaj jednostavni pristup postiže dovoljno veliku brzinu izvršavanja da se koristi u *real-time* sistemima i daje zadovoljavajuću tačnost.

Iako SORT algoritam daje dobru tačnost, javlja se drugi problem – identiteti objekata se usled okluzija prebacuju sa objekta na objekat. DeepSORT metoda rešava navedeni problem dodavanjem informacija o izgledu na postojeće podatke o pomeraju objekta. Konkretno, implementirana je konvoluciona neuralna mreža koja povećava robusnost na okluzije i prebacivanje identiteta, a održava brzinu izvršavanja. Ažurirana verzija ovog algoritma je nazvana StrongSORT algoritam, u kome su starije metode zamenjene novijim.

Još jedan pristup rešavanju problema asocijacije jeste reidentifikacija (ReID) [3]. ReID se postavlja kao dodatak na osnovnu mrežu (engl. *backbone*), i postoje dve vrste u odnosu na funkciju cene: klasifikacioni i klasterizacioni tip. Ako se pristupa kao problemu klasifikacije, funkcija cene koristi *ID-discriminative embedding* (IDE). Da bi model mogao da napravi predikciju o identitetu, dodaju se slojevi na postojeću mrežu koji imaju diskriminativnu funkciju. U suprotnom, ako se ReID posmatra kao klasterizacija, za funkciju cene uzima se neka metrika, poput *triplet loss* metrike.

2.2. Jednostepeni trekeri

U razvoju detektora i algoritama za asocijaciju (ReID) napravljen je značajan napredak i postignute su dobre performanse. Međutim, dvostepeni trekeri imaju problem u skaliranju, odnosno kada postoji veliki broj objekata u sceni oni ne mogu da postignu *real-time* vreme izvršavanja. Ovo se dešava zato što asocijacija treba da se primeni na svaki region koji detektor generiše u svakom frejmu. Primećeno je da ukoliko se napravi model u kom detekcija i asocijacija koriste isti model ili mrežu, tj. dele dele informacije, vreme izvršavanja se znatno skraćuje, ali performanse trekera mogu i da opadaju. Jednostepeni trekeri, u literaturi eng. *end-to-end* (ili *one-shot*) trekeri, jesu trekeri koji imaju jedinstveni model za izdvajanje obeležja za oba zadatka [4]. Primer koji će biti analiziran u nastavku je FairMOT algoritam [5].

3. DETEKCIJA OBJEKATA

Na ovom mestu ćemo se ukratko osvrnuti i na problem detekcije objekata, koji se takođe može razmatrati u jednostepenom i dvostepenom obliku, slika 1.

Detekcija objekata u slici podrazumeva rešavanje dva zadatka: 1) detekcije postojanja objekta u slici (problem klasifikacije), i 2) određivanja relativne pozicije objekta (problem lokalizacije). Ovi zadaci mogu da se rešavaju odvojeno, kada takve sisteme nazivamo dvostepenim, ili istovremeno, kada ih nazivamo jednostepenim, slika 1.

Primer dvostepenog detektora je R-CNN model. R-CNN model prvo koristi algoritam za predlaganje regiona od interesa koji potencijalno sadrže objekte, a zatim na svaki pravougaoni region primenjuje klasifikator. Dobijeni sistem daje visoku tačnost, ali je zbog ponovnog izdvajanja obeležja za svaki od potencijalnih regiona od interesa veoma spor, što dvostepene detektore često čini nepogodnim za rad u realnom vremenu. Generalno, najjednostavniji algoritam lokalizacije je metod klizajućeg prozora, gde se cela slika pretraži sa prozorima različitih veličina, a u drugom koraku biraju se regioni koji imaju najveću verovatnoću da sadrže objekat i klasifikuju se kategorije objekata unutar njih. Dakle, predlaganje regiona od interesa predstavlja rešavanje problema lokalizacije objekata u slici, koji se nakon potvrđene detekcije mogu još dodatno lokalizovati u odnosu na prvobitni položaj regiona od interesa.

Jednostepeni detektori lokalizaciju i klasifikaciju realizuju u istom koraku, tj. 1) predviđaju pozicije i dimenzije pravougaonih regiona od interesa i 2) kategorije objekta za celu sliku u jednoj iteraciji. Time se izbegava nezavisno izračunavanje obeležja za svaki od regiona u slici, što dovodi do znatnog povećanja brzine, ali generalno i manje tačnosti. Primer jednostepenih detektora je familija YOLO detektora, koji problem detekcije i lokalizacije formulišu kao jedinstven regresioni problem. Prva verzija YOLO detektora pokazivala je veliki porast u brzini detekcije, ali je tačnost detekcije opala, naročito za male objekte. Vremenom su dodata različita poboljšanja na osnovni YOLO model [6].

Za drugu verziju modela YOLOv2 fokus je bio na poboljšanju tačnosti, stoga ovaj model ima bolju sposobnost generalizacije, veću preciznost za male objekte, manju grešku lokalizacije i povećanu rezoluciju. U trećoj verziji modela, YOLOv3, fokus je bio na ubrzanju ali je tačnost opala. Slični pristupi primenjeni su i za narednu verziju detektora YOLOv4, ali sa više uspeha. Detektor YOLOv5 nije napravljen od strane originalnih autora i zato se generalno smatra da je ime detektora neadekvatno, jer tehnički nije nova verzija, već reimplementacija. Bez obzira na to, YOLOv5 model ima veću tačnost od prethodnih, održanu *real-time* detekciju i uz to, olakšano korišćenje modela.

4. FAIRMOT ALGORITAM

Autori FairMOT algoritma [4] ukazali su na tri razloga pada u performansama jednostepenog trekera. Faktori koji utiču na pad su (i) postojanje unapred definisanih dimenzija za pravougaone regione (eng. *anchors*), (ii) nepravilno deljenje obeležja za zadatak detekcije i ReID - ReID zahteva mnogo više detalja od detekcije. Treći problem (iii) nastaje usled dimenzija obeležja, jer detekcije zahtevaju mnogo manju dimenzionalnost nego reidentifikacija.

FairMOT arhitektura sadrži dve homogene grane nadograđene na osnovnu mrežu. Grana za detekciju koristi CenterNet arhitektu [7] koja nema unapred definisane dimenzije pravougaonih regiona, već estimira pozicije centralne tačke svakog objekta. Grana za reidentifikaciju estimira obeležja objekta opisanog centralnom tačkom. Ove dve grane su sasvim homogene, za razliku od dvostepenih trekera koji reidentifikaciju postavljaju kao sekundarni problem nakon detekcije. FairMOT eliminiše pristrasnost ka detekciji, i uči visoko kvalitetna obeležja za reidentifikaciju.

FairMOT za osnovnu mrežu (engl. *backbone*) koristi ResNet-34 arhitekturu na koju su primenjene određene korekcije (DLA, DCN) kako bi dobijena obeležja bila bolja. CenterNet arhitektura, koja se koristi za granu detekcije, ima tri paralelna izlaza koji procenjuju: mape odziva (eng. *heatmaps*), pomeraj centra (eng. *offset*) i visinu i širinu regiona objekta. NMS algoritam se primenjuje na mape odziva i čuva tačke čiji je odziv iznad definisanog praga. Ove tačke se proglašavaju centrima objekta, a zatim se regresijom procenjuju visina i širina regiona objekta. Izlaz sa informacijom o pomeraju centra služi za precizniju lokalizaciju objekta. Grana za reidentifikaciju generiše obeležja, tako da su razlike u obeležjima između objekata različitih klasa veće nego unutar klasne razlike. Asocijacija se zasniva na ovim obeležjima. Naime, prvo se inicijalizuje N identiteta (eng. *tracklets*), gde je N broj objekata u prvom frejmu. U narednom frejmu, identiteti pravougaonih regiona se povezuju sa prethodnim pomoću Kalman i Hungarian algoritma (DeepSORT algoritam). Detekcijama koje nisu asocirane inicijalizuje se novi identitet, a identiteti koji nisu asocirani sa nekim regionom se čuvaju M broj frejmova pa se ponovo asociraju ako se pojave njihovi regioni, u suprotnom se brišu (starost identiteta).

5. REZULTATI

5.1 Metrike

Oblast praćenja objekata je doživela veliku ekspanziju, delom zahvaljujući industriji autonomnih vozila. Usled porasta interesovanja, pojavljuju se i nove ideje, novi doprinosi u načinima evaluacije. Praćenje objekata je složen zadatak za koji je potrebna tačna detekcija, lokalizacija i asocijacija kroz vreme. Za kvantitativno merenje trekera, korišćene su sledeće metrike: MOTA, IDF1 i HOTA.

MOTA (eng. *Multiple Object Tracking Accuracy*) je najčešće korišćena metrika za evaluaciju trekera. Meri se poklapanje detekcija i anotiranih objekata (engl. *Ground Truth – GT*). Greške koje se javljaju pri poklapanju su lažni negativni (FN) i lažni pozitivni (FP), koji su mere detekcije. Još jedna greška koja ulazi u računanje MOTA metrike je prebacivanje identiteta (IDSW), što predstavlja meru asocijacije. MOTA (2) ne meri grešku lokalizacije, i naglašava performanse detekcije preko asocijacije.

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t (GT)_t} \quad (2)$$

IDF1 (eng. *Identity F1 score*) akcentuje merenje tačnosti asocijacije više nego detekcije i koristi se uglavnom kao propratna metrika uz MOTA. Meri poklapanje anotiranih trajektorija objekta sa predikovanim trajektorijama, i

računa odnos tačno identifikovanih detekcija prema ukupnom broju izračunatih detekcija. IDF pruža estimaciju o broju jedinstvenih objekata u sceni, takođe kao i MOTA ne meri grešku lokalizacije.

$$IDF1 = \frac{|IDTP|}{|IDTP| + 0.5|IDFN| + 0.5|IDFP|} \quad (3)$$

MOTA i IDF1 prenaplaćavaju bitnost detekcije i asocijacije, respektivno. HOTA (eng. *Higher Order Tracking Accuracy*) eksplicitno meri obe vrste greške i kombinuje ih da bi pružila jedinstvenu vrednost za evaluaciju trekera. Pored toga, HOTA meri i grešku lokalizacije što nije slučaj kod MOTA i IDF1 metrike. Ovakav pristup evaluacije pruža detaljan uvid u greške koje treker pravi i tako omogućava precizno korigovanje.

HOTA se dekomponuje na pod-metrike koje opisuju zasebne vrste problema. Na primer, tačnost detekcije (DetA) (4) je procenat poklapanja detekcija, dok je tačnost asocijacije (AssA) (5) poklapanje asociranih trajektorija, uprosečeno kroz sve detekcije.

$$DetA = Det - IoU = \frac{|TP|}{|TP| + |FN| + |FP|} \quad (4)$$

$$AssA = \frac{1}{|TP|} \sum Ass - IoU(c) \quad (5)$$

Tačnost lokalizacije (LocA) (6) računa se kao prosečno poklapanje anotiranih detekcija sa procenjenim, nad celim skupom podataka.

$$LocA = \frac{1}{|TP|} \sum Loc - IoU(c) \quad (6)$$

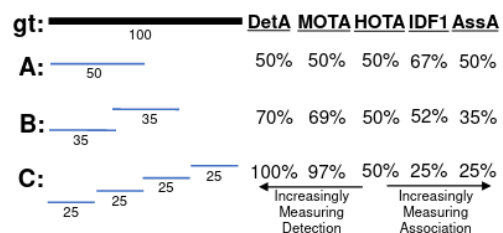
Konačni HOTA indeks (7) je geometrijska srednja vrednost DetA i AssA za različite pragove lokalizacije.

$$HOTA_\alpha = \sqrt{DetA_\alpha \cdot AssA_\alpha} \quad (7)$$

5.2. Diskusija

Testirane su performanse dva modela za praćenje objekata: dvostepeni YOLO model sa StrongSORT algoritmom i jednostepeni FairMOT model. Sa njihovih repozitorijuma preuzete su težine već obučenih modela i optimalna podešavanja. Kao što je opisano, ovi modeli su dva sasvim različita pristupa istom problemu, ali performanse oba se rangiraju pri samom vrhu u njihovim kategorijama. Korišćen je MOT16 skup [8] i oba modela su testirana za samo jednu klasu – ‘pešak’.

Na slici 2 mogu se uočiti glavne razlike između MOTA, IDF1 i HOTA metrika. Ilustrovana su tri različita trekera poređana po porastu tačnosti detekcije i smanjenju tačnosti asocijacije. Dok HOTA balansirano uzima u obzir oba zadatka, MOTA je očigledno naklonjena ka detekciji, a IDF1 ka asocijaciji.



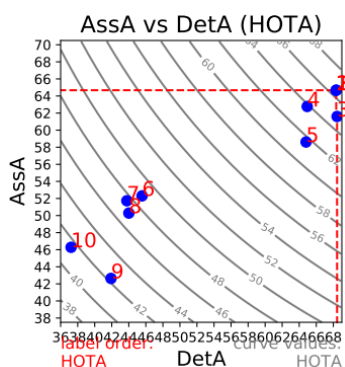
Slika 2. HOTA i druge metrike za tri trekera (A, B, C) sa sve boljom detekcijom i lošijom asocijacijom [9].

Parametri YOLO trekera koji su menjani u svrhu analize njihovog uticaja na celokupnu tačnost jesu: skor pouzdanosti detektora, odnosno donji prag verovatnoće detekcije, zatim maksimalna starost neasociranog identiteta, prag za asocijaciju, i IOU prag za NMS algoritam. Parametri koji utiču na performanse FairMOT modela su: NMS prag, prag pouzdanosti detekcije i dimenz. ReID modela. Sekvence slika propuštene su kroz svaki od ovih modela i poređeni su rezultati. Modeli su numerisani kao u tabeli 1. Originalna podešavanja odgovaraju „Modelu 2.0“ i „Modelu 1.0“.

Tabela1. Promene parametara u odnosu na polazne modele

Model 2.1	Model 2.0	Model 2.2	Model 2.3	Model 2.4
Prag pouzdanosti +0.2	Osnovni dvostepeni model	Max starost -15	Prag za asocijaciju +0.3	IOU +0.2
Model 1.1	Model 1.0	Model 1.2	Model 1.3	Model 1.4
NMS -0.2	Osnovni jednostepeni model	ReID 1024	Prag pouzdanosti -0.2	ReID 512

Performanse trekera na HOTA pod-metrikama detekcije i asocijacije prikazane su na slici 3. Plave tačke su trekeri sa različitim podešavanjima, numerisani kao u tabeli 1. Na slici 2 može se primetiti da se za različite kombinacije tačnosti detekcije i asocijacije dobijaju isti HOTA indeksi. Zato su HOTA indeksi ilustrovani sivim izolinijama.



Slika 3. Tačnost trekera u detekciji i asocijaciji. Crvena linija predstavlja Pareto optimalni front opisan u [9].

Povećanje praga pouzdanosti detekcije dovodi do skoka u tačnosti YOLO detektora (Model 2.1) u odnosu na osnovni model (Model 2.0). Ovo ponašanje je očekivano, jer povećavanjem praga eliminišemo detekcije koje su manje sigurne. Menjanje drugih YOLO parametara dovodi do pada tačnosti. Autori FairMOT modela sugerišu da prevelika dimenzionalnost ReID modela negativno utiče na tačnost. Na slici 3. možemo videti i da u odnosu na osnovni model (Model 1.0), promene parametara jednostepenog trekera dovode do pogoršanja rezultata. Svi YOLO modeli imaju manju HOTA metriku u odnosu na FairMOT modele. U tabeli 2 prikazane su i vrednosti drugih metrika za najbolje modele iz obe grupe.

Tabela 2. Poređenje metrika za najbolje modele.

	MOTA	IDF1	HOTA
YOLO Model 2.1	49.69	61.33	48.46
FairMOT Model 1.0	83.36	80.76	66.33

Primećujemo da u slučaju ova dva trekera, FairMOT daje bolje rezultate na svim metrikama.

6. ZAKLJUČAK

Rad analizira dva rešenja za praćenje objekata u realnom vremenu. Dvostepeni YOLO+StrongSORT model kombinuje različite algoritme, dok je jednostepeni FairMOT model složenije strukture i rešava problem praćenja objekata na celovit način. Na osnovu eksperimenata pokazano je da FairMOT daje bolje rezultate, ali je opšti zaključak da je dvostepeni model na bazi YOLO detektora intuitivniji za korišćenje.

7. LITERATURA

- [1] Bochinski, Erik, Volker Eiselein, and Thomas Sikora. "High-speed tracking-by-detection without using image information." *2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS)*. IEEE, 2017.
- [2] Wojke, Nicolai, Alex Bewley, and Dietrich Paulus. "Simple online and realtime tracking with a deep association metric." *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017.
- [3] Zheng, Zhedong, Liang Zheng, and Yi Yang. "A discriminatively learned CNN embedding for person reidentification." *ACM Transactions on multimedia computing, communications, and applications (TOMM)* 14.1 (2017): 1-20.
- [4] Munjal, Bharti, et al. "Joint detection and tracking in videos with identification features." *Image and Vision Computing* 100 (2020): 103932.
- [5] Zhang, Yifu, et al. "FairMOT: On the fairness of detection and re-identification in multiple object tracking." *International Journal of Computer Vision* 129.11 (2021): 3069-3087.
- [6] Zaidi, Syed Sahil Abbas, et al. "A survey of modern deep learning based object detection models." *Digital Signal Processing* (2022): 103514.
- [7] Zhou, Xingyi, Dequan Wang, and Philipp Krähenbühl. "Objects as points." *arXiv preprint arXiv:1904.07850* (2019).
- [8] <https://motchallenge.net/data/MOT16/> (10. okt. 2022).
- [9] Luiten, Jonathon, et al. "Hota: A higher order metric for evaluating multi-object tracking." *International journal of computer vision* 129.2 (2021): 548-578.

Kratka biografija:



Katarina Tomić rođena je u Novom Sadu 1996. god. Upisala je master studije na Fakultetu tehničkih nauka na studijskom programu Energetika, elektronika i telekomunikacije – Obrada signala. Osnovne akademske studije završila je 2019. godine na studijskom programu Biomedicinsko inženjerstvo.

РЕАЛИЗАЦИЈА РЕПРОДУКОВАЊА СЛИКЕ НА LED ДИСПЛЕЈУ СА ПАСИВНОМ МАТРИЦОМ

REALIZATION OF PICTURE REPRODUCTION ON LED DISPLAY WITH PASSIVE MATRIX

Стефан Новаковић, Факултет Техничких Наука, Нови Сад

Област – ПРИМЕЊЕНА ЕЛЕКТРОНИКА

Кратак садржај – У оквиру овог рада описан је рад LED дисплеја са пасивном матрицом и описана је софтверска реализација репродуковања слике на истом.

Кључне речи: LED дисплеј, Python, Raspberry Pi, C библиотека

Abstract – This paper describes working principle of LED display with passive matrix and software implementation of image reproduction on it.

Keywords: LED display, Python, Raspberry Pi, C library

1. УВОД

Тема рада је реализација репродуковања слике на LED дисплеју са пасивном матрицом. Како смо свакодневно окружени различитим дисплејима, циљ овог рада је упознавање технологије приказа слике. Иако коришћени дисплеј представља помало застарелу технологију за приказ садржаја, јер се не користи у рачунарима, мобилним телефонима, телевизорима и другим савременим уређајима, већ само као рекламни панели, за сценску кореографију или у возилима јавног превоза, ови дисплеји представљају добру основу за разумевање рада дисплеја. Постоје разне поделе дисплеја на основу њиховог начина адресирања, материјала које користе, начина рада итд. Оно што је заједничко сваком дисплеју је да се сваки пиксел састоји од три извора светлости, црвеног, зеленог и плавог извора светлости. Комбинацијом интезитета ова три извора светлости, може се добити широк спектар боја.

Осим мотивације да се више упозна начин приказивања слике на дисплеју, један од главних разлога зашто је изабрана ова тема је и мотивација за упознавањем са програмским језиком Python. Python представља моћан програмски језик, који своју примену налази у више различитих области рачунарства. Он може да се користи у роботички, аутоматизацији, Web апликацијама, анализи података, вештачкој интелигенцији и још многим другим областима.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Јован Бајић, ванр. проф.

2. LED MATRIX ДИСПЛЕЈ

У зависности од матрице приказа, чији су елементи светлосно-емитујући пиксели, дисплеје можемо поделити на активне и пасивне. Ова подела се односи на начин адресирања. Код дисплеја са пасивном матрицом сваки пиксел реагује (емитује светлост) само кад је адресиран, а у осталим интервалима је неактиван (мултиплексни режим рада). Дисплеји са пасивном матрицом представљају старију технологију и добри су само за статичан приказ (слика, текст и сл.). Главни недостаци ових дисплеја су споро време одзива, преслушавање између линија, мали максимални контраст и израда матрице може бити јакo комплексна.

2.1. LED дисплеј са пасивном матрицом

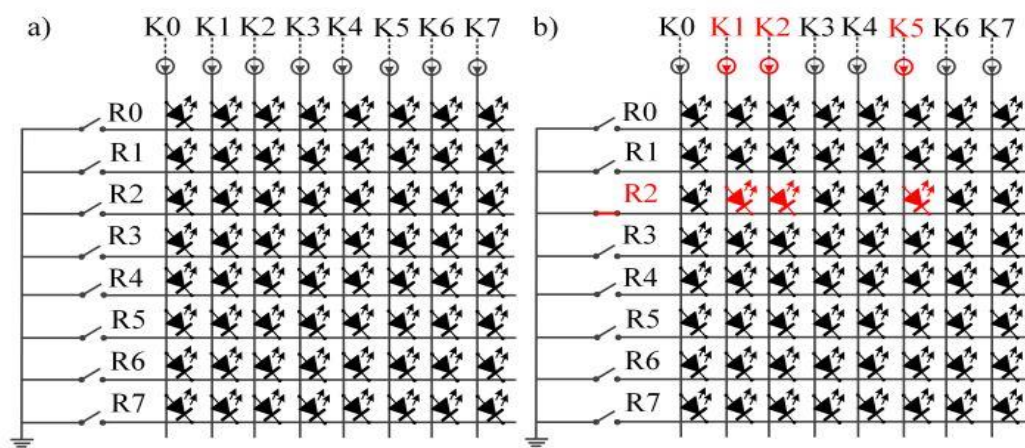
Редови дисплеја се реализују преко прекидача, који у пракси представљају транзисторе спојене на масу, а колоне на струјне изворе који служе за напајање LED диода. Испис садржаја на дисплеју врши се ред по ред. Спуштањем једног од прекидача затвара се струјно коло и селекује одговарајући ред. За LED диоде које треба да емитују светлост укључују се струјни извори у одговарајућим колонама.

LED диоде се држе у укљученом стању одређено T_{ON} време. Након тога, дати ред се искључује. Описани поступак се понавља за сваки следећи ред, почев од првог до последњег. Активирање струјних извора у колонама одређује који пиксели ће емитовати светлост у селекованом реду. Овакој реализацији је неопходно стално освежавање садржаја дисплеја, како би се садржај константно приказивао.

При том, укупно време освежавања мора бити кратко како људско око не би регистровало треперење LED диода. Ово подразумева да дисплеј већих димензија захтева веће време освежавања. Диода која треба да емитује светлост биће укључена време T_{ON} , а искључена преостало време T_{OFF} до поновног укључења:

$$T_{OFF} = (\text{број редова} - 1) * T_{ON} \quad (1)$$

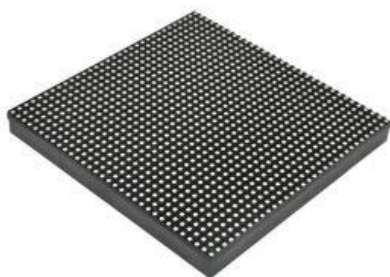
Ако је време освежења дисплеја ($T_{ON} + T_{OFF}$) довољно мало људско око неће бити у стању да региструје тренутке укључења и искључења LED диоде, већ ће регистровати средњи интезитет светлости одређен временим T_{ON} и T_{OFF} . Што је време T_{OFF} дуже у односу на T_{ON} средњи интезитет који региструје око биће мањи.



Слика 1. а) 8x8 LED дисплеј са пасивном матрицом б) 8x8 LED дисплеј са селектованим редом 2 и укљученим колонама 1, 2 и 5

2.2. LED панел

За реализацију овог пројекта користило се 6 LED панела спојених редно. Сваки LED панел садржи 1024 пиксела распоређена у матрици од 32 реда и 32 колоне. Сваки пиксел се састоји од засебних црвених, зелених и плавих LED диода унутар истог кућишта.



Слика 2. 32x32 RGB LED панел

LED панел је подељен вертикално на горњи (32 колоне и 16 редова) и доњи потпанел (32 колоне и 16 редова).

Горњи и доњи потпанел су такође вертикално подељени на по два сегмента, од по 32 колоне и 8 редова. Сегменти у оквиру истог панела су повезани редно. Колонама панела управља се помоћу једног сета управљачких сигнала, а редовима помоћу другог [1].

3. РЕАЛИЗАЦИЈА РЕПРОДУКОВАЊА СЛИКЕ

За хардверску реализацију овог пројекта коришћен је *Raspberry Pi 4* и LED дисплеј који се састоји од шест панела 32x32 HUB-75 редно везаних.

Raspberry Pi 4 представља мини рачунар опште намене са могућношћу прикључивања нестандартне опреме. На *Raspberry Pi 4* се налази оперативни систем *Raspbian OS* који је заснован на *Debian LINUX* оперативном систему и оптимизован је за рад са *Raspberry Pi* уређајима [2]. LED дисплеј се напаја посебним напајањем од 5V.

Сваки панел се састоји од два порта, улазног и излазног који служи за везивање више панела.

Сваки порт се састоји од улазних пинова A, B, C, R1, G1, B1, R2, G2, B2, OE, LAT и CLK.

3.1. Реализације апликације у Python програмском језику

Осим што *Raspberry Pi* промовише *Python* као главни програмски језик, један од главних разлога зашто је апликација првобитно реализована у том програмском језику је жеља да се научи *Python*. Разлог да се научи *Python* лежи у његовом широком спектру могућности, налазећи примену у роботизи, аутоматизацији, развоју Web апликација, анализи података, вештачкој интелигенцији итд.

За имплементацију апликације најбитније функције су функције за контролу GPIO излаза:

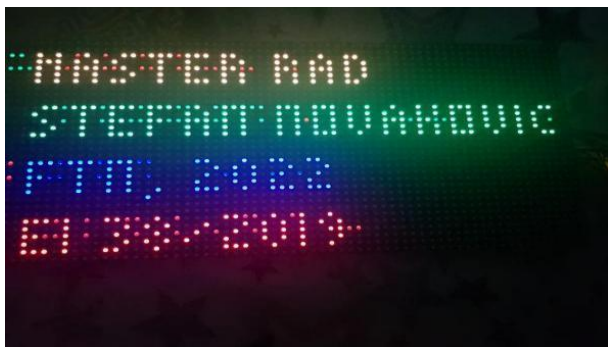
- **selectRow(row)** - функција којом се врши селекција реда LED панела (постављање стања A, B и C улазних прикључака панела). Улазни параметар је редни број реда (0-7).
- **latch()** - функција којом се подаци пребацују у паралелни излазни регистар променом стања улазног прикључка LAT панела са "1" на "0".
- **clock()** - функција којом се генерише једна периода такт сигнала, позивањем ове функције врши се померање података у померачком регистру променом стања улазног прикључка CLK са "1" на "0".
- **setColor(color, bit)** - функција којом се поставља стање улаза R1, G1, B1, R2, G2 и B2 LED панела. Улазни параметри су редни број бита променљиве color који се поставља на претходно наведене улазе.

Сигнал OE (енг. *Output Enable*), служи за укључивање и искључивање активних излазних регистра, који директно адресирају појединачне колоне. Осим функција за контролу функције GPIO излаза, битно је поменути и остале функције за исписивање садржаја, а то су: функције за цртање (*displayDrawRectangle()*, *displayDrawCircle()* итд.), функције за писање текста (*displayPutChar()*, *displayPrint()* итд.) и функција за приказивање слике (*displayLoadImage()*).

Главну функцију у реализацији представља функција *refreshDisplay()* која служи, као што јој и само име каже, за освежавање садржаја који се приказује на дисплеју. Идеја за апликације које приказују различит садржај на дисплеју је да се имплементација свих

потребних функција налази у једном модулу, док се свака апликација која приказује одређени садржај састоји од task-а који се позива само приликом иницијализације и цикличног task-а који позива само функцију `refreshDisplay()`. Функције које се налазе у иницијалном task-у уписују у низ `FRAME_BUFFER` у зависности од тога шта треба да прикажу на дисплеју, а функција `refreshDisplay()`, као што је већ поменуто, приказује тај садржај на дисплеју. За управљање `GPIO` пиновима користила се `RPi.GPIO` библиотека. Све пинови, који су спојени са пиновима `LED` дисплеја, потребно је подесити као излазне пинове (`A`, `B`, `C`, `R1`, `G1`, `B1`, `R2`, `G2`, `B2`, `OE`, `LAT` и `CLK`).

Као што је већ и поменуто, да би се садржај стално приказивао на дисплеју, потребно је периодично понављати поступак исписивања садржаја, а да при том, укупно време освежавања буде кратко како људско око не би регистровало треперење. Што је већи дисплеј, то постаје изазовније. Приликом покретања апликације примећени су основни недостаци пасивних дисплеја: треперење и појава *ghosting* ефекта.



Слика 3. Испис садржаја на дисплеју помоћу Python апликације

Постоји више разлога појаве ових недостатака, један од главних је то што је *Python* програмски језик јако спор за извршавање. Други разлог лежи у томе што се на *Raspberry Pi* налази оперативни систем који одузима доста процесорске моћи. Време потребно да се изврши функција `refreshDisplay()` је реда ~85ms.

```
0.08538103103637695
0.0854494571685791
0.08484554290771484
0.08485221862792969
0.08454108238220215
0.08472323417663574
0.08444547653198242
0.08467674255371094
0.08456110954284668
0.08563470840454102
```

Слика 4. 10 узорака мерења времена извршавања функције `refreshDisplay()` са реализацијом у *Python*-у

Постоји више начина за превазилажење ових проблема: коришћење *low-level* програмских језика (нпр. *C*), употреба *overclocking*-а (повећање фреквенције процесора), ослобађање *RAM* меморије (брисање непотребних апликација), додељивање већег приоритета апликацији (опсег приоритета је од -20 до 19, где је -20 највећи приоритет, док је по стандарду нашој апликацији додељен

приоритет 0), употреба квалитетног извора напајања, употреба система хлађења, активација *ZRAM*-а итд.

3.2. Зашто је *Python* толико спор?

Python је динамички превођен језик. У програмским језицима као што су *C*, *C++* или *Java* све променљиве су статичке, то значи да програмер сам одређује тип за своју променљиву: `int my_var = 1;` У *Python*-у можемо једноставно да напишемо `my_var = 1`. Иако је ово веома згодно за програмера, има доста мана.

Компајлирање значи превођење једног програмског језика у други, обично у нижи ниво од превођеног језика. Када се компајлира програмски језик *C*, он се преводи у машински језик, који представља инструкције за сам процесор. *Python* се компајлира у *bytecode* јер је он динамички превођен језик. Ми не одређујемо тип за нашу променљиву, ми морамо да чекамо вредност променљиве како би одредили њен тип и тек онда превели у машински језик. Ово је оно што преводилац ради. У статички превођеним језицима, компајлирање и интерпретација се дешава пре извршавања програма. Закључак: програм је успорен компајлирањем и интерпретацијом која се дешава у току извршавања самог програма [3].

4. РЕАЛИЗАЦИЈА С БИБЛИОТЕКЕ

Као што смо већ упознали најважнија функција у репродуковању слике на `LED` дисплеју представља `refreshDisplay()`, којој је потребно неколико *ms* (тачније око ~80ms) за њено извршавање. Што значи да се током 1 секунде, слика прикаже приближно 12 пута, или 12fps (енг. *frame per second*). Не постоји тачан број колико људско око може уочити слику у секунди. Просек је око 30-35fps, али постоје и изузеци. Наравно, уочавање брзине смењивања слике зависи и од њихове међусобне сличности. Па ипак, број који човек може уочити и обрадити креће се тек око 24 до 28fps [4]. Са реализацијом у *Python*-у, та вредност је дупло већа од наше, и из тог разлога ми видимо “треперење” на дисплеју. Зато ће функција `refreshDisplay()` бити реализована у *C*, спакована у *Python* библиотеку и позивана из *Python* апликације.

У функцији `refreshDisplay()` се врши манипулација излазним `GPIO` пиновима. Док се за реализацију у *Python*-у користила библиотека `RPi.GPIO` за манипулацију `GPIO` пиновима, за реализацију у *C* ће се користити библиотека `bcm2835`. Ова библиотека омогућава приступ `GPIO` и другим `IO` функцијама на *Broadcom BCM 2835* чипу. Иако се на *Raspberry Pi 4* налази чип *Broadcom BCM 2711*, ова библиотека је компатибилна и са тим чипом. Она обезбеђује функције за читање дигиталног улаза и сетовање дигиталног излаза, користећи `SPI` и `I2C` комуникацију за приступ системским бројачима. Детекција догађаја на пину је омогућена помоћу *polling*, док прекиди (енг. *interrupts*) нису могући. Ова библиотека ради на свим врстама *Debian* оперативног система. Такође је компатибилна са *C++*. Инсталира се као *header* фајл и недељена библиотека на било ком *LINUX* репозиторијуму, али наравно њена примена је могућа само на уређајима са *BCM2835* чипом и чиповима који су компатибилни са њим.

Language	Library	Tested / version	Square wave
Shell	/proc/mem access	2015-02-14	2.8 kHz
Shell / gpio utility	WiringPi gpio utility	2015-02-15 / 2.25	40 Hz
Python	RPI.GPIO	2015-02-15 / 0.5.10	70 kHz
Python	wiringpi2 bindings	2015-02-15 / latest github	28 kHz
Ruby	wiringpi bindings	2015-02-15 / latest gem (1.1.0)	21 kHz
C	Native library	2015-02-15 / latest RaspPi wiki code	22 MHz
C	BCM 2835	2015-02-15 / 1.38	5.4 MHz
C	wiringPi	2015-02-15 / 2.25	4.1 – 4.6 MHz
Perl	BCM 2835	2015-02-15 / 1.9	48 kHz

Слика 5. Упоредба брзине приступа GPIO пиновима различитих библиотека [5]

Када је цела функција имплементирана у C, потребно је ископајлирати и генерисати .so фајл (тај фајл представља дељени објектни фајл за дељену библиотеку). Да бисмо користили функције потребно је укључити Python пакет *ctypes*. Он представља библиотеку која обезбеђује C компатибилне типове података и позивање функција из DLL или дељених библиотека. Покретањем команде *CDLL* из пакета *ctypes* врши се преузимање динамичког линка библиотека. Цикличним позивањем функције *refreshDisplay()*, добијамо изглед дисплеја:



Слика 6. Испис садржаја на дисплеју помоћу C апликације

И коначно време потребно за извршавање функције *refreshDisplay()* износи у просеку ~4.5ms, што значи да број слика или освежавања током 1s износи ~220fps, што је много више од вредности које људско око може да детектује. Оваквом реализацијом, „треперење“ се више не појављује на дисплеју.

```
0.004473209381103516
0.004437923431396484
0.005208730697631836
0.005080223083496094
0.005780935287475586
0.00539708137512207
0.004472017288208008
0.004460334777832031
0.004877567291259766
0.0047566890716552734
```

Слика 7. 10 узорака мерења времена извршавања функције *refreshDisplay()* са реализацијом у C

5. ЗАКЉУЧАК

У претходним поглављима смо се више упознали са дисплејима са пасивном матрицом и доказали да без обзира колико је Python погодан за различите апликације, и даље је C бољи избор за временски критичне функције. Python није погодан за коришћење у *embedded* системима за извршавање временски критичних задатака. Ово би још више дошло до изражаја да смо користили веће дисплеје.

Оно што се исто може приметити са Сlike 5. је да постоји простор за даље убрзавање приступа GPIO пиновима, уместо коришћења неке GPIO библиотеке, директним приступом регистрима. Један од корака за убрзавање извршавања наше *refreshDisplay()* функције би био управо тај.

6. ЛИТЕРАТУРА

- [1] J. С. Бајић, “LED диспеј”, <https://www.optolab.ftn.uns.ac.rs/images/NASTAVA/OE/files/lab/pdf/2021/11-LED-panel.pdf>; [Приступљено у октобру 2022.]
- [2] http://www.racunarstvo.matf.bg.ac.rs/MasterRadovi/2014_03_10_Ksenija_Grujic-Petrovic/rad.pdf; [Приступљено у октобру 2022.]
- [3] Mike Huls, “Why Python is so slow and how to speed it up”, <https://towardsdatascience.com/why-is-python-so-slow-and-how-to-speed-it-up-485b5a84154e> [Приступљено у октобру 2022.]
- [4] <https://www.healthline.com/health/human-eye-fps> [Приступљено у октобру 2022.]
- [5] <https://codeandlife.com/2012/07/03/benchmarking-raspberry-pi-gpio-speed/> [Приступљено у октобру 2022.]

Кратка биографија:



Стефан Новаковић рођен је 23.08.1995. у Новом Саду. Основну школу “Петар Кочић” у Темерину завршио је 2010. године. Гимназију “Светозар Марковић” у Новом Саду завршио је 2014. године. Исте године уписује Факултет техничких наука, смер Енергетика, електроника и телекомуникације. У октобру 2019. године завршава основне студије, након чега уписује мастер студије такође на Факултету Техничких Наука, смер Примењена електроника.

**UPOREDNI PRISTUP GENERISANJU GEOMETRIJE OKRUŽENJA VIDEO IGARA
PRIMENOM FOTOGRAMETRIJE I DIGITALNOG VAJANJA****COMPARATIVE APPROACH OF ENVIRONMENTAL GEOMETRY GENERATION IN
VIDEO GAMES USING PHOTOGRAMMETRY AND DIGITAL SCULPTING**

Aleksandra Bobić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSKA GRAFIKA

Kratak sadržaj – U radu je prikazano istraživanje uporednog pristupa za generisanje 3D modela koji se koriste u video igrama. 3D modeli su dobijeni fotogrametrijskim modelovanjem i digitalnim vajanjem, gde su upotrebom ankete ove dve metode međusobno upoređene radi dobijanja relevantnih podataka koji model izgleda vizuelno bolji.

Ključne reči: Fotogrametrija, Digitalno vajanje, Video igre, Optimizacija

Abstract – The paper presents research on a comparative approach to generating 3D models for video games. 3D models are generated using photogrammetry and digital sculpting, while using poll these two methods are compared for getting relevant data on which model looks better.

Keywords: Photogrammetry, Digital sculpting, Video games, Optimization

1. UVOD

Industrija video igara se svakim danom sve više razvija usled velikog napretka u oblasti nauke i tehnologije. Grafika i vizuelni prikazi su iz dana u dan sve bolji pojavom različitih metoda i tehnika. Svaki 3D umetnik koji se bavi nekom od oblasti računarske grafike teži da stvori što realističnije okruženje.

S obzirom na konstantan razvoj na tržištu, 3D umetnici mogu da prikažu svoju zamisao sa mnogo većim stepenom realizma nego što je to pre bilo moguće, te ukoliko se radi o umetničkom stilu u video igrama koji se naziva fotorealizam, očekuje se da 3D modeli budu maksimalno približni realnim.

1.1. Oblast istraživanja

3D modelovanje je proces kreiranja trodimenzionalnih reprezentacija objekta ili površi. 3D modeli se generišu u okviru računarskih softvera za 3D modelovanje, a njihova upotreba je zastupljena u mnogim industrijama. Moguća je primena za niz različitih oblasti, ali trenutno je u industriji video igara najzastupljenija.

Postoje različite tehnike 3D modelovanja koje omogućavaju dobre rezultate. Trenutno najzastupljenije tehnike za dobijanje realističnih rezultata iz okruženja koje nas svakodnevno okružuje jesu:

- digitalno vajanje namenjeno za organske forme, koje se oslanja na standardan način modelovanja
- tehnika koja nije toliko zastupljena u video igrama, ali se njena primena sve više razvija, fotogrametrija

3D vajanje, odnosno digitalno vajanje je tehnika modelovanja gde 3D umetnik kreira 3D modele na računaru upotrebom softvera.

Fotogrametrija je proces izvlačenja vizuelnih podataka iz niza fotografija sa lokacije u stvarnom svetu da bi se generisali visokopoligonalni modeli sa potpuno identičnom teksturom originalu.

1.2. Predmet istraživanja

Iako fotogrametrija kao alat postoji već dugo, bilo je potrebno mnogo vremena da resursi potrebni za njenu primenu dostignu nivo mašte 3D umetnika koji stvaraju video igre. Konstantnim razvojem i napretkom dostigla je nivo da skoro svako okruženje koje bi naš um mogao da zamisli, može da stvori. Stvaranjem 3D sadržaja procesom fotogrametrije prati se potpuno drugačiji proces rada koji zamenjuje dugotrajan proces digitalnog vajanja i ručnog pravljenja tekstura praćenjem reference, odnosno fotografija. Fotogrametriju trenutno koriste neke kompanije za razvoj AAA igara, ali je moguće efikasno koristiti za ozbiljne igre kao i simulacije.

Glavni fokus u ovom radu biće istražiti oblasti fotogrametrije i digitalnog vajanja primenjene za modelovanje organskih formi okruženja video igara. Kako bi se ispitalo da li je ova tema relevantna, potrebno je sagledati nekoliko primera iz ove oblasti.

1.3. Pregled stanja u oblasti

Kako se razvijala tehnologija, razvijali su se i napredniji alati koji su poboljšavali sam proces modelovanja, odnosno dobijanja gotovih 3D modela. Upotreba različitih softvera znatno ubrzava rad, ali se i dalje javljaju pitanja kako doneti ispravnu odluku, kako bi se na najefikasniji način postigao zadovoljavajući rezultat. Pregled stanja u oblasti će započeti sagledavanjem procesa fotogrametrije i nakon toga digitalnog vajanja, kako bi osnovni pojmovi mogli biti razjašnjeni, nakon čega će se videti njihova primena u industriji video igara.

1.3.1. Fotogrametrija

Fotogrametrija nije nova oblast nauke jer se pojavila još u 19. veku ali što se tiče industrije igara, 3D umetnici nisu bili za korišćenje metoda fotogrametrije jer računari i

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Marko Jovanović, docent.

softveri nisu bili pogodni za njenu primenu, pa je ova inovacija ostala niskog profila [1].



Slika 1. Prikaz celokupnog procesa dobijanja fotogrametrijskog modela.

Razvojem tehnologija, fotogrametrija je postala je pristupačna i za mobilne telefone. Razvijeni su softveri koji omogućavaju primenu ovog alata gde uz pomoć algoritama za merenje, softver može da spoji fotografije i da rekonstruiše 3D model. Lidar koji je sada deo novih generacija Ajfon (eng. Iphone) telefona, namenjen za detekciju i domet setlosti, koristi pulsni laser za merenje udaljenosti gde mu je meta objekat ili površina. Uz pomoć ove opcije i kamere telefona, moguće je snimiti celu scenu, čak enterijer i dobiti izgenerisan 3D model. Još jedna primena fotogrametrije jeste fotografisanje površina poput opeke, kamena, tepiha ili betona i kreiranje setova tekstura koji se koriste kasnije kao texture za ručno izmodelovane modele.

1.3.2. Digitalno vajanje

3D modelovanje, jedna od komponenti rendering pajplajna (eng. rendering pipeline)¹ je proces generisanja modela u tri dimenzije uz pomoć specijalizovanih softvera, gde se manipuliše ivicama, verteksima i poligonima u 3D prostoru kako bi se dobio 3D model. Digitalno vajanje je tehnika modelovanja za kreiranje 3D modela organske forme. Ova metoda omogućava stvaranje slobodnih formi od nule, rad na teksturama, elemente digitalnog slikanja i smatra se da doprinosi umetničkom osećaju. Postoje programi za vajanje kao što su 3D-Coat, Zbrush, Mudbox koji omogućavaju integraciju svojih načina funkcionisanja sa standardnim programima za 3D modelovanje i renderovanje.

Tehnika koja se koristi prilikom digitalnog vajanja je skalpting. Skalpting predstavlja metodu organskog modelovanja koja se koristi za kreiranje objekata sa većim brojem poligona. Obično podrazumeva složeniju anatomiju, nema veze da li karakteri izgledaju realistično ili stilizovano, oni mogu da se dobiju metodom skalptinga.

1.3.3. Primena pomenutih metoda u video igrama

Tehnika digitalnog vajanja je široko primenjiva u različitim tipovima video igara. Bilo da se radi o realističnom, stilizovanom ili nekom drugom stilu za video igre i njihovo okruženje, ova tehnika modelovanja može da posluži.

Prilikom primene 3D modela u video igrama, nije dovoljno da ti modeli budu izgenerisani sa kvalitetnim

teksturama. Kako bi mogli da se upotrebe u video igrama, potrebno je da se izvrši priprema tih modela, odnosno njihova optimizacija za video igre, što se još naziva i retopologija.

Retopologija je proces pojednostavljenja složene mreže 3D modela kako bi se učinila lakšom za rad. Ovaj proces neophodan je kod modela koji su rezultat digitalnog vajanja i fotogrametrije. Nakon retopologije dobija se model sa smanjenim brojem poligona. Topologija (eng. Topology)² modela je izmenjena, samim tim tekstura originalnog modela nije pogodna za primenu na novom modelu, te je zbog toga neophodno uraditi bejking.

Prva značajna video igra u kojoj su stvarni objekti prikazani upotrebom tehnike fotogrametrije je Nestanak Itana Kartera (eng. The Vanishing of Ethan Carter), koja je nastala 2014 [2]. Grafika i vizuelni prikazi ove igre su pokazali kao veoma imersivni za igrače [3].

Ostale značajne igre koje su takođe koristile tehniku fotogrametrije su: Forza Motorsport, The Talos Principle, Star Wars Battlefront [4]. Od 2014. kada je tehnologija omogućila upotrebu fotogrametrijskih modela u video igrama, napredak na polju dobijanja modela pomoću slika je dosegao još veće visine [5].

1.4. Cilj istraživanja

Cilj ovog istraživanja je utvrditi kod koje od ponuđenih metoda modelovanja, fotogrametrije ili digitalnog vajanja je kvalitet finalnog modela bolji, brzina potrebna za njegovo modelovanje veća kao i optimizovanost 3D modela odgovarajuća određenom okruženju video igre. Podaci do kojih se dođe prilikom istraživanja predstavljaju nedosmišlenu nameru da posluže daljem razvoju fotogrametrije u ovoj industriji. Rad je strukturiran tako da se prvo izlože metode za modelovanje primenom fotogrametrije i digitalnog vajanja, koje unutar sebe uključuju i osvrt na teksturisane i optimizaciju. Nakon toga, dati modeli su ubačeni u scenu i formirano je okruženje koje podseća na prirodne forme i elemente prirode. Na kraju, data je anketa ispitanicima kako bi se utvrdilo da li se može primetiti razlika u datim modelima. Na osnovu ovakve strukture, postavljene su određene hipoteze.

1.5. Hipoteze

Pre nego što se počne za procesom ispitivanja, moguće je postaviti nekoliko hipoteza na osnovu datih informacija.

Hipoteza 1– Fotogrametrijski modeli će se kreirati brže nego oni dobijeni digitalnim vajanjem.

Hipoteza 2– Optimizovanost modela koji se ogleda kroz optimalan broj poligona u modelu, po objektu do 50K poligona, bolje se postiže digitalnim vajanjem.

Hipoteza 3– Nakon sprovedene ankete, mišljenja ispitanika će indikovati da ne postoji velika razlika u kvalitetu između fotogrametrijskog modela i modela dobijenog standardnim načinom modelovanja.

¹ Rendering pipeline- Proces koji grafički sistem treba da izvrši da bi prikazao 3D scenu na 2D ekranu.

² Topology- prikaz i struktura verteksa, ivica i poligona 3D modela

2. METODE

Kao što je već napomenuto, u ovom istraživanju biće primenjena dva načina modelovanja, fotogrametrijsko modelovanje i standardan način modelovanja, digitalno vajanje. Kako bi se na adekvatan način uporedili modeli, biće fotografisani elementi u prirodi, a nakon toga date reference će biti korištene u procesu modelovanja. Nakon što se kreiraju 3D modeli, geometrija će se upotrebiti za implementaciju u video igri. Na kraju, kako bi se utvrdilo da li se dobijeni modeli razlikuju po realističnosti, kreirana je anonimna anketa koja će poslužiti za dobijanje relevantnih rezultata istraživanja.

2.1 Referentno fotografisanje

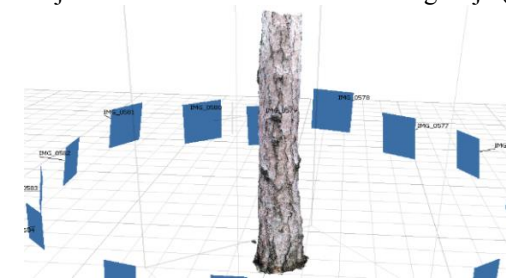
Za oba načina modelovanja bilo je potrebno izaći na teren i uslikati objekte koji će poslužiti kao modeli. Kod standardnog načina modelovanja, izabran je model, čija slika je poslužila kao referenca. Tekstura 3D modela kao i sam oblik rađeni su po uzoru na slike. Svaki fotogrametrijski model slikao se tako što se išlo u krug oko modela, vodeći računa da dolazi do preklapanja fotografija kako bi softver mogao da prepozna iste delove.

2.2. Fotogrametrijsko modelovanje

Nakon što su je objekat fotografisan iz svih uglova i fotografije su spremne, one se zatim učitavaju u softver za rekonstrukciju, AgiSoft Photoscan. Proces dobijanja fotogrametrijskog modela, stoga, podrazumeva generisanje geometrije i teksture, ali takođe i postprodukciju i optimizaciju dobijenog modela u odnosu na broj poligona u mreži. Svaki od delova biće posebno objašnjen u narednim potpoglavljima.

2.2.1. Generisanje geometrije i teksture 3D modela

Ovaj proces sastoji se iz četiri faze, gde prva faza podrazumeva učitavanje svih slika i zatim poravnavanje, imajući na umu da su fotografisane sa barem 80 posto preklapanja duž putanje snimanja i 20 posto upravno na to. Redak oblak tačaka se koristi kako bi softver video koje slike se preklapaju, odnosno imaju dovoljan broj zajedničkih tačaka, a zatim se tokom generisanja gustog oblaka tačaka kreiraju mape dubine za takve parove slika. Zatim, generisanje poligonalne mreže 3D modela na osnovu prethodno kreiranog oblaka tačaka. Pre same instrukcije za kreiranje mreže poligona, potrebno je obrisati višak tačaka koje nisu deo modela. Poslednja faza u procesu rekonstrukcije i generisanja fotogrametrijskog 3D modela je kreiranje teksture na osnovu učitanih fotografija (Sl. 2).



Slika 2. Fotogrametrijski model sa primenjenom teksturom

2.2.2. Postprodukcija

Čak i ako se koriste fotografije visoke rezolucije, mogu se javiti greške u modelu poput delova koji su neuredni i nepotpuni, koji imaju izbočine ili rupe. Da bi se ovo popravilo, koristi se softver Zbrush u kom se otklanjaju nepravilnosti na geometriji modela. Softver Photoshop takođe može da se koristi za zajedničko podešavanje balansa belog i drugih karakteristika fotografija radi jasnoće i tačnije boje na samoj teksturi.

Iako je model unapređen u procesu postprodukcije, i dalje nije spreman za primenu u video igrama, zbog čega treba dodatno da se optimizuje.

2.2.3. Optimizacija

Nakon što je model očišćen, potrebno je da se uradi retopologija. Kako bi se dobijeni 3D model optimizovao, prvobitno je učitao u softver Instant Meshes. Ovaj softver omogućava automatsku retopologiju 3D modela koji sadrže veliki broj poligona i verteksa. Generisani model sastoji se od trouglova koji čine triangulisanu mrežu poligona. U samom softveru, pre početka celog procesa potrebno je da se označi da mreža nakon retopologije bude sačinjena od kvadova. Iz tog razloga, u softver se učitava model bez teksture jer su nakon optimizacije šavovi modela drugačije raspoređeni, a sam izgled mreže je potpuno drugačiji.

2.3. Digitalno vajanje

Proces generisanja modela jednom od standardnih tehnika modelovanja, digitalnim vajanjem zasnivaće se na sličnoj proceduri kao za fotogrametrijsko modelovanje, samo upotrebom drugačijih softvera. Fotografije uslikane prilikom izlaska na teren korišće se kao referentne slike za modelovanje, gde će se zatim kreirati materijali i generisati teksture za izmodelovani 3D model.

2.3.1. Generisanje modela

Samo modelovanje započeto je u softveru Zbrush na osnovu referenci koje su praćene kao ideja. Proces je započet tako što je dodato par kocki koje su spojene, i zatim je uz pomoć četkica započet skalpting. Podešavanja četkice određuju kako će se model deformisati njihovom primenom na površini modela. Postoje različite vrste četkica koje se koriste zavisno od rezultata koji žele da se postignu.

2.3.2. Optimizacija modela

Proces optimizacije i smanjenja broja poligona vrši se u softveru Zbrush uz pomoć decimacije. Decimation Master je alatka koja se koristi za optimizaciju karaktera i modela za okruženje video igara, a takođe pogodan je i za optimizaciju modela za 3D štampu. Osim upotrebe za dobijanje modela sa značajno smanjenim brojem poligona, ova alatka omogućava smanjenje broja poligona i bez gubljenja detalja, kako model ne bi bespotrebno imao suvišne poligone i tako otežavao rad u softveru, a bez primetnih gubitaka na površini modela

2.3.3. Generisanje teksture

Objekti kreirani u fazi modelovanja, primenom digitalnog vajanja su najčešće u podrazumevanoj, sivoj boji što u

Zbrushu simulira materijal gline. Teksturisiranje je upravo proces "obmotavanja" 3D modela 2D slikama, tako da površina modela odgovara konceptu dizajna umetnika ili reference koja se prati, iz stvarnog sveta.

2.4. Generisanje scene u Unreal Engine-u

Nakon što je proces kreiranja 3D modela u obe faze završen, optimizovani modeli su uveženi u Unreal Engine. Glavni osnov za kreiranje scene u Unreal-u bio je da se pokaže kako se uz pomoć dobijenih 3D modela može izgenerisati scena gde se identična geometrija iznova ponavlja i tako kreira scenu koja može biti dugačka i do nekoliko kilometara, uz smanjen utrošak vremena.

2.4. Generisanje ankete

Formirana je anketa kako bi se mogli uporediti objekti dobijeni različitim tehnikama, koji su fizički slični, kao i slične složenosti, te kako bi se moglo doći do zaključka koji model izgleda vizuelno bolji. Anketa koja se koristila u ovom istraživanju bila je anonimna. Osmišljena je i prilagođena potrebama za dobijanje relevantnih saznanja koji način modelovanja daje vizuelno bolje rezultate u odnosu na status ispitanika, na osnovu oblasti kojom se bave i zavisno od toga da li su upoznati sa fotogrametrijom i njenom primenom.

3. REZULTATI

Nakon što je završeno modelovanje na oba načina, slični modeli su upoređeni po tri kriterijuma: izgledu modela, brzini dobijanja modela i njihovoj optimizovanosti za video igre.

3.1. Prikaz svih izgenerisanih 3D modela i scene

Upotrebom svih generisanih 3D, kako fotogrametrijskih tako i modela dobijenih digitalnim vajanjem napravljena je scena koja je namenjena za koncept Endless Runner video igre. Konačan rezultat prikazan je na slici 3.



Slika 4. Izgenerisana scena u softveru Unreal Engine

3.2. Rezultati istraživanja na osnovu ankete

Kada se napravi analiza rezultata dobijenih anketiranjem na osnovu postavljenih pitanja, može se doći do zaključka da su se za svaki upoređen primer fotogrametrijskog modela i modela dobijenog digitalnim vajanjem ispitanici odlučili za fotogrametrijski model. Bolju ocenu u sva tri slučaja su dale žene. Što se tiče starosne dobi, najviše ispitanika je bilo između 20 i 35 godina, dok po pitanju oblasti kojom se ispitanici bave, bolju ocenu fotogrametrijskom modelu su dali ljudi iz oblasti dizajna i arhitekture, dok je nešto manji broj bilo programera.

3.3. Poređenje brzine generisanja modela i njihove optimizovanosti

Kada se napravi poređenje koliko je vremenski trebalo da se kompletno izgeneriše 3D model koji je spreman za video igru upotrebom fotogrametrije i digitalnog vajanja, može se zaključiti da je za fotogrametrijski model bilo potrebno manje vremena. Kada su u pitanju jednostavniji modeli, ova razlika ne igra toliku ulogu, ali svakako da bi kod kompleksnijih scena dolazilo do mnogo veće razlike u vremenu, samim tim fotogrametrija bi značajno ubrzavala proces dobijanja visokokvalitetnih i tačnih modela.

4. ZAKLJUČAK

Prevladavajući cilj ovog istraživanja bio je da se napravi analiza uporednog pristupa generisanju geometrije okruženja, primenom fotogrametrije i digitalnog vajanja. Analizirajući postavljene hipoteze, može se konstatovati da je potvrđena prva istraživačka hipoteza kojom se pretpostavljalo da će se brže dobiti fotogrametrijski model u odnosu na standardan, uzimajući u obzir i vreme potrebno za odlazak na teren.

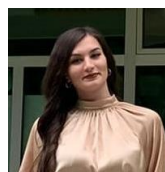
Što se tiče druge hipoteze, može se zaključiti da ona nije potvrđena iz razloga što se tokom obrade ove teme došlo do zaključka da su dobijeni podjednaki rezultati optimizovanosti modela primenom obe tehnike.

Trećom hipotezom pretpostavljeno je da će se ispitanici izjasniti tako da ne postoje velike razlike u kvalitetu 3D modela dobijenih fotogrametrijskim modelovanjem i modelovanjem na standardan način. Međutim, ova hipoteza se nije potvrdila jer je istraživanjem pokazano da postoji velika razlika u kvalitetu 3D modela dobijenih ovim tehnikama.

5. LITERATURA

- [1] Joel R., 2019. Photogrammetry for 3D Content Development
- [2] Statham N./Jacob J., 2019. Photogrammetry for Game Environments 2014-2019: The Vanishing of Ethan Carter
- [3] Terzioglu Y, 2015. Immersion and identity in video games,
- [4] Taylor L., 2002. Video games: perspective, point of view and immersion, University of Florida
- [5] Michailidis L./Balaguer-Ballester E./ He X., 2018. Flow and Immersion in Video Games: The Aftermath of a Conceptual Challenge

Kratka biografija:



Aleksandra Bobić rođena je u Novom Sadu 1998. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Računarske grafike pod nazivom „Uticaj različitih tipova nameštaja u enterijeru na celokupan utisak rendera“ odbranila je 2021.god. kontakt:bobic.aleksandra@gmail.com

ENERGETSKA SANACIJA U CILJU POVEĆANJA ENERGETSKE EFIKASNOSTI NA PRIMERU STAMBENOG OBJEKTA U INDIJI**ENERGY REHABILITATION OF A RESIDENTIAL BUILDING IN INDIJA WITH THE AIM OF INCREASING ENERGY EFFICIENCY**

Zlata Marijanović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ENERGETSKE TEHNOLOGIJE

Kratak sadržaj – Rad se sastoji iz dva dela, teorijskog i praktičnog. U teorijskom delu, obrađena je zakonska regulativa iz oblasti energetske efikasnosti u EU i Republici Srbiji, i mere za povećanje energetske efikasnosti, uključujući toplotnu izolaciju, zamenu stolarije, korišćenje energetske efikasne rasvete i obnovljivih izvora energije u stambenim zgradama. U praktičnom delu u softveru URSA urađen je proračun energetske efikasnosti za postojeće stanje stambenog objekta u Indiji, izvršena tehnno-ekonomska analiza različitih alternativa energetske sanacije, i po izboru i usvajanju najoptimalnije alternative, izvršena analiza zamene konvencionalnog sistema grejanja toplotnom pumpom.

Ključne reči: Energetska efikasnost, energetska sanacija, građevinska fizika, termički omotač

Abstract – The paper consists of two parts, theoretical and practical. In the theoretical part, legal regulations in the field of energy efficiency in the EU and the Republic of Serbia are analyzed, as well as measures with the aim of increasing energy efficiency, including thermal insulation, replacement of carpentry, use of energy-efficient lighting and renewable energy sources in residential buildings. In the practical part, using the URSA software, an energy efficiency calculation was made for the existing state of the residential building in India, as well as a techno-economic analysis of various energy rehabilitation alternatives. After the selection and adoption of the most optimal alternative, an analysis of the replacement of the conventional heating system with a heat pump was performed.

Keywords: Energy efficiency, energy rehabilitation, building physics, thermal envelope

1. UVOD

Poslednjih godina sve je aktuelnija tema o potrebi za globalnom energetsom tranzicijom, kako zbog sve progresivnijih klimatskih promena, tako i zbog svetske ekonomske krize koja je istakla važnost koncepta održivog razvoja i energetske efikasnosti. Veliki deo postojećih ekoloških problema proističe iz potrošnje energije koju podstiču zgrade i aktivnosti u njima [1].

NAPOMENA

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

U Evropskoj uniji, sektor zgradarstva čini više od 40% ukupne potrošnje energije, te predstavlja područje sa najvećim potencijalom za intervencije i što brže delovanje [2]. Smanjenje potražnje za energijom u ovom sektoru važan je cilj, a glavnu ulogu u njegovom ostvarenju upravo ima energetska efikasnost, koja je prepoznata kao “novi energetska izvor” koji bi bio rešenje za povećanje sigurnosti snabdevanja energijom i smanjenje uvozne zavisnosti.

2. ZAKONSKA REGULATIVA**2.1 Zakonska regulativa u Evropskoj Uniji**

Kao sredstvo za rešavanje problema klimatskih promena, energetska efikasnost u zgradarstvu se sve više stavlja u fokus u politici zaštite životne sredine EU. Usvajanjem brojnih Direktiva postavljeni su minimalni standardi za sve zemlje članice u smislu unapređenja energetske efikasnosti u stambenim zgradama.

Direktiva 2002/91/ES predviđa nekoliko različitih mera za postizanje racionalnog korišćenja energetske izvora a to su: zajednička metodologija za proračun energetske karakteristika zgrada, primena minimalnih zahteva energetske efikasnosti za nove zgrade, primena minimalnih zahteva energetske efikasnosti za postojeće zgrade prilikom većih rekonstrukcija, energetska sertifikacija zgrada i redovne inspekcije kotlova i sistema za kondicioniranje vazduha u zgradama [3].

Direktiva 2010/31/EU predstavlja prekretnicu EU u oblasti smanjenja potrošnje energije u zgradarstvu, a propisuje da oglašava za iznajmljivanje i prodaju zgrada treba da uključuju energetske sertifikate o performansama zgrade, da je u državama EU potrebno uspostaviti univerzalnu šemu za inspekciju sistema grejanja i klimatizacije, kao i da sve nove javne i privatne zgrade od 2019. treba da budu zgrade sa približno nultim emisijama.

Direktiva 2012/27/EU - propisuje da su članice EU u obavezi da izvrše energetske efikasno renovirane najmanje 3% zgrada koje su u vlasništvu i zaposednute su od strane centralne Vlade, da bi Vlade zemalja članica EU trebalo da kupuju samo zgrade koje su visokog energetske razreda, kao i da države EU moraju izraditi dugoročne nacionalne strategije obnove.

Direktiva 2018/844/EU – dokument koji je proizašao iz revizije Direktive o energetske efikasnosti i Direktive o energetskim performansama zgrada i njihovog usklađivanja sa novim Energetskim i klimatskim ciljevima za 2030.

2.2 Zakonska regulativa u Republici Srbiji

Unapređenje energetske efikasnosti i implementacija obnovljivih izvora energije prepoznati su kao ključni elementi energetske politike radi obezbeđenja energetske bezbednosti naše države i održivog razvoja nacionalne energetike, i iz tog razloga usvojena su 4 Zakona:

Zakon o efikasnom korišćenju energije čiji je glavni cilj da se obezbedi i stimuliše racionalno i održivo korišćenje energije uspostavljanjem tržišta za usluge koje doprinose povećanju energetske efikasnosti [4].

Zakon o planiranju i izgradnji koji predstavlja instrument čijim posredstvom su prenete odredbe evropskih Direktiva iz oblasti racionalne upotrebe energije u naše zakonodavstvo i čijim su usvajanjem doneti novi Pravilnici o energetske efikasnosti zgrada i o uslovima, sadržini i načinu izdavanja sertifikata o energetskim svojstvima zgrada.

Zakon o energetske efikasnosti i racionalnoj upotrebi energije uređuje uslove i način efikasnog korišćenja energenata i energije, propisuje mere politike energetske efikasnosti i reguliše podsticajne i druge mere u ovoj oblasti, sa ciljem ostvarivanja ušteda energije i povećanja sigurnosti snabdevanja energijom [5].

Zakon o korišćenju obnovljivih izvora energije Usvajanjem ovog Zakona, teži se ka tome da u Republici Srbiji do 2040. godine iz obnovljivih izvora dolazi najmanje 40 % energije.

3. MERE ZA POVEĆANJE ENERGETSKE EFIKASNOSTI U ZGRADARSTVU

U cilju poboljšanja energetskog razreda određenog objekta prilikom njegove sanacije, od velikog je značaja da se na ispravan način pristupi optimizaciji termotehničkih sistema u objektu kroz primenu različitih mera.

Na samom početku, potrebno je razmotriti grupu mera za poboljšanje karakteristika same zgrade, zatim analizirati mere za smanjenje toplotnih gubitaka pri proizvodnji i distribuciji toplote, a tek u finalnoj etapi analize razmotriti mere koje podrazumevaju zamenu opreme i uređaja za grejanje, klimatizaciju, osvetljenje itd. [6].

Toplotna izolacija je materijal ili kombinacija materijala koji usporavaju brzinu toplotnog toka kondukcijom, konvekcijom i zračenjem, što usporava protok toplote u ili iz zgrade zbog visokog toplotnog otpora. Ona daje glavni doprinos i predstavlja logičan prvi korak ka postizanju veće energetske efikasnosti.



Slika 1. Funkcija toplotne izolacije objekta [7]

Najpopularniji proizvodi za toplotnu izolaciju jesu veštački materijali. Proizvodi od mineralne vune čine oko 50-55%, a plastične pene oko 40-45% ukupne proizvodnje [8]. Pored njih takođe postoje i oni prirodnog porekla.

- **Prirodni termoizolacioni materijali** – Najzastupljeniji prirodni termoizolacioni materijali jesu ilovača, pluta, konoplja, lan, slama itd. Karakterišu ih dobre građevinske karakteristike kao što su regulisanje i zadržavanje vlage u prostoru i dobra akumulacija toplote, a takođe omogućavaju dobru zvučnu izolaciju u unutrašnjim prostorima i sprečavaju pojavu statičkog elektriciteta. Toplotna provodljivost iznosi oko 0,045 W/mK.

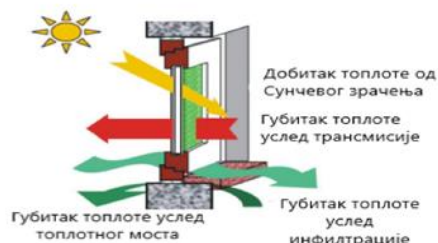


Slika 2. Blokovi od industrijske konoplje za termoizolaciju objekata [9]

- **Sintetički termoizolacioni materijali na bazi plastičnih masa** - Najčešće korišćeni materijali iz ove grupe su ekspandirani polistiren (stiropor) i ekstrudirani polistiren (stirodur). Toplotna provodljivost ovih materijala se kreće u opsegu između 0,035 i 0,040 W/mK.
- **Termoizolacioni materijali mineralnog porekla** - U ovu grupu svrstavaju se staklena i kamena vuna. Staklena vuna sadrži kvarcni pesak, krečnjak, dolomit i 50-60% recikliranog stakla, a njena toplotna provodljivost iznosi 0,032-0,045 W/mK. Kamena vuna se proizvodi topljenjem krečnjaka i balzata, a njena toplotna provodljivost iznosi 0,035-0,04 W/mK [10].
- **Termoizolacioni betoni i malteri** - Malteri i betoni su pre svega konstrukcijski materijali, međutim oni poseduju i termoizolaciona svojstva. Laki izolacioni beton sa polistirenskim perlama ili ćelijasti laki beton pokazuju toplotnu provodljivost od svega 0,07 do 0,33 W/mK [11].

Zamena prozora i vrata termički kvalitetnijim

Ukupni gubici kroz stolariju (Slika 3) iznose oko 50% toplotnih gubitaka zgrade, odakle proističe velika važnost njihovog pravilnog odabira prilikom energetske sanacije objekata. Neophodan je izbor stolarije sa što boljim termičkim performansama, odnosno što nižim koeficijentom prolaza toplote.



Slika 3. Toplotni dobici i gubici kroz stolariju [12]

Koeficijent prolaza toplote kod drvenih prozora zavisi od debljine prozorskog profila i vrste drveta, a kreće se u opsegu od 1,1 do 1,8 W/m² K, kod plastične stolarije

kreće se između 0,9 i 1,4 W/m² K, a kod aluminijumskih profila u rasponu od 1,5 do 3,2 W/m²K. Najoptimalnije rešenje jeste drvo-aluminijumska stolarija koja predstavlja spoj velike otpornosti aluminijuma i estetike i odlične toplotne izolacije drveta.

Kombinacijom ovih materijala postiže se odlična toplotna i zvučna izolacija, dobra stabilnost i izuzetno dug vek trajanja. Koeficijent prolaza topline ovog tipa stolarije iznosi 0,7 W/m²K.

Upotreba energetske efikasne rasvete - Osvetljenje je odgovorno za približno 15% utrošene električne energije u domaćinstvima, zbog čega je sve veće interesovanje ljudi za energetske efikasne rasvete i sistemima za upravljanje osvetljenjem, čija je primarna uloga racionalizacija potrošnje električne energije. Poslednjih godina masovno se promovise LED rasveta koju karakteriše mala snaga, usmereni svetlosni snopovi i mogućnost promene intenziteta i boje svetlosti, vek trajanja od oko 50000 radnih sati, te ušteda u potrošnji električne energije do 88%.

Primenjena obnovljivi izvori energije i nekonvencionalnih sistema grejanja u stambenim objektima - Kako sistemi grejanja značajno doprinose emisiji CO₂, naophodna su značajna unapređenja na polju energetske efikasnosti u toj sferi, koja se mogu postići primenom efikasnih nekonvencionalnih tehnologija, kao što su različiti sistemi sa toplotnim pumpama, solarne termalne tehnologije i visokoefikasni kotlovi koji kao gorivo koriste biomasu, Prema podacima iz 2015. godine, obnovljivi izvori energije u Evropi predstavljaju rastući udeo u ukupnoj potražnji za grejanje stambenih objekata.

4. ENERGETSKA SANACIJA STAMBENOG OBJEKTA U INDIJI

4.1 Tehnički opis

Stambeni objekat koji je tema ovog rada jeste porodična kuća koja se nalazi u Indiji. Prizemlje i potkrovlje su predviđeni kao dve zasebne stambene jedinice. Za vertikalnu komunikaciju između prizemlja i potkrovlja predviđeno je spoljno stepenište.

Dimenzije objekta u osnovi su 8,88 x 10,26 m, a neto površina objekta unutar termičkog omotača iznosi 142,08 m².

Temeljni zidovi objekta izvedeni su od opeke u produženom malteru, a ukupna debljina konstrukcije iznosi 42 cm. Konstruktivni zidovi i pregrade izvedeni su od opeke u krečnom malteru, debljine 25 cm. Tavanica je izolovana polistirenskim pločama debljine 10 cm. Objekat se greje na gasni kotao stepena korisnosti 0,8, instalisane snage 24 kW.

4.2 Građevinska fizika i alternative za sanaciju

Uz pomoć softverskog rešenja URSA Građevinska fizika za postojeće stanje objekta izrađen je Elaborat energetske efikasnosti.

Proračunom sumiranom u Tabeli 1. je zaključeno da postojeći objekat pripada energetskom razredu G i da ne zadovoljava energetske zahteve za postojeće objekte prema Pravilniku o energetskoj efikasnosti zgrada iz 2011. godine.

Tabela 1. Godišnji proračun potrebne energije za grejanje objekta

Koeficijent transmisivnog gubitka topline	440,15 W/K
Koeficijent ventilacionog gubitka topline	135,08 W/K
God. potrebna en. za nadoknadu gubitaka topline	34458,72 kWh
Godišnja količina energije koja potiče od unutrašnjih dobitaka topline	1740,11 kWh
Godišnja količina en. koja potiče od dobitaka usled sunčevog zračenja	4458,76 kWh
Godišnja potrebna energija za grejanje	28259,8 kWh
Specifična godišnja potrebna energija za grejanje	198,9 kWh/m ²
Energetski razred zgrade	G

Kao što se može zaključiti na osnovu rezultata proračuna, razmatrani objekat u postojećem stanju karakteriše neracionalno velika potrošnja energije za grejanje.

U cilju povećanja energetske efikasnosti objekta, analizirani su tehnički i finansijski benefiti 6 alternativa za energetske sanaciju, čiji je prikaz dat u Tabeli 2.

Tabela 2. Alternative za energetske sanaciju

Alternativa	Mera unapređenja en. efikasnosti
A1	Zamena postojeće stolarije sa PVC trokomornim prozorima i vratima, dvostruko zastakljenim, ispunjena vazduhom Uw= 1,8W/m ² K
A2	Zamena postojeće stolarije sa PVC petokomornim prozorima i vratima, dvostruko niskoemisiono staklo, ispunjena argonom Uw= 1,3 W/m ² K
A3	Izolacija sp. zidova polistirenskim pločama 5 cm λ=0,041 W/mK
A4	Izolacija sp. zidova polistirenskim pločama 10 cm λ=0,041 W/mK
A5	Izolacija sp. zidova polistirenskim pločama 10 cm + zamena postojeće stolarije PVC petokomornim prozorima, dvostruko niskoemisiono staklo, ispunjena argonom Uw= 1,3 W/m ² K
A6	Izolacija sp. zidova polistirenskim pločama 10 cm+ zamena postojeće stolarije PVC trokomornim prozorima, dvostruko zastakljeni, ispunjena vazduhom Uw= 1,8 W/m ² K

Rezultati proračuna troška energetske sanacije analiziranog stambenog objekta po alternativama, prikazani su grafički na Slici 4, dok su na Slici 5 prikazani rezultati tehničke analize, kojom je utvrđeno da bi primenom Alternative 5 bila ostvarena najveća ušteda godišnje specifične energije za grejanje, te da bi se energetskom sanacijom dostigao D energetski razred objekta.



Slika 4. Trošak energetske sanacije po alternativama



Slika 5. Specifična en. potrebna za grejanje prilikom primene različitih alternativa en. sanacije

Ipak, vrednost investicionih troškova energetske sanacije, kao ni specifična energija potrebna za grejanje objekta, ponaosob nisu pogodni indikatori u postupku izbora najoptimalnije alternative. Iz tog razloga, uveden je kriterijum koji u sebi inkorporira oba pokazatelja. Na taj način moguće je odrediti koeficijent energetske sanacije koji bi predstavljao relativni pokazatelj uštede energije po svakoj uloženoj novčanoj jedinici. Koeficijent energetske sanacije moguće je odrediti uz pomoć jednačine:

$$K_{EC} = \frac{Q_1 - Q_n}{T_n} \quad (1)$$

Gde je:

K_{EC} [kWh/(€ god)] - koeficijent energetske sanacije

Q_1 [kWh/god] - godišnje potrebna en. za grejanje objekta za A1

Q_n [kWh/god] - godišnje potrebna en. za grejanje objekta za i-tu alternativu

T_n [€] - ukupni trošak en. sanacije za i-tu alternativu

Najveći koeficijent energetske sanacije, tj. najpozitivniji efekat odnosa ostvarene uštede godišnje potrebne energije i uložених sredstava za energetsку санацију, karakteriše alternativu A3, te je izolacija spoljašnjih zidova polistirenskim pločama 5 cm najoptimalnije rešenje za energetsку санацију породичне куће u Indiji.

5. ANALIZA ZAMENE KOTLOVSKE GREJNE INSTALACIJE TOPLOTNOM PUMPOM

Proračunom u softveru GeoT*SOL utvrđeno je da bi postojeći sistem za grejanje, nakon usvajanja alternative A3 i smanjenja godišnje potrebne toplote za grejanje na 16807 kWh, mogao biti zamenjen toplotnom pumpom tipa zemlja-voda model TERRA SW Twin 35, kapaciteta grejanja 35,25 kW i COP 4,96. Tako bi se ostvarilo smanjenje cene toplotne energije i ona bi bila oko 1,5 puta manja u odnosu na slučaj pre termičke izolacije objekta i ugradnje savremenog sistema za grejanje. Pored toga, energetska sanacija bi kao benefit imala i smanjenje emisija CO₂ od 3,7 t/god.

Tabela 3. Troškovi sistema grejanja

	Investicioni troškovi	Eksplatacioni troškovi
Gasni kotao	1595 €	1387 €
Toplotna pumpa	9121 €	401 €

Uzimajući u obzir troškove nabavke izolacionog materijala, toplotne pumpe i potrebne opreme, njen

transport i dovođenje sistema u optimalne uslove rada i cenu energije i ljudskog rada sumirane u Tabeli 3, prost period otplate investicije iznosio bi 7,6 godina, nakon čega bi počelo ostvarivanje finansijske dobiti.

6. ZAKLJUČAK

Iz proračuna sprovedene integralne analize, zaključuje se da je najoptimalnija od razmatranih alternativa za energetsку санацију стамбеног објекта termička izolacija spoljašnjih zidova polistirenskim pločama debljine 5cm, jer se tako uz minimalna ulaganja ostvaruje najveće smanjenje godišnje potrebne energije za grejanje. Analiza zamene postojećeg sistema grejanja toplotnom pumpom rezultirala je zaključkom da takva investicija ima kako finansijske, tako i ekološke benefite, te da je ovaj projekat u potpunosti tehnо-економski оправдан.

7. LITERATURA

- [1] D. Ürge-Vorsatz, L. F. Cabeza, S. Serrano, "Heating and cooling energy trends and drivers in buildings", Renewable and Sustainable Energy Reviews, 2015., 41, 85-98
- [2] P. L. Simona, P. Spiru, I. V. Ion, "Increasing the energy efficiency of buildings by thermal insulation", ScienceDirect, 2017., 128, 393-399
- [3] I. C. Korkut, "Building Energy Performance, Regulations and Certificates", Int'l Journal of Advances in Agricultural & Environmental Engg., 2016., 3, 175-180
- [4] A. Solujić, "Zakon o efikasnom korišćenju energije-obaveze i podsticaji za unapređenje energetske efikasnosti", Polis, 2013., 6, 4-9
- [5] Zakon o energetskeј efikasnosti i racionalnoj upotrebi energije ("Sl. glasnik RS", br. 40/2021)
- [6] M. Banjac, M. Todorović, F. Kulić, B. Živković, "Priručnik za energetske menadžere za oblast energetike zgrada", UNDP, Beograd, 2017.
- [7] <https://www.isowall.co.za/the-main-advantages-of-thermal-insulation/> Pristupljeno: 05.12.2021
- [8] D. Bozsaky, "The historical development of thermal insulation materials", Periodica polytechnica, 2010., 41, 49-56
- [9] <https://drvotehnika.info/clanci/konoplja-u-gradjevinarstvu> Pristupljeno: 05.12.2021.
- [10] R. T. Bynum, "Insulation Handbook", The McGraw-Hill Companies, New York, 2001.
- [11] N. F. Shahedan, M. M. Abdullah, N. Mahmed, "Review on thermal insulation performance in various type of concrete", Advanced Materials Engineering and Technology, 2017.
- [12] S. Kojić, "Procena stanja, energetska sanacija i revitalizacija glavne zgrade Kotekovor dvorca u Futogu", Zbornik radova FTN, 2020., 7, 1207-1210

Kratka biografija:



Zlata Marijanović rođena je u Rumi 1997. godine. Master rad na Fakultetu tehničkih nauka iz oblasti Energetskih tehnologija je odbranila 2022. godine.
Kontakt:
zlata.marijanovic@gmail.com

U realizaciji Zbornika radova Fakulteta tehničkih nauka u toku 2022. godine učestvovali su sledeći recenzenti:

Aco Antić	Dragana	Marinko Maslarić	Nemanja Sremčev
Aleksandar	Konstantinović	Marko Lazić	Nemanja Tasić
Anđelković	Dragana Šarac	Marko Marković	Nenad Grahovac
Aleksandar Kovačević	Dragoljub Šević	Marko Todorov	Nenad Simeunović
Aleksandar	Drago Žarković	Marko Vekić	Nikola Vojnović
Kupusinac	Duško Bekut	Maša Bukurov	Petar Mirković
Aleksandar Ristić	Đorđe Vukelić	Matija Stipić	Platon Sovilj
Aleksandar Selakov	Goran Jeftenić	Mijodrag Milošević	Radivoje Dinulović
Aleksandra Radulović	Goran Savić	Milan Delić	Radomir Kojić
Aleksandra Pešterac	Goran Sladić	Milan Gavrić	Ratko Obradović
Andraš Anderla	Goran Švenda	Milan Marinković	Sandra Dedijer
Andrija Rašeta	Goran Tepić	Milan Mirković	Saša Medić
Atila Zelić	Gordan Stojić	Milan Rapajić	Slavica Mitrović
Bojan Batinić	Gordana Ostojić	Milan Rackov	Senka Bajić
Bojan Matić	Igor Dejanović	Milan Trivunić	Slobodan Morača
Bojan Tepavčević	Igor Peško	Milan Vidaković	Slobodan Šupić
Borislav Savković	Iva Šiđanin	Milena Krklješ	Srđan Popov
Branislav Atlagić	Ivana Mihajlović	Milica Kostreš	Srđan Vukmirović
Branislav Stevanov	Igor Maraš	Milica Miličić	Stevan Gostojić
Branka Nakomčić	Ivan Prokić	Miloš Simić	Stevan Grabić
Branko Milosavljević	Ivana Katić	Milovan Lazarević	Stevan Milisavljević
Branko Škorić	Ivana Maraš	Milja Simeunović	Stevan Stankovski
Damir Đaković	Ivana Miškeljin	Miodrag Milutinov	Strahil Gušavac
Danijela Ćirić	Jasmina Dražić	Miodrag Žigić	Svetlana Bačkalić
Danijela Gračanin	Jelena Atanacković	Mirjana Malešev	Svetlana Nikoličić
Danijela Lalić	Jeličić	Miroslav Zarić	Tamara Čeranić
Darko Čapko	Jelena Borocki	Mirko Borisov	Veran Vasić
Darko Reba	Jelena Demko Rihter	Mirko Raković	Vesna Stojaković
Darko Stefanović	Jelena Ivetić	Miro Govedarica	Višnja Žugić
Dejan Ecet	Jelena Radonić	Miroslav Kljajić	Vladimir Ilić
Dejan Lukić	Jelena Slivka	Miroslav Popović	Vladimir Katić
Dejan Reljić	Jelena Spajić	Miroslav Zarić	Vladimir Mučenski
Dejan Jerkan	Kalman Babković	Mitar Jcanović	Vlastimir Radonjanin
Dejan Movrin	Lazar Kovačević	Mladen Tomić	Vuk Bogdanović
Dejan Ubavin	Lidija Krstanović	Mladen Radišić	Vuk Vranjkovic
Dejana Nedučin	Ljiljana Popović	Nataša Samardžić	Zdravko Tešić
Dragan Ivanović	Ljubica Duđak	Nebojša Brkljač	Zoran Čepić
Dragan Ivetić	Magdolna Pal	Nebojša Radović	Zoran Jeličić
Dragan Jovanović	Maja Turk Sekulić	Nebojša Ralević	Zoran Papić
Dragan Pejić	Maja Petrović	Neda Milić Keresteš	Željen Trpovski
Dragan Ružić	Marija Silađi	Nemanja Kašiković	

