УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА

ЗБОРНИК РАДОВА
ФАКУЛТЕТА ТЕХНИЧКИХ НАУКА

Едиција: Техничке науке - зборници
Година: XXXVII
Број: 2/2022

Нови Сад

Едиција: „Техничке науке – Зборници“
Година: XXXVII Свеска: 2

Издавач: Факултет техничких наука Нови Сад
Главни и одговорни уредник: проф. др Срђан Колаковић, декан Факултета техничких Наука у Новом Саду

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 друга 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 web 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 29.09.2021. до 26.10.2021. год., а који се промовишу 27.01.2022. 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 2. објављени су радови из области:

- електротехнике и рачунарства.

У свесци са редним бројем 1. објављени су радови из области:

- машинства,
- грађевинарства,
- саобраћаја,
- графичког инжењерства и дизајна и
- архитектуре.

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

- инжењерског менаџмента,
- инжењерства заштите на раду и заштите животне средине,
- мехатронике,
- геодезије и геоматике,
- регионалне политике и развоја,
- управљања ризиком од катастрофалних догађаја и пожара,
- инжењерства информационих система,
- сценске архитектуре и дизајна,
- биомедицинског инжењерства,
- анимације у инжењерству,
- информационог инжењеринга и
- чистих енергетских технологија.

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

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

„Високо место у друштву најбољих“

Уредништво

SADRŽAJ

STRANA

Radovi iz oblasti: Elektrotehnika i računarstvo

STRANA

STRANA

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.41
DOI: https://doi.org/10.24867/16BE01Tomic

РАЧУНАРСКА ОБРАДА И АНАЛИЗА ПОДАТАКА У ИСТРАЖИВАЊУ ОРАЛНОГ ЗДРАВЉА

COMPUTER-BASED PROCESSING AND ANALYSIS OF DATA IN ORAL HEALTH RESEARCH

Мирослав Томић, *Факултет техничких наука, Нови Сад*

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – *У овом раду описано је рачунарско испитивање података у вези са оралним здрављем код трудница. Описане су статистичке и истражне анализе података, након чега је дат опис чишћења и трансформације података. Описани су употребљавани предикциони модели и које су припреме извршене у циљу њиховог покретања. Приказани су добијени резултати и предочено је њихово могуће тумачење.*

Кључне ријечи: *Орално здравље, Орално здравље трудница, Анализа података, Предикциони модели*

Abstract – *This paper describes computer-based investigation of data in relation with oral health in pregnant women. Statistical and exploratory data analyses are described, followed by a description of data cleansing and transformation. The predictive models used are described, together with preparations that were made in order to run them. The obtained results are presented, as well as their possible interpretation.*

Keywords: *Oral health, Oral health of pregnant women, Data analysis, Prediction models*

1. УВОД

Дуго времена опште здравље човјека није довођено у везу са оралним здрављем због недостатка информација о њиховој вези. Орално здравље утиче на опште здравље изазивајући знатне болове и патњу, мијењајући исхрану људи, њихов говор и њихов квалитет живота [1]. Орално здравље такође утиче и на друге хроничне болести [2]. Оно такође утиче на физичко и психичко стање људи и на то како расту, уживају у животу, говоре, жваћу, пробају храну и друже се [3]. Озбиљан каријес умањује квалитет живота дјеце која доживљавају бол, нелагодност, унакаженост, акутне и хроничне инфекције [1].

За опште здравље жене која је била трудна и здравље њеног новорођенчета, важно је и питање оралног здравља [4,5]. Трудноћа је природни процес праћен значајним физиолошким и хормонским промјенама у женском тијелу, укључујући и усну шупљину [6]. Неки од честих оралних проблема у трудноћи су труднички гингивитис, бенигне гингивалне лезије, покретљивост зуба, зубни каријес и пародонтитис [7]. Ови проблеми могу довести до губитка зуба. До сада су сагледани различити фактори који могу утицати на здравље труднице, али није прецизно утврђено који су то фактори који директно утичу.

Колико је превенција важна говори и податак да је Свјетска здравствена организација креирала глобалну стратегију за превенцију и контролу незаразних болести која уводи нову стратегију за превенцију и контролу оралних болести [2]. Нажалост у нашој околини програми за превенцију се не промовишу довољно и не уређују редовно. Актуелна уредба о националном програму превентивне стоматолошке здравствене заштите [8] датира из 2009. године, а програм који му је претходио, програм превентивне стоматолошке заштите становника Србије [9], датира из 1996. године.

Циљ овог рада је предикција недостатка зуба код трудница помоћу предиктивних модела. За достизање овог циља примјењена је истражна и статистичка анализа над подацима који су прикупљени путем анкетних упитника. Након тога примјењени су предиктивни модели: логистичка регресија (енгл. *Logistic Regression*) [10], стабла одлучивања (енгл. *Decision Trees*) [10] и случајна шума (енгл. *Random Forest*) [10].

Опис ранијих истраживања је у 2. поглављу. Описи скупа података и његовог чишћења, трансформације и анализе су у 3. и 4. поглављу, а опис припреме предикционих модела у 5. поглављу. Резултати и дискусија су у 6. поглављу, а закључак у 7. поглављу.

2. ПРЕГЛЕД ПРЕТХОДНИХ ИСТРАЖИВАЊА

Значајан број радова посвећује пажњу трудничким навикама и њиховој анализи. У студијама је истицано да постоји велика преваленција у нарушеном оралном здрављу код трудница и да је стање забрињавајуће [11-17]. Погрешна увјерења о заштити зуба током трудноће, велики број покварених зуба, проблеми са каријесом, лоша орална хигијена, депресија, старост и социоекономски показатељи су неки од фактора који су пронађени у литератури као утицајни [12-17].

Разне студије су показале да је потребно повећати промоцију оралног здравља [4,14,18-21]. Савјетује се подизање свијести трудница о важности оралног здравља као и важности одласка на преглед код стоматолога.

НАПОМЕНА:
Овај рад је проистекао из мастер рада чији ментор је био доц. др Владимир Иванчевић.

У прегледу литературе могле су се уочити ствари које су заједничке за већину радова, а то су: социоекономски фактори, орална хигијена и недовољна превенција одласцима на преглед код стоматолога. У већини радова се истиче и да је лоше орално здравље мајке повезано са новорођенчетом. Промоција оралног здравља и подизање свијести трудница и пружалаца стоматолошких услуга се уочавају као најбоља средства за борбу против лошег оралног здравља.

3. ОПИС СКУПА ПОДАТАКА

Скуп података је настао у оквиру посебног претходног истраживања на основу анкетног упитника који је креиран за потребе унапређења превенције каријеса раног детињства (КРД) у АП Војводини. За спровођење овог истраживања је издато одобрење Етичког одбора Института за јавно здравље Војводине. Узорачки оквир су чиниле све породиље које су се породиле у периоду 1. јун – 30. новембар 2015. године.

Питања из анкетног упитника представљају обиљежја у скупу података. На нека питања је било могуће дати више одговора па су она дијељена на више обиљежја приликом уноса у скуп података. Анкетни упитник се састојао од 35 питања, а изворни скуп података од 43 обиљежја. Изворни скуп података је био сачуван у формату *xlsx* и садржао је укупно 4310 редова. Сва обиљежја, сем обиљежја везаног за старост које је нумеричког типа, категоричког су типа. Приликом анкетирања ниједна испитаница није дала одговор на свако питање. У упитнику је постојала варијанта у којој се питања прескачу у зависности од одговора. Поред овога постоји варијанта уноса података и за близанце. Овакве варијанте су могле условити појаву недостајућих вриједности у подацима.

У разговору са доменским експертом закључено је да нису потребна сва обиљежја из изворног скупа података јер нису сва релевантна за посматрани проблем. Такође закључено је да се три питања која су у вези са пушењем могу трансформисати у једно. Овим поступком добијен је подскуп од 22 обиљежја.

4. ЧИШЋЕЊЕ, ТРАНСФОРМАЦИЈА И АНАЛИЗА СКУПА ПОДАТАКА

За потребе имплементације коришћен је програмски језик *Python v3.7* у оквиру окружења *JupyterLab v3.0.16*. Коришћене су и додатне библиотеке, *Pandas v1.2.4* и *NumPy v1.19.2* за манипулацију над подацима. За визуелизацију података су употребљаване библиотеке *Matplotlib v3.4.2* и *Seaborn v0.11.1*.

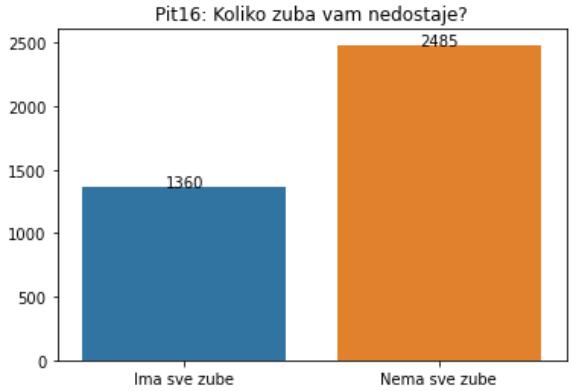
4.1. Чишћење и трансформација података

Све вриједности одговора које су погрешно унијете било је потребно замијенити недостајућим вриједностима. Подаци који представљају одговор на неко питање су заправо редни број одговора и морају бити у одређеном опсегу. Било је могуће прескакање неких питања у зависности од одговора. Требало је обезбједити да се на тим мјестима нађу недостајуће вриједности. Обиљежја која су говорила да ли је трудница пушила у неком триместру трудноће су сведена на једно које говори да ли је пушила током читаве трудноће. Након чишћења и трансформације код појединих обиљежја се промијенио број недостајућих вредности.

Дефинисана је још једна трансформација пресликавања која се извршавала за потребе визуализације података и покретања модела. Пресликавање се односи на разврставање одговора у двије класе. Класом 0 су означавани одговори који индицирају позитивне факторе односно добре навике труднице. Класом 1 су означавани одговори који представљају негативне факторе односно лоше навике труднице. Пошто је од интереса уочавање негативних фактора зато је и одабрано да се такви одговори пресликавају на класу 1, па се ова класа назива још и позитивном класом.

4.2. Статистичка и истражна анализа

Пошто скуп података посједује већи број обиљежја биће истакнута битна запажања. Циљна варијабла у скупу података је питање број шеснаест јер се њиме провјерава да ли трудници недостају зуби.



Слика 1. *Дистрибуција циљне варијабле након пресликавања*

На основу слике 1. може се закључити да у скупу података постоји знатно више, скоро па дупло више, трудница којима недостаје бар један зуб у односу на оне које имају све зубе. Првенствено са аспекта оралног здравља ово запажање је веома забрињавајуће. Због много већег броја трудница којима недостаје бар један зуб јавља се проблем небалансираности скупа података. Требало је адекватно приступити ријешавању овог проблема.

Код анализе осталих обиљежја је такође уочено да постоји небалансиран однос. За сваки одговор које су испитанице дале било је више оних које нису имале све зубе, посматрано у односу на циљну варијаблу. Међутим у оним одговорима који представљају добре навике уочено је да је однос скоро једнак.

5. ПРИПРЕМА ПРЕДИКЦИОНИХ МОДЕЛА

За имплементацију предикционих модела коришћене су и библиотека *scikit-learn v0.24.1* и библиотека *statsmodels v0.12.2*. Модел стабала одлучивања и случајне шуме су имплементирани помоћу библиотеке *scikit-learn*. За различите технике балансирања скупа података је коришћена библиотека *imbalanced-learn v0.8.0*.

Како је циљ овог рада извршити предикцију недостатка зуба код трудница у односу на њене навике тако и предикциони модели представљају централну ствар од интереса. Прије покретања модела било је неопходно спровести одређене поступке припреме који су подијељени у четири фазе.

Прву фазу представља подјела скупа података. Да би се предикциони модели могли обучити неопходан је тренинг скуп података. За евалуацију модела је потребно обезбједити валидациони скуп и на крају је потребно модел покренути над непознатим тестним подацима. Након примјене ове фазе, од прочишћеног скупа података у коме су одбачени непотребни подаци, настају три скупа: тренинг, валидациони и тест скуп података. У једном сценарију покретања (сценарио 1) је коришћена унакрсна валидација за евалуацију модела, а у другом (сценарио 2) валидациони скуп.

У другој фази је дефинисан метод евалуације модела, одабрана је референтна метрика одзив (енгл. *Recall*) у односу на коју се мјере перформансе модела.

Трећа фаза укључује дефинисање предикционих модела. За предикцију недостатка зуба код трудница одабрана су три предикциона модела: логистичка регресија, стабла одлучивања и случајна шума.

У четвртој фази је дефинисано на који начин су модели покретани. Пошто скуп података садржи недостајуће вриједности било је неопходно креирати стратегије за њихово третирање. Примјењене су четири стратегије: одбацивање свих редова који садрже недостајуће вриједности (стратегија 1), замјена недостајућих вриједности медијаном (стратегија 2), замјена недостајућих вриједности заокруженом средњом вриједношћу одговора (стратегија 3) и замјена недостајућих вриједности најчешћом вриједношћу одговора (стратегија 4). Такође дефинисан је и начин на који се ријешава проблем небалансираности скупа података. Првенствено задржани су и небалансирани подаци да би модели били покретани и над њима, а након тога су примјењене и три технике балансирања, а то су: *under-sampling* [22], *over-sampling* [22] и *SMOTEENN* [22].

6. ПРИКАЗ РЕЗУЛТАТА И ДИСКУСИЈА

За модел логистичке регресије и случајне шуме постоји могућност испитивања важности обиљежја. Ово може помоћи у прегледу фактора који би могли утицати на орално здравље труднице. У табелама 1. и 2. су приказани најбољи резултати покретаних модела, у односу на метрику одзив.

Модел случајне шуме омогућава анализу вриједности средњег смањења нечистоће гдје се гледа које од обиљежја је током развијања стабла вршило највећи утицај да опредијели којој класи примјерак припада. У оба сценарија као важни фактори су се издвојили: фактори самопроцјене стања зуба труднице, фактори посјете стоматологу, фактори који се односе на навике прања зуба и економски показатељи.

Код модела логистичке регресије могуће је анализирати *p*-вриједности. Уколико је ниво значајности ове вриједности за неко обиљежје мањи од 0,05 онда се то обиљежје узима као статистички значајно. У овом случају као статистички значајни фактори се издвајају: фактори самопроцјене стања зуба труднице**,** фактори посјете стоматологу, трудничке навике исхране, ношење протезе, фактор старости, едукација и фактори везани за пушење.

Видљиво је да се резултати оба модела у одређеној мјери поклапају. Поред тога резултати се поклапају једним дијелом са оним што је пронађено у литератури и већ раније навођено.

Табела 1. *Приказ најбољих резултата сценарио 1*

Модел	**Стратегија – техника балансирања**	**Мјера над валидационом скупом**	**Мјера над тест скупом**
		одзив	одзив
Логистичка регресија	стратегија 2 – без балансирања	0,9704	0,8538
Случајна шума	стратегија 4 – без балансирања	0,9125	0,8656

Табела 2. *Приказ најбољих резултата сценарио 2*

Модел	**Стратегија – техника балансирања**	**Мјера над валидационом скупом**	**Мјера над тест скупом**
		одзив	одзив
Логистичка регресија	стратегија 3 – без балансирања	0,8480	0,8300
Случајна шума	стратегија 4 – без балансирања	0,9520	0,8814

7. ЗАКЉУЧАК

Примјеном истражне анализе уочено је да велики проценат популације има проблема са оралним здрављем што је могуће и тврдити јер је узорак веома велики у односу на оно што је пронађено у литератури [11,13,15-19,21]. У случају овог упитника има дупло више трудница које су одговориле да им недостаје бар један зуб у односу на оне које имају све зубе.

Помоћу предикционих модела пронађени су потенцијално важни фактори који могу утицати на орално здравље. Код свих предикционих модела су се истакли следећи фактори: трудничке навике посјете стоматологу и самопроцјена стања зуба. Други фактори су се такође показали као потенцијално важни, а то су: навике прања зуба, економски показатељи, ниво едукације, навике конзумирања цигарета, као и поједине навике исхране.

Као примарни циљ је била истакнута предикција да ли трудница може имати проблема са недостатком зуба. У односу на остварене резултате може се рећи да су модели показали изузетно добре перформансе и да би могли дати допринос у раном откривању потенцијалних проблема са недостатком зуба. Труднице би могле попунити већ на самом почетку трудноће упитник о својим навикама. Модели би се могли покренути над прикупљеним подацима, а затим би се могла извршити предикција. Резултати предикције путем модела би трудници могли указати на потенцијалне проблеме те би је то могло потаћи да посјети стоматолога и да превенцијом не дође до проблема са оралним здрављем.

8. ЛИТЕРАТУРА

[1] A. Sheiham, “Oral health, general health and quality of life,” *Bull. World Health Organ.*, vol. 83, pp. 644–644, Sep. 2005, doi: 10.1590/S0042-96862005000900004.

[2] P. E. Petersen, “The World Oral Health Report 2003: continuous improvement of oral health in the 21st century – the approach of the WHO Global Oral Health Programme,” *Community Dent. Oral Epidemiol.*, vol. 31, no. s1, pp. 3–24, 2003, doi: 10.1046/j..2003.com122.x.

[3] “measuring-oral-health-and-quality-of-life.pdf.” Accessed: Sep. 18, 2021. [Online]. Available: https://www.adelaide.edu.au/arcpoh/downloads/publications/reports/miscellaneous/measuring-oral-health-and-quality-of-life.pdf

[4] S. Vanka, A. Vanka, A. Bhambal, V. Saxena, S. Saxena, and S. Kumar, “Association of pregnant women periodontal status to preterm and low-birth weight babies: A systematic and evidence-based review,” *Dent. Res. J.*, vol. 9, pp. 368–80, Jul. 2012, doi: 10.4103/1735-3327.102757.

[5] V. Marla, R. Chettiar, D. Roy, and H. Ajmera, “The Importance of Oral Health during Pregnancy: A review,” vol. 5, Mar. 2018, doi: 10.5935/MedicalExpress.2018.mr.002.

[6] M. Pirie, I. Cooke, G. Linden, and C. Irwin, “Dental manifestations of pregnancy,” *Obstet. Gynaecol.*, vol. 9, no. 1, pp. 21–26, 2007, doi: 10.1576/toag.9.1.021.27292.

[7] L. W. Mills and D. T. Moses, “Oral Health During Pregnancy,” *MCN Am. J. Matern. Nurs.*, vol. 27, no. 5, pp. 275–280, Oct. 2002.

[8] “Uredba o Nacionalnom programu preventivne stomatološke zdravstvene zaštite: 22/2009-11.” https://www.pravno-informacioni-sistem.rs/SlGlasnikPortal/eli/rep/sgrs/vlada/uredba/2009/22/4/reg (accessed Sep. 18, 2021).

[9] M. Vulović *et al.*, *Program preventivne stomatološke zaštite stanovnika Srbije*. Zavod za udžbenike i nastavna sredstva, 1996.

[10] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, Illustrated edition. Cambridge, MA: The MIT Press, 2012.

[11] E. Kateeb and E. Momany, “Dental caries experience and associated risk indicators among Palestinian pregnant women in the Jerusalem area: a cross-sectional study,” *BMC Oral Health*, vol. 18, no. 1, p. 170, Oct. 2018, doi: 10.1186/s12903-018-0628-x.

[12] S. L. Russell, J. R. Ickovics, and R. A. Yaffee, “Exploring Potential Pathways Between Parity and Tooth Loss Among American Women,” *Am. J. Public Health*, vol. 98, no. 7, pp. 1263–1270, Jul. 2008, doi: 10.2105/AJPH.2007.124735.

[13] M. Deghatipour *et al.*, “Oral health status in relation to socioeconomic and behavioral factors among pregnant women: a community-based cross-sectional study,” *BMC Oral Health*, vol. 19, no. 1, p. 117, Jun. 2019, doi: 10.1186/s12903-019-0801-x.

[14] E. G. Kim, S. K. Park, and J.-H. Nho, “Factors Related to Maternal Oral Health Status: Focus on Pregnant and Breastfeeding Women,” *Healthcare*, vol. 9, no. 6, Art. no. 6, Jun. 2021, doi: 10.3390/healthcare9060708.

[15] J. A. Gil-Montoya, X. Leon-Rios, T. Rivero, M. Expósito-Ruiz, I. Perez-Castillo, and M. J. Aguilar-Cordero, “Factors associated with oral health-related quality of life during pregnancy: a prospective observational study,” *Qual. Life Res.*, May 2021, doi: 10.1007/s11136-021-02869-3.

[16] M. Radnai *et al.*, “The oral health status of postpartum mothers in South-East Hungary,” *Community Dent. Health*, vol. 24, no. 2, pp. 111–116, Jun. 2007.

[17] M. Wandera, I. M. Engebretsen, I. Okullo, J. K. Tumwine, A. N. Åstrøm, and the PROMISE-EBF Study Group, “Socio-demographic factors related to periodontal status and tooth loss of pregnant women in Mbale district, Uganda,” *BMC Oral Health*, vol. 9, no. 1, p. 18, Jul. 2009, doi: 10.1186/1472-6831-9-18.

[18] J.-N. Vergnes *et al.*, “Frequency and Risk Indicators of Tooth Decay among Pregnant Women in France: A Cross-Sectional Analysis,” *PLOS ONE*, vol. 7, no. 5, p. e33296, Jul. 2012, doi: 10.1371/journal.pone.0033296.

[19] N. Thomas, P. Middleton, and C. Crowther, “Oral and dental health care practices in pregnant women in Australia: A postnatal survey,” *BMC Pregnancy Childbirth*, vol. 8, p. 13, Feb. 2008, doi: 10.1186/1471-2393-8-13.

[20] A. Villa *et al.*, “Oral health and oral diseases in pregnancy: a multicentre survey of Italian postpartum women,” *Aust. Dent. J.*, vol. 58, no. 2, pp. 224–229, 2013, doi: 10.1111/adj.12058.

[21] S. A. Khamis, K. Asimakopoulou, T. Newton, and B. Daly, “The effect of dental health education on pregnant women’s adherence with toothbrushing and flossing — A randomized control trial,” *Community Dent. Oral Epidemiol.*, vol. 45, no. 5, pp. 469–477, 2017, doi: 10.1111/cdoe.12311.

[22] “imbalanced-learn documentation — Version 0.8.0.” https://imbalanced-learn.org/stable/ (accessed May 25, 2021).

Кратка биографија:

Мирослав Томић рођен је 1997. године у Бијељини, РС, БиХ. Завршио је средњу техничку школу „Михајло Пупин“ у Бијељини. Факултет техничких наука у Новом Саду, студијски програм Рачунарство и аутоматика, уписао је 2016. године. Након завршених основних студија уписао је мастер студије на истом факултету.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.39
DOI: https://doi.org/10.24867/16BE03Pajevic

METODE NENADGLEDANE SEGMENTACIJE SLIKE ZA MAPIRANJE STRESOM POGOĐENIH REGIONA POLJOPRIVREDNOG ZEMLJIŠTA

UNSUPERVISED IMAGE SEGMENTATION METHODS FOR MAPPING STRESS-AFFECTED REGIONS OF AGRICULTURAL LAND

Nina Pajević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U okviru rada opisana je segmentacija Sentinel-2 satelitskih snimaka na osnovu izvedenih vegetacionih indeksa u cilju mapiranja stresom pogođenih regiona poljoprivrednog zemljišta. Baza podataka sastoji se od vremenske serije multispektralnih slika 48 poljoprivrednih parcela u Vojvodini. Vremenska serija obuhvata slike za 13 datuma u periodu od marta do septembra 2020. godine. Primenjene su tri metode segmentacije: segmentacija postavljanjem praga, segmentacija upotrebom k-means algoritma, i segmentacija upotrebom pyImSegm biblioteke. Segmentacijom prvom metodom nije uspešno detektovana cela oštećena površina, dok su druge dve metode pokazale dobru sposobnost mapiranja slabije razvijenih regiona.*

Ključne reči: *Segmentacija, Mašinsko učenje, Kompjuterska vizija, Sentinel-2*

Abstract – *The paper describes the segmentation of Sentinel-2 satellite images based on derived vegetation indices in order to map stress-affected regions of agricultural land. The database consists of a time series of multispectral images of 48 agricultural parcels in Vojvodina. The time series includes images for 13 dates in the period from March to September 2020. Three segmentation methods were applied: threshold segmentation, segmentation using the k-means algorithm, and segmentation using the pyImSegm library. The first damaged area was not successfully detected by segmentation with the first method, while the other two methods showed a good ability to map less developed regions.*

Keywords: *Segmentation, Machine learning, Computer vision, Sentinel-2*

1. UVOD

Poljoprivredne parcele podložne su raznim negativnim uticajima koji mogu dovesti do smanjenja prinosa. Neki od faktora koji se odražavaju na razvoj kulture kao i vitalnost i potencijal klijavosti semena na parceli mogu biti suša, poplave i ostale vremenske nepogode koje zahvataju celu površinu parcele, ali i štetočine, bolesti, ili sastav zemljišta, koji mogu uticati samo na određene delove parcele.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio doc. dr Branko Brkljač.

Kako su poljoprivredne parcele uglavnom velikih površina i nepregledne sa Zemljine površine bez terenskog prolaska kroz celu parcelu, što je uglavnom nepraktično i vremenski vrlo zahtevno [1], detekcija problema ukoliko nije zahvaćena cela parcela (već postoje manje anomalije unutar parcele) predstavlja problem za poljoprivrednike.

Predmet rada jeste nenadgledana segmentacija multispektralnih slika poljoprivrednih parcela dobijenih pomoću satelita Sentinel-2 u svrhu detekcije (negativnih) anomalija unutar parcela koje bi mogle dovesti do smanjenja prinosa. Kombinovanjem različitih kanala multispektralne slike izvode se vegetacioni indeksi koji nose informacije o generalnom stanju useva, ali i o različitim fiziološkim parametrima biljke.

Razmotrena su tri pristupa rešavanju problema:
- segmentacija postavljanjem praga na osnovu raspodele vrednosti NDVI vegetacionog indeksa,
- segmentacija upotrebom *k-means* algoritma nad superpikselima multispektralne slike,
- segmentacija pomoću biblioteke *pyImSegm*.

2. PODACI SA SENTINEL-2 SATELITA

Sentinel-2 (S2) predstavlja satelit čiji cilj jeste fotografisanje Zemljine površine u srednje visokoj rezoluciji i u više spektralnih opsega [2]. Snimci dobijeni pomoću ovih satelita (pošto je u pitanju konstelacija od dva satelita, tzv. S2-A i S2-B) u prostornom domenu podeljeni su u granule (ploče) fiksne veličine kojima je predstavljena površina od 100 km^2, pri čemu je vremenski razmak između dve slike iste površine 5 dana. Na svakom od dva satelita u konstelaciji nalazi se optički multispektralni instrument (MSI) koji meri Sunčevo zračenje reflektovano sa Zemlje u 13 spektralnih opsega (kanala), od kojih 4 opsega imaju prostornu rezoluciju od 10 m po pikselu, 6 opsega rezoluciju 20 m po pikselu, i 3 opsega rezoluciju 60 m po pikselu. Granula sadrži po jednu zasebnu sliku za svaki kanal. U ovom radu, pre upotrebe opisanih satelitskih slika bilo je potrebno interpolirati kanale sa manjom prostornom rezolucijom na rezoluciju od 10 m.

3. OPIS BAZE PODATAKA

Satelitske slike korišćene u ovom radu preuzete su iz javno dostupne baze podataka Evropske svemirske agencije kroz program *Copernicus* za 13 datuma u periodu od marta do septembra 2020. godine. Parcele nad kojima je bilo potrebno izvršiti segmentaciju nalaze se u Vojvodini i zahvaćene su granulom sa oznakom 34TCR.

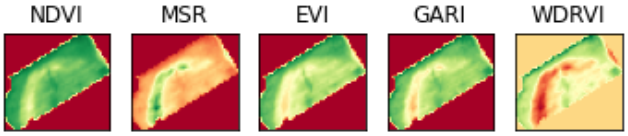
Iz preuzetih satelitskih snimaka isečene su slike unapred određenih poljoprivrednih parcela iz digitalnog katastra. Parcela ukupno ima 48 i nepoznato je koje kulture su posađene na njima. Međutim, pošto je cilj detekcija anomalija unutar površine parcela bez obzira na njihov sadržaj, navedeno nije predstavljalo otežavajuću okolnost.

3.1. Vegetacioni indeksi

Kako je cilj rada bio izdvojiti delove parcele koji zaostaju u razvoju za ostatkom parcele, bila je potrebna informacija o stanju useva. Procena ove informacije može da se dobije na osnovu analize različitih kombinacija kanala multispektralne slike kako bi se izveli odgovarajući vegetacioni indeksi. Dati postupak izvlačenja informacije zasniva se na činjenici da način na koji biljke apsorbuju i reflektuju svetlost različitih frekvencija zavisi od fiziološkog stanja i faze razvoja u kojoj se nalaze.

Korišćeni vegetacioni i spektralni indeksi [3] ilustrovani su odgovarajućim indeksiranim slikama korščenjm iste kolor mape na slici 1. i predstavljaju redom:

- NDVI (engl. *Normalized Difference Vegetation Index*)
- MSR (engl. *Modified simple ratio*)
- EVI (engl. *Enhanced vegetation index*)
- GARI (engl. *Green Atmospherically Resistant Index*)
- WDRVI (engl. *Wide Dynamic Range Vegetation Index*)

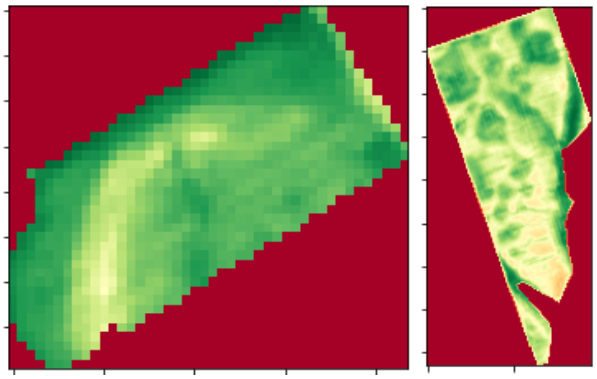


Slika 1. *Korišćeni vegetacioni indeksi*

3. METODE SEGMENTACIJE

S obzirom na to da su na parcelama posađene različite kulture i da se parcele prate i menjaju kroz vreme, odnosno usevi se razvijaju, vrednosti vegetacionih i spektralnih indeksa variraju od slike do slike. Iz tog razloga nije moguće postaviti jednu univerzalnu vrednost praga za segmentaciju ili napraviti metodu koja će raditi nad skupom slika, već je potrebno segmentaciji pristupiti na nivou jedne slike. Takođe, korisnicima ovakvog sistema za detekciju anomalija od koristi bi bilo da u svakom trenutku mogu da dobiju mapu segmenata na osnovu trenutnog stanja parcele. Sa druge strane, istorijski podaci mogu da posluže za detaljnije analize površina koje odgovaraju detektovanim segmentima (anomalijama) i predstavljaju potencijalan problem. Na taj način bilo bi moguće i prepoznavati vrste anomalija, od koje neke mogu biti prolaznog karaktera, a neke stalno prisutne kao posledica heterogenosti sastava zemljišta i njegove vlažnosti na nivou iste katastarske parcele. U ovom radu razmotrene su tri različite metode segmentacije za pojedinačne slike, bez istovremenog razmatranja čitave vremenske serije.

Vegetacioni indeks NDVI daje procenu opšteg stanja i zdravlja biljke i kao takav jedan je od najkorišćenijih izvedenih indeksa pri monitoringu vegetacije. Iz tog razloga rezultati metoda segmentacije evaluiraće se na osnovu NDVI slika parcela, npr. kao na slici 2.



Slika 2. *NDVI slike za dve različite parcele*

3.1. Segmentacija NDVI slike postavljanjem praga

Data metoda zasniva se na činjenici da vegetacioni indeks NDVI u slučaju parcele gde nema anomalija ima približno normalnu raspodelu. Ukoliko na parceli postoje anomalije u vidu oštećenja dela parcele, data oštećenja se u raspodeli indeksa izražavaju kao jače izražen rep Gausove krive na mestima gde NDVI uzima niže vrednosti. U tom slučaju moguće je postaviti prag baziran na srednjoj vrednosti i standardnoj devijaciji raspodele.

Međutim, ako su oštećene površine suviše velike, to se ne manifestuje kao rep Gausove krive, nego se menja oblik cele krive i više nije moguće postaviti prag po tom principu. Zbog ovoga postaje teško detektovati celu oštećenu površinu, već se detektuju samo najoštećeniji delovi parcele.

3.2. *K-means* algoritam

Segmentacija u ovom slučaju realizovana je korišćenjem iterativnog *k-means* algoritma. *K-means* algoritam [4] bazira se na definisanju centroida ili težišta klastera odnosno grupa u prostoru obeležja (višedimenzionalnih merenja) i grupisanju uzoraka u klaster čiji centroid je najbliži.

Kako bi se redukovala računska složenost algoritma i postigla manja osetljivost na incijalizaciju početnog stanja, pre klasterizacije NDVI slika parcele je izdeljena na superpiksele [5] pomoću *slic* (engl. *Simple Linear Iterative Clustering*) algoritma. Superpikseli predstavljaju grupe, odnosno regione susednih piksela koji su slični po određenim karakteristikama (npr. intenzitet) i mogu se smatrati lokalnim homogenim zonama unutar satelitske slike. Formiranjem superpiksela doprinosi se smanjenju računske složenosti (jer su entiteti koji se grupišu superpikseli kojih ima značajno manj u odnosu na broj piksela), ali i tome da konačna klasterizacija bude robusnija i sa manje šuma (uzoraka pogrešno pridruženih definisanim centroidima).

Nakon rezultata klasterizacije NDVI slike odgovarajuće parcele formira se nova slika koja se sastoji od pet kanala, odnosno od pet vegetacionih i spektralnih indeksa: NDVI, WDRVI, MSR, GARI i EVI. Za datu višekanalnu sliku se na osnovu granica superpiksela dobijenih u prethodnom koraku (na osnovu NDVI) za svaki od pet kanala izračunaju prosečne vrednosti intenziteta originalnih piksela unutar svih definisanih superpiksela i generiše se konačna petokanalna slika u kojoj svi pikseli unutar pojedinačnih superpiksela jednog kanala imaju istu

vrednost (izračunatu srednju vrednost). Motivacija za opisani postupak zasniva se na pretpostavci da će preterana segmentacija površine parcele kojoj odgovaraju granice superpiksela omogućiti lakše grupisanje sličnih zona unutar parcele. Pri tome su karakteristike zona opisane prosečnim vrednostima svakog od izračunatih indeksa na nivou odgovarajućeg superpiksela, tj. zone. Ovakav pristup često se označava i kao engl. "bottom-up approach".

Nad tako dobijenom slikom u kojoj vrednosti piksela u svakom od kanala odgovaraju prosečnim intenzitetima na nivou superpiksela, vrednosti se dodatno normalizuju po kanalima kako bi mogla da se primeni *PCA* (engl. *principal component analysis*) metoda za redukciju dimenzionalnosti pomoću koje se broj kanala smanjuje sa pet na samo dva. Naredni korak je *k-means* klasterizacija u 3 očekivana klastera ili grupe (pozadina, bolji klaster i lošiji klaster).

Nakon klasterizacije superpiksela, na osnovu NDVI slike izračunava se srednja vrednost celokupnog boljeg klastera. Svi pikseli unutar boljeg klastera klasifikuju se kao delovi parcele na kojima nije prisutno oštećenje. Potom se prolazi kroz superpiksele lošijeg klastera i računa se odnos srednje vrednosti datog superpiksela i srednje vrednosti boljeg klastera. Ukoliko je izračunati odnos manji od 90% dati superpiksel se finalno klasifikuje kao oštećenje, dok vrednosti veće od 90% znače da se superpiksel klasifikuje kao neoštećen deo parcele.

3.3. Biblioteka *pyImSegm*

Dati pristup bazira se na upotrebi biblioteke pyImSegm u svrhu nenadgledane segmentacije, a koja je implementirana na osnovu algoritma opisanog u radu [6]. Algoritam za segmentaciju sastoji se od sledećih koraka:

1. Segmentacija slike na superpiksele
2. Kreiranje obeležja na osnovu superpiksela
3. Računanje verovatnoće pripadnosti klasi
4. Klasifikacija superpiksela korišćenjem metode sečenja grafa (engl. *graph cut*)

Segmentacija pomoću ove biblioteke izvršena je na slici sastavljenoj od tri vegetaciona i spektalna indeksa: WDRVI, MSR i GARI.

4. REZULTATI

Na slici 2 prikazane su NDVI slike za dve različite parcele.

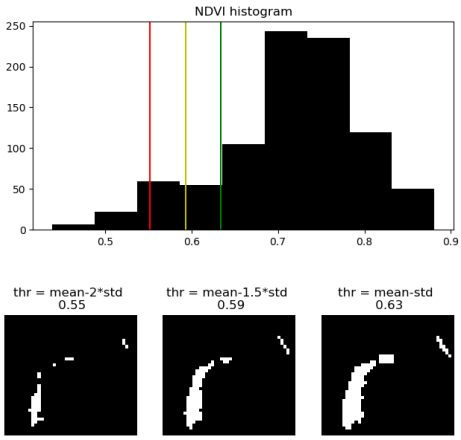
4.1. Rezultati metoda segmentacije

Na slikama 3-7 prikazani su rezultati segmentacije za sve tri metode.

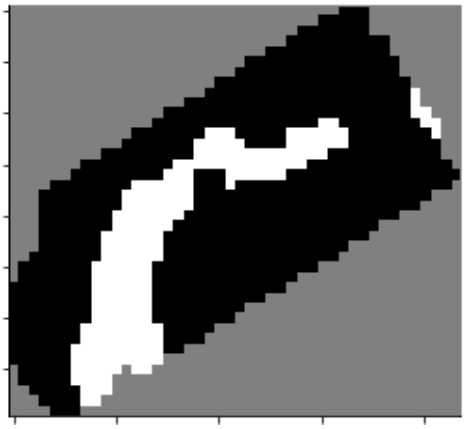
Belom bojom označeni su delovi parcele na kojima se nalaze anomalije, odnosno oštećenja. Crnom bojom označeni su delovi parcele koji nisu oštećeni, a sivom bojom označena je pozadina.

Vizuelnim upoređivanjem NDVI slika parcela i dobijenih rezultata dolazi se do zaključka da je prva metoda, odnosno metoda segmentacije pomoću praga nedovoljna da pravilno detektuje i obuhvati čitavu oštećenu površinu.

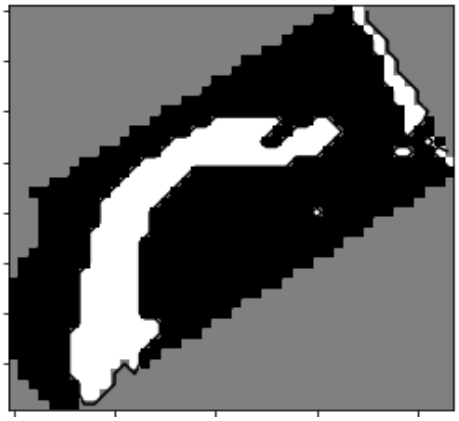
U slučaju druge i treće metode rezultati su dosta slični. Obe metode uhvatile su značajan deo površine parcele gde su vrednosti NDVI slabije, kao i zdrav deo parcele.



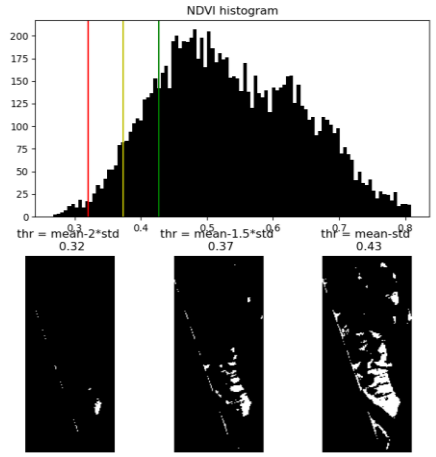
Slika 3. *Rezultat segmentacije prvom metodom*



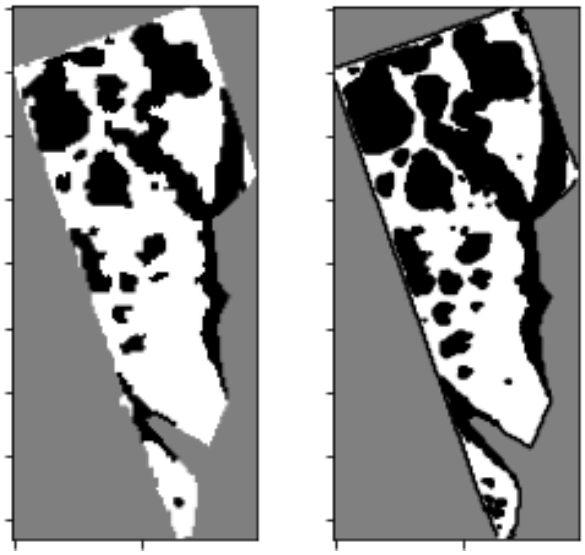
Slika 4. *Rezultat segmentacije drugom metodom*



Slika 5. *Rezultat segmentacije trećom metodom*



Slika 6. *Rezultat segmentacije prvom metodom*



(a) (b)

Slika 7. *Rezultat segmentacije: drugom (a) i trećom metodom (b)*

Rezultati treće metode pokazuju više uhvaćenih detalja prilikom segmentacije, međutim treba uzeti u obzir da je problem, odnosno zadatak ovog rada, bio da se segmentiraju delovi parcele koji zaostaju u razvoju za zdravim delom parcele. Samim tim, prednost druge metode jeste to što se može odrediti koliki stepen zaostatka u odnosu na celu parcelu je dopušten da bi se deo parcele smatrao zdravim. Ovo može predstavljati upravljački parametar za odgovarajući korisnički interfejs kojim bi se u zavisnosti od postavljenog praga generisale različite mape segmentacije.

Na taj način, ukoliko bi se dato rešenje upotrebljavalo u praksi, poljoprivrednici bi sami mogli da podešavaju segmentaciju u zavisnosti od potreba.
Segmentacijom parcela na opisani način moguće je pratiti razvoj useva kroz vreme. Na slici 8. prikazane su 3 slike iste katastarske parcele segmentirane drugom metodom za datume 02.08.2020, 17.08.2020. i 27.08.2020.

Slika 8. *Rezultati segmentacije vremenske serije slika drugom metodom* (datumi slika su 2.8.2020, 17.8.2020. i 27.8.2020, posmatrano sa leva na desno)

Praćenjem stanja parcele kroz vreme moguće je pravovremeno reagovati i sanirati štetu ukoliko dođe do pojave oštećenja. Segmentacijom ove vremenske serije slika utvrđeno je da je površina oštećenog dela parcele 22.72% cele površine parcele za prvi datum, 44.95% za drugi, i 40.5% za poslednji datum.

5. ZAKLJUČAK

U poljoprivredi se neretko dešava da manji delovi parcele zaostanu u razvoju zbog gubitaka vigora semena, pojave štetočina, bolesti. Upotrebom satelitskih slika u svrhu monitoringa parcela, praćenja razvoja i zdravlja useva, moguće je predupredi ti i sanirati na vreme neke od negativnih pojava. Kreiranje rešenja za detekciju slabije razvijenih regiona poljoprivrednog zemljišta moglo bi biti od velike koristi poljoprivrednicima.

Predložene metode za segmentaciju mogle bi dodatno da se poboljšaju ukoliko bi se unapred identifikovali i definisali tipovi anomalija koje bi trebalo detektovati, i u skladu sa tim odredio minimalni skup merenja koja bi omogućila optimalno odlučivanje.

Takođe, predložena nenadgledana segmentacija može da predstavlja polaznu osnovu za identifikaciju tipova anomalija koje su od interesa za određenu vrstu useva, s obzirom da određene anomalije mogu predstavljati kombinaciju različitih uzroka. Tako identifikovani tipovi anomalija omogućili bi primenu metoda nadgledane segmentacije, bez obzira na raspoloživi skup merenja.

6. LITERATURA

[1] Zheng, Q., Huang, W., Cui, X., Shi, Y. and Liu, L., 2018. New Spectral Index for Detecting Wheat Yellow Rust Using Sentinel-2 Multispectral Imagery. *Sensors*, 18(3), p.868.

[2] Sentinel-2 User Handbook, https://sentinel.esa.int/documents/247904/685211/Sentinel-2_User_Handbook (pristupljeno u septembru 2021.)

[3] Chemura, A., Mutanga, O. and Dube, T., 2016. Separability of coffee leaf rust infection levels with machine learning methods at Sentinel-2 MSI spectral resolutions. *Precision Agriculture*, 18(5), pp.859-881.

[4] https://stanford.edu/~cpiech/cs221/handouts/-kmeans.htm l (pristupljeno u septembru 2021.)

[5] https://www.mathworks.com/help/images/ref/-superpixels.html (pristupljeno u septembru 2021.)

[6] Borovec, J., Švihlík, J., Kybic, J. and Habart, D., 2017. Supervised and unsupervised segmentation using superpixels, model estimation, and graph cut. *Journal of Electronic Imaging*, 26(06), p.1.

Kratka biografija:

Nina Pajević rođena je u Novom Sadu 1997. god. Osnovne akademske studije na Departmanu za energetiku, elektroniku i telekomunikacije, Fakultet tehničkih nauka, Univerzitet u Novom Sadu, smer Obrada signala, uspešno je završila 2020. god. Na istom fakultetu i studijskom programu upisuje i master akademske studije i brani master rad 2021. godine.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004:007.5
DOI: https://doi.org/10.24867/16BE04Dimitrijevic

PROJEKTOVANJE SISTEMA ZA MERENJE SNAGE PODRŽANOG VIRTUELNOM INSTRUMENTACIJOM

DESIGN OF A POWER MEASUREMENT SYSTEM SUPPORTED BY VIRTUAL INSTRUMENTATION

Aleksandar Dimitrijević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U radu je prikazana praktična realizacija dodatne funkcije za merno-akvizicioni sistem (MAS) koja omogućava precizno merenje snage uz pomoć CompactDAQ platforme i LabVIEW razvojnog okruženja. U radu je prikazan način rada modula potrebnih za realizaciju navedene funkcije, njihova međusobna sinhronizacija, detaljna analiza i objašnjenje rezultata merenja, kao i softverski način proračuna snage i kompenzacije transportnog kašnjenja koja je posledica nesavršenosti prilagodnika signala. Takođe, prikazan je kratak opis mernih kanala MAS-a i instrumentacije.*

Ključne reči: *merenje snage, sinhronizacija, merno-akvizicioni sistem, transporno kašnjenje*

Abstract – *It is presented and practically implemented in the paper an additional function for the measurement and acquisition system. This function allows the precise measurement of the power using the CompactDAQ platform and LabVIEW development environment. It is presented in the paper the work methods for every module, which was used for implementation of this function, their mutual synchronization, detailed analysis and measurement, as well as software calculations of power, and compensation of the transport delay which is a consequence of the imperfection of signal conditioners. Also, it is presented the short description of measurement channels and instrumentation.*

Keywords *power measurement, synchronization, measurement and acquisition system, transport delay*

1. UVOD

Merenje električne snage, sa aspekta energetike, predstavlja jedno od bitnijih merenja čiji rezultat korisniku omogućava pregled trenutnog stanja posmatranog sistema.

Precizno merenje snage je jedan od zahteva pri projektovanju takvih sistema.

Ideja za izdradu ovog rada, proistekla je iz potrebe da se na već postojećem merno-akvizicionom sistemu (*MAS*) koji je razvio Elektrotehnički institut Nikola Tesla a.d Beograd implementira funkcija za precizno merenje snage. Implementacijom funkcije preciznog merenja snage, unapređuje se već postojeća multifunkcionalnost *MAS-a,* ubrzava se i olakšava postupak merenja, isključuje se potreba za primenu drugih, posebnih instrumenata za merenje snage.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Platon Sovilj, red. prof.

2. OPIS HARDVERA

MAS se sastoji od šasije sa mernim i izlaznim karticama, priključnih klema, napajanja, prilagodnika signala i izbornika koji su međusobno povezani i postavljeni u isto kućište. Šasija i merne kartice su proizvodnje *National Instruments (NI)*

2.1. cDAQ-9178

Šasija *cDAQ-9178* [1] je šasija koja je dizajnirana za male, prenosne merne i simulacione sisteme. Ima mogućnost *plug-and-play-a*, odnosno mogućnost jednostavne izmene konfiguracije sistema vađenjem jedne i/ili ubacivanjem druge kartice. Za rad koristi *I/O* module *C* serije kojima se mogu kreirati različite kombinacije analognih *I/O* modula, digitalnih *I/O* modula kao i brojača/tajmera. Komunikaciju sa računarom izvršava putem USB porta. Izgled šasije bez umetnutih kartica prikazan je na *sl.1.*

Slika 1: Izgled cDAQ-9178 šasije

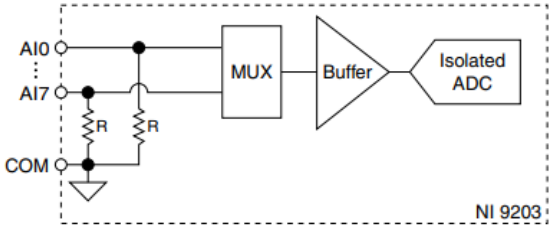
2.2. NI 9203 DAQ kartica

Analogne ulazne kartice *C* serije za *NI CompactDAQ* i *CompactRIO* platforme pružaju podršku visoko performansnim merenjima za široki spektar industrijskih, automobilskih i laboratorijskih senzora. U svaku karticu ugrađen je interni prilagodnik signala i priključnica za lako pričvršćivanje provodnika.

Modul *NI-9203* [2] je merna kartica *C* serije sa 8 analognih ulaznih kanala namenjen za aplikacije kontrole i nadzora visokih performansi. Ima mogućnost programabilnih ulaznih opsega. Bipolarni opseg ±20 mA i unipolarni od 0 do 20 mA. Poseduje jedan 16-bitni *A/D* konvertor. Maksimalna brzina uzorkovanja je 200 kS/s. Izgled kartice prikazan je na *sl.2.*

Slika 2: *Izgled NI-9203 merene kartice*

Tokom izrade ovog rada, primenom merne kartice ovog tipa merili su se naponi i struje trofaznog sistema uz primenu dodatnih eksternih prilagodnika signala. Na *sl. 3.* prikazana je šema merne kartice *NI-9203.*



Slika 3: *Šematski prikaz NI-9203 merne kartice*

Svaki signal doveden na karticu može biti povezan na dva načina:

- Povezivanje ulaza u odnosu na zajedničku tačku (AI0-COM)
- Diferencijalno povezivanje (AI0-AI1)

3. KORIŠĆENI UREĐAJI I MERILA

Kao simulator trofaznog sistema korišćen je trofazni izvor struja i napona promenljive amplitude, učestanosti i faze RETOM-51 [3]. Za određivanje kašnjenja signala na eksternim prilagodnicima signala korišćen je osciloskop FLUKE 190-102 [4] dok je za precizno merenje snage korišćen precizni mrežni analizator ZES ZIMMER LMG640 [5].

3.1. FLUKE 190-102 osciloskop

FLUKE 190-102 je dvokanalni osciloskop sa sigurnosnom ocenom *CAT III 1000V/CAT IV 600V*, koga odlikuju visoke performanse od kojih izdvajamo:

- Propusni opseg od 100MHz
- Odbirkovanje u realnom vremenu brzine do 1.25GS/s
- Model poseduje memorijski prostor koji čuva do 10000 semplova po kanalu čime omogućava detaljnu analizu svakog dela signala

Izgled uređaja prikazan je na *sl.4.*

Slika 4: *Osciloskop FLUKE 190-102*

3.2. RETOM-51

RETOM-51 je programabilni uređaj za simulaciju naponskih i strujnih signala.
Izgled uređaja prikazan je na slici 5.

Slika 5: *RETOM-51*

RETOM-51 može da generiše prostoperiodične signale efektivne vrednosti napona do 120V i struja do 20A, u trofasnom rasporedu.

3.4. ZES ZIMMER LMG640 precizni analizator snage

LMG640 je uređaj za merenje snage i kvaliteta električne energeije. U ovom radu upotrebljen je u svrhu kontrolnog merenja za verifikaciju projektovanog sistema. Neke od njegovih osobina su:

- Tačnost od 0,015% izmerene vrednosti +0,01% od opsega
- Puni dinamički opseg struje od 500µA do 21A i napona od 3mV do 1000V po kanalu
- Analogni propusni opseg DC do 10Mhz
- Modularna konfiguracija sa 1 do 4 katala za marenje snage
- Uzorkovanje bez greške do 18 bita i vreme ciklusa od 30ms
- Kašnjenje između naponskog i strujnog ulaza <3ns, vrlo precizna merenja pri malim faktorima snage ili/i visokim frekvencijama

Izgled uređaja prikazan je na *slici 6.*

Slika 6: *ZES ZIMMER LMG640*

4. KORIŠĆENI SOFTVER

Tokom izrade rada za upotrebu sa *CompactDAQ* platformom, korišćeno je *LabVIEW* razvojno okruženje proizvođača *NI*. Ovo razvojno okruženje zasnovano je na vizuelnom programiranju što omogućava korisniku da relativno brzo napreduje u projektovanju željene aplikacije.

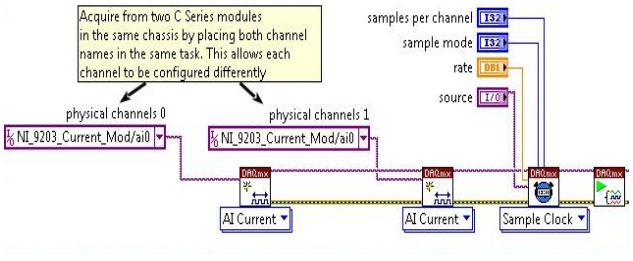
Najčešće primene LabVIEW-a su u izradi sistema za akviziciju, upravljanje instrumentacijom, automatizaciju ispitivanja, analizu i procesiranje signala, industrijsku kontrolu procesa, izradu embeded sistema i dr.

5. SINHRONIZACIJA IZMEĐU MODULA

Prvi korak ka realizaciji rada je bila sinhronizacija dve merne kartice tipa *NI-9203*. Prva tri kanala jedne merne kartice bila su namenjena za merenje napona dok su prva tri kanala druge merne kartice bila namenjena za merenje struja trofaznog sistema.

Kartice se povezuju na šasiju *cDAQ-9178* čiji je zadatak da podatke preuzete sa kartica prosledi ka računaru.

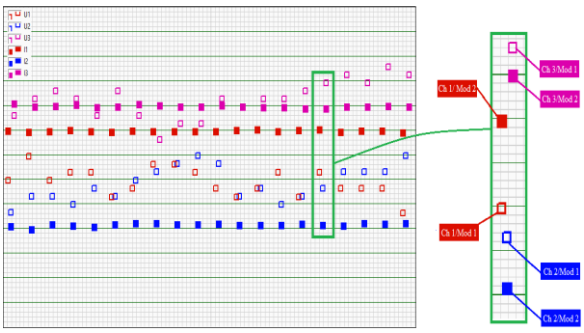
Kako bi se dobio efekat istovremenosti u očitavanju multipleksera svake od kartica neophodno je proslediti isti sinhronizacioni signal svakoj kartici. Na *sl.7.* prikazan je deo koda koji prikazuje jedan od načina na koji se moduli mogu sinhronizovati.



Slika 7: *Primer koda za sinhronizaciju mernih kartica*

Kao što je već navedeno u predhodnom tekstu, moduli se povezuju u isti task, prosleđuje im se isti takt za sinhronizaciju koji dobijaju od šasije konfigurisanjem *DAQmx Timing* funkcije, nakon čega započinju merenje u isto vreme kada funkcija *DAQmx Start Task* postane aktivna.

Kako bi se pokazalo da je sistem sinhronizovan, korišćenjem aplikacije napravljen je snimak koji dokazuje sinhrono očitavanje kanala, videti *sl. 8.*



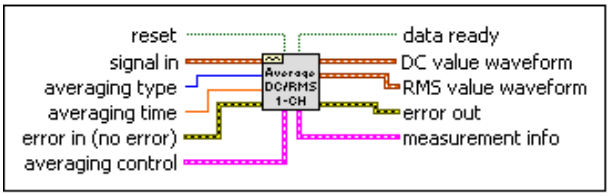
Slika 8: *Pregled trenutaka odbirkovanja signala svakog kanala*

Na slici praznim kvadratima obeleženi su odbirci signala prvog modula koji snimaju naponske signale. Sa popunjenim kvadratima obeleženi su odbirci signala drugog modula koji snimaju strujne signale. Na osnovu priložene slike zaključuje se da kanali, parovi signala iste boje, imaju istovremeno očitavanje.

6. MERENJE SNAGE

Funkcija za merenje snage implementirana je u postojeću aplikaciju *MAS*. Funkcija koristi neobrađene signale koji dolaze iz memorijskog bafera. Prilikom izrade, zahtev korisnika je bio da projektovani sistem treba da zadovoljava merenje snage samo za frekvenciju od 50Hz, koja pokriva većinu potreba merenja.

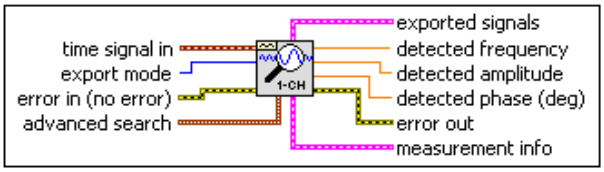
Navedena funkcija računa snagu na osnovu trenutne vrednosti struje i trenutne vrednosti napona. Pošto je frekvencija poznata, od tih trenutnih vrednosti signala može se proračunati efektivna vrednost signala sa ugrađenom funkcijom *Averaged DC-RMS*, videti *sl. 9.*



Slika 9: LabVIEW *funkcija za proračun efektivne vrednosti signala*

U funkciju *Averaged DC-RMS* dovodi se određeni ulazni signal. Kao parametar se prosleđuje vreme merenja, koji označava dužinu intervala signala za koju će funkcija proračunavati efektivne vrednosti signala. U ovom slučaju interval u kojem se vrši proračun je *t=0,02s.*

Nakon proračuna efektivne vrednosti, pristupa se merenju fazne razlike između signala. U prethodnom poglavlju prikazano je objašnjenje sinhronizacije kartica i način očitavanja kanala. Zaključeno je da ne postoji fazna razlika u očitavanju između parova kanala. Funkcija koja obezbeđuje ovakvo merenje naziva se *Single Tone Information* koja je u okviru palete za obradu signala. Prikaz izgleda funkcije je na *sl. 10.*



Slika 10: LabVIEW *funkcija za proračun faznog pomeraja signala*

Navedena funkcija kao izlaz daje veliki broj informacija o signalu koji je doveden na njen ulaz. Korišćene su:

- Merenje frekvencije
- Merenje amplitude
- Merenje faze u stepenima

Korišćenjem prethodno navedenih funkcija izračunava se snaga sistema:

$$S = U_{ef1} \times I_{ef1} + U_{ef2} \times I_{ef2} + U_{ef3} \times I_{ef3} \quad (1)$$

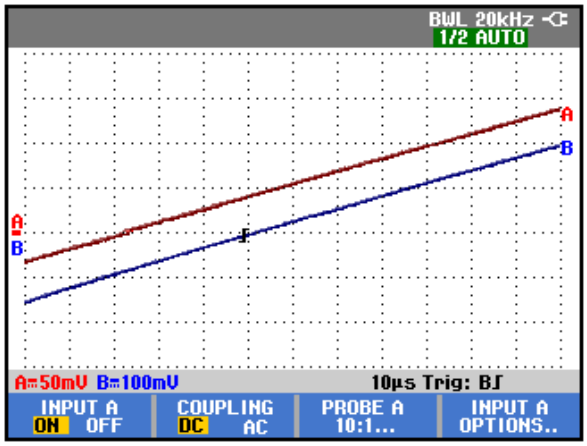
dok se uvođenjem vrednosti međufazne razlike između napona i struja izračunavaju aktivna i reaktivna snaga sistema kao:

$$P = U_{ef1} \times I_{ef1} \times \cos\varphi + U_{ef2} \times I_{ef2} \times \cos\varphi + U_{ef3} \times I_{ef3} \times \cos\varphi \quad (2)$$

$$Q = U_{ef1} \times I_{ef1} \times \sin\varphi + U_{ef2} \times I_{ef2} \times \sin\varphi + U_{ef3} \times I_{ef3} \times \sin\varphi \quad (3)$$

5. KOMPENZACIJA KAŠNJENJA

Analizirano je kašnjenje u svakom mernom kanalu. Utvrđeno je da prilagodnik signala koji preslikava ulazni naponski signal na svoj izlaz, u opsegu od ±20 mA, unosi određenu stalnu razliku u fazi između ulaznog i izlaznog siglala. Na *sl. 11.* prikazani su signali snimljeni osciloskopom FLUKE 190-102.



Slika 11: *Snimak fazne razlike signala*

Prvi signal, obeležen sa *A*, predstavlja mereni signal koji se dovodi na ulaz prilagodnika signala. Drugi signal, obeležen sa *B*, predstavlja signal na njegovom izlazu. Pre merenja svaki od signala izjednačen je po amplitudi i postavljen je da bude simetričan u odnosu na osu koja je odabrana za referentnu.

Izmerena je stalna vrednost kašnjenja $\Delta t = 30\mu s$.

Za potrebe kompenzacije prethodno utvrđenog kašnenja projektovano je odgovarajuće softversko rešenje. Primenjeno rešenje ogleda se u tome što se vrednosti strujnih signala ukupno vremenski pomeraju u odnosu na naponske signale za vrednost izmerenog kašnjenja. Rezultat je proračun tačnog faznog stava između vektora napona i struje u svakoj fazi.

U kanalima za merenje struje kompenzacija u kašnjenju nije bila potrebna jer je utvrđeno kašnjenje koje je manje od 0,2µs

6. VERIFIKACIJA

Softverski realizovana funkcija merenja snage sa uključenom kompenzacijom faznog kašnjenja implementirana je u aplikaciju *MAS*.

Verifikacija ispravnosti i tačnosti merenja funkcije izvršena je upotrebom preciznog analizatora snage opisanog u 3.4 koji je sa *MAS* bio istovremeno priključen na izor trofaznog napona i struja koji je opisan 3.3.

Merenjem je utvrđeno da je razlika u vrednosti izmerene trenutne snage bez kompenzacije faznog kašnjenja i sa uključenom kompenzacijom faznog kašnjenja veća od 1%.

7. ZAKLJUČAK

U radu je prikazana uspešna praktična primena softverskog alata u realizaciji merenja snage trofaznog sistema pomoću merno-akvizicionog sistema opšte namene.

Svaki korak u projektovanju softverskog rešenja zasnovan je na merenjima. Konačni rezultat prikazanog softverskog rešenja verifikovan je merenjem.

Softversko rešenje je uspešno praktično primenjeno.

8. LITERATURA

[1] https://www.ni.com/pdf/manuals/372838e.pdf

[2] https://www.ni.com/pdf/manuals/374070a_02.pdf

[3] https://www.fluke.com/en-us/product/electrical-testing/portable-oscilloscopes/190-series-ii/fluke-190-ii-190-102#specs-tab

[4] RETOM™-51 User Manual

[5] https://www.zes.com/en/Products/Discontinued-Products/Energy-and-Power-Meters/LMG640

Kratka biografija:

Aleksandar Dimitrijević rođen je u Surdulici 1997. god. Bachelor rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Merenje i regulacija odbranio je 2020. god. kontakt: adimitrijevic22@gmail.com

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.85
DOI: https://doi.org/10.24867/16BE05Katic

SISTEM ZA ODGOVORE NA PITANJA U DOMENU FITNESA ZASNOVAN NA MAŠINSKOM UČENJU

QUESTION ANSWERING SYSTEM IN FITNESS DOMAIN BASED ON MACHINE LEARNING

Sava Katić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu predstavljen je sistem za odgovore na pitanja u oblasti fitnesa i nutricionizma, koji dovoljno dobro generalizuje i za opšti domen. Kao ulaz model prima pitanje u obliku niza karaktera, a potom traži najbolje kandidate dokumente u bazi znanja, koji imaju odgovor na pitanje koje je na ulazu u sistem postavljeno.*

Ključne reči: *Odgovori na pitanja, Jezički modeli, NLP, QA, BERT*

Abstract – *This paper presents system for question answering focused on fitness and nutrition field, that works just as well in open domain. As an input model accepts a question in a form of array of characters and finds best document candidates in a knowledge base from which the actual answer is extracted.*

Keywords: *Chatbot, Language models, NLP, QA, BERT*

1. UVOD

Jezik i govor su delom ono što nas čini ljudima. Nije iznenađenje da nam je daleko lakše komunicirati prirodnim jezikom nego tipkanjem, kliktanjem i kodiranjem. Noviji algoritmi veštačke inteligencije pomažu nam da premostimo tu razliku tako što simuliraju razumevanje našeg prirodnog jezika i konverzaciju u istom. U konkretnom domenu fitnesa na koji se ovaj rad fokusira, ovakav sistem bi omogućio da se besplatno dođe do odgovora na pitanja (engl. *question answering*) u vezi sa zdravljem, vežbanjem i ishranom na brz, jednostavan i stalno pristupačan način. Komponente koje čine sistem su: baza znanja, čitač i sakupljač. Baza znanja je baza podataka koja sadrži tekstualne dokumente iz određenog domena na osnovu kojih se daju odgovori, organizovanih tako da ih je jednostavno pretraživati. Sakupljač je komponenta koja pretražuje dokumente iz baze znanja i pronalazi kandidate koji bi mogli da sadrže odgovor na postavljeno pitanje koje je ulaz u sistem, dok je čitač neuronska mreža koja u dokumentima kandidatima pronalazi odgovor na dato pitanje. Za skup podataka korišćeno je više izvora kako bi se izgenerisali dokumenti koji sadrže bitne informacije u polju fitnesa i nutricionizma. Na osnovu ovih dokumenata su generisana pitanja i odgovori koristeći *cloze translation* [2].

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

Nakon analize skupa podataka, pretprocesiranja teksta i izdvajanja odgovarajućih obeležja, iskorišćena je čitač sakupljač arhitektura koja ima za cilj da izdvoji relevantne dokumente na osnovu prosleđenog pitanja, a zatim i nađe odgovor u pasusima tih dokumenata. Za čitač komponentu korišćeni su *RoBERTa* [4] i *MiniLM* [3] arhitekture mreže, koje koriste *transformer* i bidirekcionalnost čiji su se rezultati međusobno poredili. Za sakupljač komponentu implementirani su *TF-IDF*, *BM25* i *DPR* [5] pristupi. Baza znanja se čuvala u *ElasticSearch* [1] servisu koji omogućava brzu pretragu. Nakon toga, vršeno je evaluiranje i poređenje rezultata između različitih arhitektura kako bi utvrdili u kojim slučajevima model greši i kako bi se greške mogle izbeći. Napravljena je i *web* aplikacija koja nudi interfejs za postavljanje pitanja i poziva odgovarajuće modele putem *HTTP* protokola.

U narednom poglavlju biće analizirana relevantna literatura za problem odgovora na pitanja koji se analizira. U trećem poglavlju biće opisana predložena arhitektura sistema kao i implementacija rešenja dok četvrto poglavlje nudi uvid u rezultate i njihovu diskusiju. U petom poglavlju dat je zaključak o samom radu i problemu koji je rešavan.

2. PRETHODNA REŠENJA

Modeli za odgovore na pitanja su modeli mašinskog ili dubokog učenja koji mogu da odgovore na određena pitanja na osnovu konteksta, tj. dela teksta koji sadrži odgovor na postavljeno pitanje. Postoje implementacije koje ne zahtevaju da kontekst u kom se nalazi odgovor bude prosleđen i ovaj rad se bavi jednim takvim rešenjem sa kraja na kraj (engl. *end-end*).

Postoji više *NLP* modela koji su u prošlosti primenjeni za rešavanje problema odgovora na pitanja. Jedan od njih koristi gramatičko označavanje i *TF-IDF* pristup da pitanje koje je postavljeno pretraži na *Yahoo Answers*[2] i nađe odgovor na isto. Zatim, potpuno nova arhitektura neuronskih mreža zasnovana na pažnji, posebno na samopažnji (engl. *self-attention*) nazvana *transformer* je zaista ponudila najbolji pristup za reprezentaciju teksta i prirodnog jezika [1, 3, 4, 6, 7].

Jezički model je probabilistički model koji uči verovatnoću pojavljivanja rečenice ili sekvence reči na osnovu primera teksta viđenih tokom treninga, gde je najzastupljeniji model *BERT* [1]. Razlog za popularnost *BERT*

[1] https://www.elastic.co/
[2] https://www.yahoo.com/

modela je što uvodi inovaciju koristeći bidirekciono treniranje *transformera* za modelovanje jezika (engl. *language modeling*). Ovo se razlikuje od prethodnih pokušaja da se tekst analizira i da se modeli treniraju ili sa leva na desno ili kombinujući sa leva na desno i sa desna na levo pristup.
Postoji dosta pretreniranih modifikacija na *BERT* model koje su dale nove *state of the art* modele , a takođe i treniranih za specifični domen kao što su *BioBERT*, *SciBERT* i *ClinicalBERT* [7]. Takođe, drugi modeli dobijeni korišćenjem destilacije znanja izložene u *MiniLM* radu [3] zadržavaju najveći deo performansi pretreniranih *state of the art* modela, dok imaju mnogo manje parametara.
Za domen fitnesa, nije pronađen odgovarajući model, skup podataka kao ni istraživanje na temu odgovora na pitanja. Iz tog razloga skup podataka će se generisati sintetički. U narednom poglavlju biće opisan implementirani sistem sa kraja na kraj za odgovore na pitanja.

3. METODOLOGIJA I ALATI

Ovo poglavlje posvećeno je korišćenim alatima (poglavlje 3.1) kao i prikazu primenjene metodologije i arhitekturi sistema sa kraja na kraj koji na osnovu pitanja parsira veliki korpus dokumenata u potrazi za odgovorom.

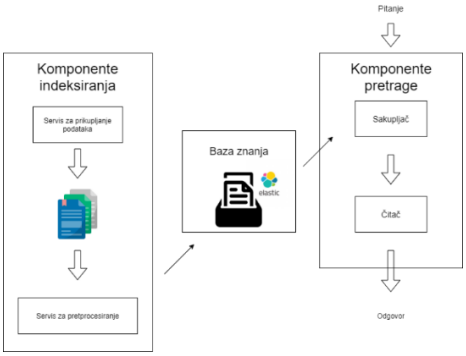
3.1. Korišćeni alati

Server na kom je treniran model je *p2.xlarge* tip instance sa jakom grafičkom karticom u okviru *AWS*[3] alata. Server ima 2.7 GHz *Intel Xeon E5-2686* v4 procesor i *NVIDIA K80 12 GB* grafičku karticu. Radi portabilnosti treniranja iskorišćena je kontejnerizacija (engl. *containerization*) koristeći *Docker*[4] softver.

Pretrenirani modeli za problem odgovora na pitanja specificirani su koristeći *Deepset - Haystack* [5] radno okruženje koje se oslanja na *PyTorch* [6] biblioteku za razvoj modela veštačke inteligencije.

3.2. Arhitektura rešenja

Predloženo rešenje fokusira se na podatke u domenu fitnesa, ali je dovoljno generično da bude upotrebljeno i za druge domene. Iskorišćena je čitač sakupljač arhitektura kako bi se dobio što performantniji sistem, a komunikacija komponenti može se videti na slici 1. Objašnjenje svake komponente je u nastavku.



Slika 1: *Arhitektura rešenja*

[3] https://aws.amazon.com/
[4] https://www.docker.com/
[5] https://github.com/deepset-ai/haystack
[6] https://pytorch.org/

Da bi većina pitanja i pojmova o fitnesu i nutricionizmu bila pokrivena, korišćeno je nekoliko izvora podataka kako bi pokrivali što veći skup činjenica. Tekstovi za trening i evaluacione podatke su skinuti sa sajtova *muscleandfitness*[7], *myfitnesspal*[8], *breakingmusclefitness*[9] , *advancedhumanperformance*[10]. Pored ovih sajtova, korišćeni su artikli sa *Wikipedia* u kategorijama fitnesa, trčanja, bodibildinga, joge i nutricionizma. Da bi bila pokrivena i pitanja o najboljim vežbama za određenu grupu mišića korišćen je *API*[11] koji daje najbolje vežbe spram prosleđene grupe mišića, kako bi se generisale rečenice koje prave preporuke u vidu vežbi. Potom se nad ovako dobijenim podacima vrši pretprocesiranje i dobijeni dokumenti se smeštaju u bazu dokumenata.
U okviru pretprocesiranja podataka izbačene su stop reči i rađena je lematizacija (engl. *lemmatization*) i *stemming*. Generisani dokumenti su se prosleđivali *BERT* modelu koji je generisao tri ključne reči tako što je pravio reprezentaciju (engl. *embedding*) celog dokumenta i svake reči u dokumentu. Potom su dokumenti čije su ključne reči van domena relevatnog za ovo istraživanje bili isfiltrirani.
Za generisanje pitanja i odgovora kako bi se napravio trening skup kojim će se modeli dotrenirati korišćen je pristup putem ekstrakcije imenica, fraza ili imenovanih entiteta [2] koji se potom maskiraju i služe za generisanje pitanja u prirodnom govoru. Tako generisan odgovor je smatran tačnim (engl. *ground truth*) Potom se generisano pitanje prosleđuje *T5 Text-To-Text Transfer Transformer* modelu koji je istreniran nad *Quora*[12] skupom podataka da parafrazira pitanje prosleđeno na ulazu.
Pored ovako generisanog skupa podataka, iskorišćen je i *Natural Questions Google Search*[13] skup podataka koji sadrži pitanja otvorenog domena, da bi se proverilo da li model gubi sposobnost generalizacije nakon što se dotrenira na generisanim pitanjima iz domena fitnesa.

Sakupljač ili pretraživač je filter komponenta koja prolazi kroz celokupnu bazu znanja i pronalazi dokumente koji su kandidati na osnovu pitanja prosleđenog na ulazu. Ovaj alat služi da se izbace negativni kandidati i uštedi vreme čitaču da ne radi dodatan posao i prolazi kroz sve dokumente kako bi našao odgovor na pitanje, ubrzavajući proces upita na taj način.

Neke od mogućih reprezentacija sakupljača:

- *TF-IDF* - statistička mera odnosno funkcija rankiranja relevantnosti dokumenata spram upita koji se prosleđuje
- *BM25* - funkcija rankiranja relevantnosti dokumenata spram upita koja ne koristi neuronsku mrežu za rankiranje i predstavlja varijantu ranije opisanog *TF-IDF* pristupa
- *DPR* - visoko performantan metod za računanje relevantnosti koristeći duboko učenje. Funkcioniše tako što koristi jednu *BERT* mrežu da enkodira dokumente i jednu *BERT* mrežu da enkodira upite.

[7] https://www.muscleandfitness.com/
[8] https://www.myfitnesspal.com/
[9] https://breakingmuscle.com/
[10] https://www.advancedhumanperformance.com/
[11] https://github.com/davejt/exercise
[12] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs
[13] https://ai.google.com/research/NaturalQuestions/

U ovom sistemu iskorišćeni su *TF-IDF*, *BM25* i *DPR* i njihovo poređenje dato je u poglavlju 5.

Baza znanja je baza podataka koja čuva tekst relevantan za domen u vidu dokumenata (koji mogu biti paragrafi) i nudi te podatke pri upitu.

Neke od najčešćih implementacija baze znanja:

- *SQL baza* – relaciona baza, podaci organizovani u tabelama, loše se skalira
- *ElasticSearch* – fleksibilna i brza biblioteka otvorenog koda, čuva dokumente u vidu *inverted index* što omogućava brzu pretragu teksta

U ovom sistemu iskorišćen je *ElasticSearch* zbog načina na koji čuva tekstualne podatke i jer nudi veoma brzu pretragu nad velikim korpusom dokumenata, a takođe je eksperimentisano i sa *SQL* rešenjem koje se pokazalo da ima dosta manju brzinu izvršavanja upita u poređenju sa *ElasticSearch*.

Čitač predstavlja glavnu komponentu sistema za određivanje odgovora na pitanje. Ulazi u ovu komponentu su dokumenti koji su kandidati da sadrže odgovor dobijeni od sakupljača i pitanje a izlaz su mogući odgovori na postavljeno pitanje. Ova komponenta koristi *transformer* arhitekturu najčešće jer takvi modeli ostvaruju *state of the art* rezultate za NLP oblast jer razumeju semantiku i kontekst, u poređenju sa *LSTM* pristupom koji je daleko sporiji i nije zaista bidirekcion.

Za čitač komponentu u implementaciji ovog sistema korišćeni su *RoBERTa* , koji uvodi nasumično maskiranje u *BERT* model [4], i *MiniLM* [3] model koji su upoređeni u poglavlju 5.

MiniLM koristi tehniku destilacije znanja (engl. *knowledge distillation*) koja kompresuje veliki model koji se naziva učitelj (engl. *teacher*) u mali model, koji se naziva učenik (engl. *student*). Glavna ideja je oponašanje slojeva samopažnje, koji su ključna koponenta *transformer* modela. Tačnije, vrši se destilacija modula samopažnje poslednjeg *transformer* sloja u modelu učitelja. Na ovaj način se odbacuju poteškoće koje mogu da nastanu ako se vrši mapiranje svakog sloja učitelja na neki sloj u modelu učenika [3].

Pored navedenih pristupa, napravljen je i osnovni *baseline* pristup za poređenje sa ostalim modelima koji koristi *TF-IDF* metriku za pronalaženje rečenica kandidata, a potom koristi prepoznavanje imenovanih entiteta da pronađe odgovor u okviru tih rečenica. U poglavlju 4 dati su rezultati i poređenje ovih pristupa.

4. EKSPERIMENTALNI REZULTATI I DISKUSIJA

Eksperimentalna evaluacija se vrši radi određivanja performansi klasifikacije svih algoritama za mašinsko učenje u kombinaciji sa tehnikama pretprocesiranja, navedenim u prethodnom poglavlju ovog rada.

Tekstualni podaci, dobijeni sa 3 prethodno pomenuta izvora, nakon obrade i formatiranja prosleđeni su algoritmu za generisanje pitanja i odgovora. Sveukupno, bilo je 20356 *(pitanje, odgovor, kontekst)* trojki u trening skupu i 5124 u test skupu. Pored ovoga, iskorišćen je i *Natural Questions Google Search* skup podataka koji sadrži pitanja otvorenog domena, da bi proverili da li pretrenirani model gubi sposobnost generalizacije nakon što se dotrenira. Kao metrika je pri evaluaciji modela korišćena *F1* mera.

4.1. Evaluacija sistema

Modeli sa kojima je rađena evaluacija čitača su osnovni, jednostavni model, kao i *MiniLM* i *RoBERTa* sa pretreniranim i dotreniranim verzijama. U tabeli 4.1 dati su rezultati evaluacije čitača.

Tabela 1: F1 mera *RoBERTa* i *MiniLM* modela

Podaci / Model	NQ Google General QA skup podataka	Sintetički fitnes skup podataka
Osnovni (*baseline)*	0,05	0,15
MiniLM pretrenirani	0,19	0,39
RoBERTa pretrenirani	0,30	0,42
MiniLM dotrenirani	0,28	0,70
RoBERTa dotrenirani	0,40	0,75

Rezultati su u skladu sa očekivanjima, jer *MiniLM* model ne može da ostvari iste rezultate kao *RoBERTa* koji ima daleko složeniju arhitekturu. Pored toga, modeli dotrenirani na sintetički generisanom skupu podataka o fitnesu ostvarili su znatno bolje rezultate na testnom skupu za pitanja o fitnesu, što je takođe očekivano. Ono što takođe možemo primetiti je da su dotrenirani modeli bolji i na pitanjima otvorenog domena, tj. generalizuju bolje od pretreniranih modela, na osnovu *F1* mera koje su dobijene. Takođe, obzirom da nije pronađen postojeći sistem za fitnes domen, u poređenju sa *XLNet* modelom [6] pretreniranim nad *NewsQA* i *TriviaQA* skupovima gde je ostvarena 0.72 i 0.76 F1 mera, možemo zaključiti da su dotrenirani čitači dovoljno tačni i upotrebljivi.

Dalje, data je evaluacija komponente sakupljača sa *topK* metrikom, obzirom da je to najčešće korišćena metrika za evaluaciju ove komponente. U tabeli 4.2 dati su rezultati koje su postigli modeli *DPR*, *BM25* i *TF-IDF*.

Tabela 2: *TopK* za *TF-IDF*, *BM25* i *DPR* sakupljača

Metrika / Sakupljač	top5	top10	top20
TF-IDF	0,63	0,65	0,68
BM25	0,66	0,67	0,69
DPR	0,82	0,83	0,84

Možemo zaključiti da je za ovaj sistem pogodniji *DPR* jer pored toga što daje bolji rezultat, ovaj pristup će raditi za veći korpus dokumenata. Takođe, možemo zaključiti da su u poređenju sa postojećom implementacijom *BM25* i *DPR* sakupljača [5], koji ostvaruju 0.59 i 0.79 *top20* vrednosti, implementirani sakupljači dovoljno tačni i primenjivi u realnom okruženju.

Pored tačnosti, za ovaj sistem je bilo bitno i vreme izvršavanja nad većim korpusom dokumenata. Brzina izvršavanja sistema sa kraja na kraj koji uključuje i čitač i sakupljač komponentu data je u tabeli 4.3.

Tabela 3: *Performanse sistema sa kraja na kraj*

Model \ Vreme	Prosečno vreme izvršavanja za pitanje u sekundama
MiniLM + TF-IDF	0,55
MiniLM + DPR	0,46
RoBERTa + TF-IDF	1,27
RoBERTa + DPR	1,11

Možemo zaključiti da je *MiniLM* najpogodniji u smislu brzine. Ipak, ako brzina nije najveći prioritet sistema, *RoBERTa* i *DPR* sistem je bolji izbor zbog tačnosti. U poređenju sa *Mindstone* sistemom sa kraja na kraj [8] koji je pretrenirani *BERT* i ne koristi destilaciju znanja ali je navodi kao moguće poboljšanje, možemo zaključiti da je implementirano rešenje dovoljno brzo obzirom da je vreme upita za *Mindstone* 0.73s.

4.2. Analiza grešaka

Pored evaluacije pojedinačnih komponenti i rešenja sa kraja na kraj, vršena je i analiza primera u kojima je *MiniLM* lošiji od *RoBERTa* modela. Neki od ovih primera su pitanja „*What is a chinup*?“, gde *MiniLM* daje odgovor „*bringing the chin up through space*“ ili pitanje „*How do I get stronger*?“ na koje se dobija odgovor „*stronger muscles will improve* posture“, gde se može zaključiti da MiniLM prepoznaje da je u pitanju ista reč u odgovoru, ali ne razume kontekst u potpunosti.

Takođe, slučajevi u kojima *RoBERTa* model greši su pitanja kao što su „How long should I recover from exercising?“ gde je odgovor „two to three minutes between exercises“ ili „How much carbs should I eat?“ gde je odgovor „Eating carbs and fats will make you nervous.“ što pokazuje da iako dobro razume kontekst, ima probleme u slučajevima u kojima je potreban dodatni kontekst tj. konverzacija sa korisnikom. U nastavku dat je zaključak o implementiranom sistemu.

5. ZAKLJUČAK

U ovom radu, implementirano je rešenje problema odgovora na pitanja gde se na osnovu pitanja kao ulaza parsira baza znanja, odnosno dokumenti sa tekstualnim podacima za fitnes i nutricionizam i pronalazi deo teksta u kom se nalazi odgovor na postavljeno pitanje. Implementirana je čitač sakupljač arhitektura koja je vrlo zastupljena u rešavanju problema odgovora na pitanja [5].
Za čitač komponentu isprobani su *MiniLM* i *RoBERTa* modeli, dok su za sakupljač komponentu implementirani *TF-IDF*, *BM25* i *DRP* pristupi. Da bi implementirano rešenje bilo što performantnije, pažljivo je birana sakupljač komponenta kako bi pretraga nad velikog korpusa dokumenata bila dovoljno brza.
Postojao je problem nedostatka podataka za fitnes domen, koji je rešen generisanjem sintetičkog skupa podataka.
Na kraju, pažljivim odabirom i optimizacijom pojedinih delova arhitekture ovog sistema dobijeni su istrenirani modeli koji su u stanju da odgovore na pitanja o fitnesu i nutricionizmu sa visokom tačnošću, a pritom da generalizuju bolje od pretreniranih. Kroz eksperimente i poređenje sa drugim modelima, potvrđeno je da je pristup opisan u ovom radu za rešavanje problema odgovora na pitanja primenljiv.

Moguće su dodatne optimizacije vremena izvršavanja predloženog sistema, tako što će se pažljivo odabrati čitač komponenta koja je računski najzahtevnija u ovakvom sistemu. Pažljiv odabir uređaja na kom će se sistem izvršavati može bitno uticati na brzinu.

Modeli koji su korišćeni u ovom sistemu su dizajnirani za paralelno procesiranje grafičkih kartica. Pored toga, pažljiv odabir parametara kao što su broj dokumenata kandidata koje vraća sakupljač ili broj rezultata iz čitača mogu takođe uticati na brzinu izvršavanja.

6. LITERATURA

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

[2] Lewis, P., Denoyer, L., Riedel, S. (2019). Unsupervised Question Answering by Cloze Translation. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.

[3] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, Ming Zhou. (2020). MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers.

[4] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, & Veselin Stoyanov. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach

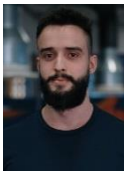
[5] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, Wen-tau Yih. (2020). Dense Passage Retrieval for Open-Domain Question Answering.

[6] Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. "Xlnet: Generalized autoregressive pretraining for language understanding." arXiv preprint arXiv:1906.08237 (2019).

[7] Victor Sanh, Lysandre Debut, Julien Chaumond, & Thomas Wolf. (2020). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.

[8] Sina J. Semnani, & Manish Pandey. (2020). Revisiting the Open-Domain Question Answering Pipeline.

Kratka biografija:



Sava Katić rođen je u Novom Sadu 21. februara 1997. godine. Master rad na Fakultetu tehničkih nauka, oblast Elektrotehnika i računarstvo – Softversko inženjerstvo i informacione tehnologije odbranio je 2021. godine.

Kontakt: savakatic555@gmail.com

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.41
DOI: https://doi.org/10.24867/16BE06Hodzic

RAZVOJ APLIKACIJE ZA PODELU GRAFA PRIMENOM UČENJA SA PODSTICAJEM

DEVELOPMENT OF AN APPLICATION FOR GRAPH PARITIONING USING REINFORCEMENT LEARNING

Maja Hodžić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – *Ovaj rad obrađuje podelu grafa korišćenjem jednog od algoritama učenja sa podsticajem. Grafovi se često koriste kao apstrakcije prilikom modeliranja problema. Podela grafa na manje delove jedna je od osnovnih algoritamskih operacija. Cilj ovog rada je prikaz implementacije algoritma učenja sa podsticajem za podelu grafa na dve particije.*

Ključne reči: *Podela grafa, Učenje sa podsticajem*

Abstract – *This paper deals with the division of graphs using one of the incentive learning algorithms. Graphs are often used as abstractions when modeling problems. Cutting graphs into smaller pieces is one of the basic algorithmic operations. The aim of this paper is to present the implementation of a learning algorithm with an incentive to divide a graph into two partitions.*

Keywords: *Graph partitioning, Reinforcement learning*

1. UVOD

Teorija grafova je oblast matematike veoma zastupljena u informatici. Česta je upotreba grafova za opis modela i struktura podataka, tj. njima se mogu predstaviti i vizualizovati mnogi problemi. Grafovi su sposobni da predstave složene podatke prisutne u različitim domenima. Za lakšu i bržu analizu grafova koristimo particioniranje (podelu). Postoje mnogobrojni algoritmi za ovu podelu a poslednjih godina je sve veće interesovanje za korišćenje tehnika mašinskog učenja za rešenje ovog problema.

Automatsko rezonovanje i mašinsko učenje imaju važne uloge u veštačkoj inteligenciji. Metode zasnovane na logici, koje se razvijaju u okviru automatskog rezonovanja pogodne su u slučajevima u kojima je problem moguće precizno matematički definisati.

Obično se radi o problemima koje čovek može relativno lako da formuliše, ali ih vrlo teško rešava i u kojima nisu prihvatljiva pogrešna rešenja. Sa druge strane, mašinsko učenje je posebno pogodno upravo za suprotnu vrstu problema - probleme koje čovek ne može lako ni da definiše, iako neke od njih čak vrlo lako rešava i u kojima je prihvatljiva povremena greška. Primer takvog problema može da bude prepoznavanje lica, vrste, oblika na slikama... Za ljude je ovo neobično nazivati problemom i govoriti o njegovom rešavanju, ali ako pokušamo da taj problem precizno definišemo naletećemo na mnoštvo problema.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Darko Čapko, vanr. prof.

Zato se ovakvim problemima ne pristupa automatskim rezonovanjem već mašinskim učenjem [2], koje poput ljudi može vrlo uspešno da se nosi sa ovakvim problemom.

Obično se identifikuju tri grupe problema mašinskog učenja: to su problemi *nadgledanog učenja* (*eng. supervised learning*), problemi *nenadgledanog učenja* (*eng. unsupervised learning*) i problemi *učenja sa podsticajem* (*eng. reinforcement learning*). Ovaj rad odnosiće se upravo na *učenje sa podsticajem*, odnosno na podelu grafa korišćenjem jednog od algoritama *učenja sa podsticajem*. Grafovi se često koriste kao apstrakcije prilikom modeliranja problema. Rezanje grafa na manje komade jedna je od osnovnih algoritamskih operacija. Pojavom sve većih primera u aplikacijama kao što su naučna simulacija, društvene mreže ili putne mreže, particioniranje grafova (*graph partitioning*) stoga postaje sve važnije, mnogostranije i izazovnije.

Cilj ovog rada je prikaz implementacije algoritma *učenja sa podsticajem* za podelu grafa na dve particije približno jednake veličine.

2. UČENJE SA PODSTICAJEM

Postoje tri tipa mašinskog učenja, u odnosu na prirodu problema koji se rešava pomoću učenja:

- Nadgledano učenje
- Nenadgledano učenje
- Učenje sa podsticajem

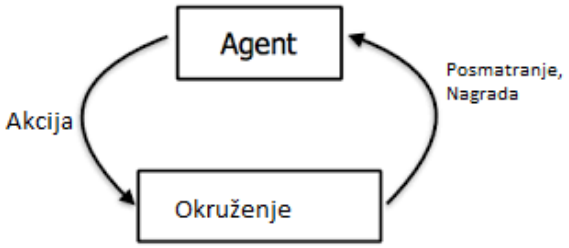
Pretpostavlja se da postoji agent koji prati tekuće stanje okruženja i ima mogućnost da izvršava određene akcije. Na osnovu izvršene akcije dobija određene numeričke vrednosti (nagrade), a cilj je da se kao ishod učenja prođe kroz aktivnosti koje dovode do maksimalne (ili dovoljno visoke) ukupne nagrade. Ključna pretpostavka kod ovog vida učenja je da nije poznato koja od preduzetih akcija je bila prava u datom kontekstu. Postoje pozitivne i negativne nagrade, a agent koristi signale za korekciju svog ponašanja. Ovaj vid učenja ima primenu u robotici, industrijskoj automatizaciji, sistemima za trening, obradu podataka, itd.

2.1. Teorijske osnove učenja sa podsticajem

Cilj algoritama učenja sa podsticajem je pronaći najbolju moguću akciju koju treba preduzeti u određenoj situaciji. Ova vrsta mašinskog učenja može naučiti da postigne cilj u neizvesnim i složenim okruženjima.

2.1.1 Osnovni pojmovi

Teorijski okvir učenja sa podsticajem se opisuje Markov-ovim procesima odlučivanja (*eng. Markov decision processes*), ili skraćeno MDP [4].



Slika 1. *Agent – Okruženje* [4]

Gradivni elementi učenja sa podsticajem su agent koji se kreće, akcija (radnja) koju agent preduzme, nagrada koju stiče, okruženje u kom je agent i stanje u kom se agent nalazi (na tačno određenom mestu u okruženju). U svakom koraku agent izvrši akciju At primi stanje okruženja (sistema) Ot i primi nagradu/kaznu Rt. U isto vreme okruženje primi akciju At, šalje stanje okruženja Ot i šalje nagradu Rt. Agent izvršava akciju na osnovu posmatranja okruženja i nagrade. Agent je algoritam. Stanje je informacija koja se koristi za određivanje sledećeg koraka. Stanje može da gleda samo poslednje u istoriji, da ga prethodno ne zanima.

Istorija je skup posmatranja, akcija i nagrada prikazana u izrazu 1:

$$H_t = A_1 O_1 R_1, \dots, A_t O_t R_t \quad (1)$$

Stanje St je funkcija istorije H_t:

$$S_t = f(H_t) \quad (2)$$

Stanje na osnovu prethodnog stanja je stanje Markov-a ako bi takvo bilo i na osnovu svih prethodnih. Stanje okruženja je stanje Markov-a [4].

$$P[S_{t+1}|S_t] = P\,[S_{t+1}|S_1, \dots, S_t] \quad (3)$$

Iz izraza 3 vidimo da bi stanje S_{t+1} na osnovu prethodnog stanja S_t bilo isto i na osnovu svih prethodnih. Agent i okruženje interaguju u diskretnim trenucima $t = 0, 1, \dots$ U svakom trenutku t, agent opaža stanje okruženja S_t iz konačnog skupa stanja S i preduzima akciju A_t iz konačnog skupa dopustivih akcija $A(s)$ u konkretnom stanju s. Pritom, dobija nagradu R_{t+1} iz konačnog skupa nagrada R i prelazi u novo stanje S_{t+1}. MDP i agent zajedno proizvode putanju: $S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \dots$

Osnovno svojstvo MDP-a je Markov-o svojstvo, da novo stanje i nagrada zavise samo od prethodnog stanja i preduzete akcije, a ne od cele istorije procesa.

Uloga nagrada je da na implicitan način definišu cilj agenta. Treba da budu tako izabrane da maksimizujući nagradu, što je agentov cilj, agent istovremeno obavlja posao koji želimo da obavi. Nagrada je način da agentu saopštimo šta treba da uradi, a ne kako to treba da uradi. Cilj agenta je da maksimizuje dugoročnu nagradu. Dugoročna nagrada G_t u trenutku t u odnosu na neku putanju definišemo na sledeći način:

$$G_t = R_{t+1} + R_{t+2} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (4)$$

pri čemu je ɣ metaparametar umanjenja (*eng. discount*) za koji važi $0 \leq \gamma \leq 1$.

3. PODELA GRAFA

Particioniranje tj. podela grafa je smanjenje grafa na manji graf podelom njegovog skupa čvorova u međusobno isključujuće grupe. Rubovi originalnog grafa koji se ukrštaju između grupa proizvešće grane u particioniranom grafu. Ako je broj rezultirajućih grana mali u poređenju sa originalnim grafom, tada će particionirani graf možda biti pogodniji za analizu i rešavanje problema od originala. Pronalaženje particije koja pojednostavljuje analizu grafova je težak problem. Iizmeđu ostalog ima primenu u naučnom računanju, raspoređivanju zadataka u višeprocesorskim računarima...

Dva uobičajena primera podele grafova su problemi sa minimalnim rezanjem i maksimalnim rezanjem. Analiza grafova je ključ obrade velikih grafova: podela velikog grafa na podgrafove i distribucija podgrafa.

Dat je graf $G = (V, E)$ gde je V skup čvorova, a E skup grana, k uravnotežena podela grafa deli ga na k podgrafova

$$\frac{|E|}{\mathrm{k}\,(1+\mathrm{e})} \quad (5)$$

gde je u izrazu 5 k broj particija, a $e > 0$ je neuravnoteženi odnos.

Nasumični i Hash algoritmi za particioniranje izuzetno su loši u pozicioniranju i odsecanju. Učenje sa podsticajem je klasa algoritama mašinskog učenja inspirisanog principom stimulus-odgovor.

3.2. Revolver algoritam

Revolver [3] je aplikacija učenja sa podsticajem (*Reinforcement Learning)* za podelu (particioniranje) grafa. Algoritam koristi *Learning Automata* i *Label Propagation* za particioniranje grafova.

Pri obuci *Learning Automata* akcije se biraju pomoću vektora verovatnoće P. Za svaku particiju čvora v daje se rezultat ($\psi(v)$). Čvorovi migriraju na svoju kandidatsku (novu) particiju sa verovatnoćom migracije.

Signal nagrade R izračunava se na sledeći način: signal nagrade ako $\psi(v)$ ima najveći rezultat ili ima pozitivnu verovatnoću migracije, kazneni signal suprotno. Vektor verovatnoće se ažurira uzimajući u obzir izračunati signal.

LA pomaže *Revolver*-u da proizvede lokalizovane particije (lokalitet) i uravnotežene particije (skalabilnost). *Revolver* je paralelni algoritam za particioniranje sposoban za particioniranje velikih grafova na jednoj mašini sa zajedničkom memorijom. *K*oristi asinhroni okvir za obradu, koji koristi *Reinforcement learning* i *Label propagation* za prilagodljivo razdvajanje grafa. Pored toga, usvaja vertikalno centrični prikaz grafa gde svakom čvoru je dodeljen autonomni agent odgovoran za odabir odgovarajuće particije, distribuirajući time računanje u svim čvorovima.

Otud normalizovani *LP*, koji se sastoji od normalizovanog ponderisanja τ i kaznenog π definisanog na sledeći način [3]:

$$Score\,(v,l) = \tau\,(v,l) + \frac{\pi(l)}{2} \quad (6)$$

$$\tau\,(v,l) = \sum_{u \in N_{(v)}} \frac{\varpi\,(u,v) \cdot \delta(\psi b(u),l)}{\sum_{u \in N_{(v)}} \varpi\,(u,v)} \quad (7)$$

$$\pi(l) \frac{1-(\frac{b(l)}{C})}{\sum_{i=1}^{k} 1-(\frac{b(l_i)}{C})} \quad (8)$$

U *Revolveru,* novi normalizovani algoritam *LP* ekstrapoliran je da bi formirao ciljnu funkciju koja stvara pondere koji izražavaju adekvatnost dodeljenih particija pomoću *LA*. Štaviše, *Revolver* deli graf na vertekscentrični način. Konkretno, svaki čvor izvlači informacije iz svojih susednih čvorova da bi izračunao rezultat za svaku particiju, pre nego što vrati obračunate ocene (kao težine). Potom se nagrade izračunavaju u svakom čvoru na osnovu akumuliranih težina prikupljenih od svih suseda čvorova. Konačno, verovatnoće akcija povezanih sa particijama se ažuriraju koristeći težine i nagrade u skladu s tim.

Learning Automata (LA) je potklasa učenja sa podsticajem algoritma koji bira svoje nove radnje koristeći prethodno iskustvo sa određenim sredinama.

Label Propagation (*LP*) je polunadgledano mašinsko učenje, algoritam koji dodeljuje oznake velikom skupu neobeleženih podataka koristeci malu količinu označenih podataka.

Suštinska ideja je podela grafa *G = (V, E)* sa *LA = (A, P, R)* na *k* particija

- Mreža *LA*-a analogna *G*-u je stvorena gde je |V| = |LA|
- *LA* je dodeljen svakom čvoru *v*
- Susedni *LA* mogu se ispitati pomoću skupa *E*.
- Skup akcija *A* je isti kao skup dostupnih particija, *|A| = k*
- Vektor verovatnoće *P* je inicijalizovan sa 1 / *k*
- Skup nagrada *R* izračunava se pomoću *LP*

Pri obuci *Learning Automata* akcije se biraju pomoću vektora verovatnoće *P*. Za svaku particiju čvora *v* daje se rezultat *(ψ(v))*. Čvorovi migriraju na svoju kandidatsku (novu) particiju sa verovatnoćom migracije.
Signal nagrade *R* izračunava se na sledeći način: signal nagrade ako *ψ(v)* ima najveći rezultat ili ima pozitivnu verovatnoću migracije, kazneni signal u suprotnom. Vektor verovatnoće se ažurira uzimajući u obzir izračunati signal. Ukoliko je *R* signal nagrada verovatnoće se azuriraju pomocu izraza:

$$P_j\,(n\ +\ 1) = \begin{cases} P_j(n) + \alpha \cdot w_j(n)\left(1 - P_j(n)\right) & j\ = i \\ P_j(n)\left(1 - \alpha \cdot w_j(n)\right) & j \neq i \end{cases} \quad (9)$$

U slucaju da je *R* kazna verovatnoće se ažuriraju sa kaznom:

$$P_j\,(n\ +\ 1) = \begin{cases} P_j(n)(1 - \beta \cdot w_j(n))) & j = i \\ P_j(n)\big(1 - (\beta \cdot w_j(n))\big) + \frac{\beta}{m-1} & j \neq i \end{cases} \quad (10)$$

4. IMPLEMENTACIJA REVOLVER ALGORITMA

Kod implementiran u radu može se predstaviti pseudo kodom na sledeći način (11):
Input: graf predstavljen matricom susedstva
Output: podeljen graf na dve particije P_1 i P_2

1 Inicijalizacija α i β parametara nagrade i kazne
2 Random odabir početnog čvora od kog se deli
3 **repeat**
4 Biranje sledećeg kandidata na osnovu liste verovatnoće izbora (RoulleteWheel)
5 Za izabrani čvor izračunati Score funkciju (7)
6 **if** Score > 0 i verovatnoća > 0
7 Postaviti čvor u posmatranu particiju, ažurirati verovatnoće sa nagradom (10)
8 **else** Čvor ne seli u particiju, verovatnoće se ažuriraju sa kaznom (11)
9 **until** veličina particije <= polovina grafa
10 **repeat**
11 **for each** Čvor in P_1 do
12 P = veze sa čvorovima iz druge particije - veze sa čvorovima iz trenutne particije
13 **end for each**
14 **for each** Čvor in P_2 do
15 D = veze sa čvorovima iz druge particije - veze sa čvorovima iz trenutne particije
16 **end for each**
17 Za čvorove $V_i \in P_1$ & $V_j \in P_2$ izračunati J = P+D, J_{ij} = max(J) □ V_i i V_j menjaju particije
18 **until** (max(J) <=0)

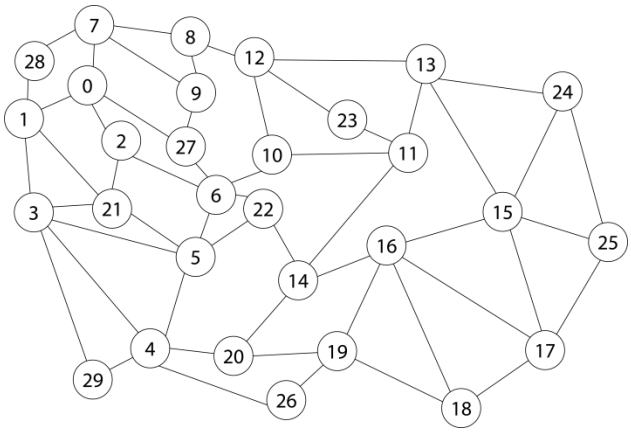
U prvom delu koda (linija 2 do 9) koristi se princip *Revolver* algoritma za početnu (inicijalnu) podelu grafa. Na slučajan način se izabere početni čvor, zatim se za početni čvor i sve naredne lista verovatnoće izbora njegovih suseda prosleđuje na Roulete-selekcija (Roulete Wheel funkciju), gde čvor sa većom verovatnoćom ima veću šansu da bude izabran. Za izabrani čvor se računa vrednost Score funkcije po izrazu 6. Prvi član izraza računa vrednost *τ (v, l)*, po izrazu 7, koja daje odnos koliko je susednih čvorova čvora *v* u particiji za koju računamo u odnosu na broj u drugoj particiji.

Kronerova konstanta *δ* ima vrednost *1* ukoliko je čvor u razmatranoj particiji. Treba naglasiti da se rad odnosi na bestežinske grafove tj. težine svih grana imaju vredost *1*. Dakle, rezultat računanja *τ (v, l)* je razlomak gde je u brojiocu broj suseda koji se nalaze u particiji za koju računamo dok je u imeniocu ukupan broj suseda čvora *v*. Iz izraza 8, gde je *b(l)* trenutno opterećenje particije a *C* ukupan kapacitet, dobijamo drugi član *π(l)* koji proizvodi kazne za particije i normalizuje se na osnovu ukupnog opterećenja sistema. Ukoliko je vrednost Score-a veća od nule vrednost nagrade je *1*, čvor se seli u razmatranu particiju i verovatnoće se ažuriraju pomoću izraza 9. U suprotnom čvor ne seli, a verovatnoće se ažuriraju prema izrazu 10.

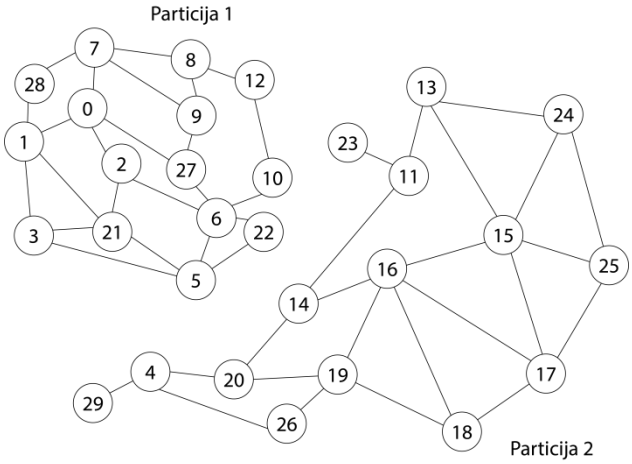
Nakon inicijalne podele naveden je iterativni postupak za unapređenje podele. Razmenjuju se čvorovi između particija kako bi se smanjio broj grana između čvorova koji se nalaze u različitim particijama. Implementiran je po uzoru na KL algoritam (Kernighan i Lin) [5]. Za svaki čvor proveravaju se njegovi susedi tj. pronalaze se čvorovi čijom se razmenom između particija smanjuje broj presečenih grana uz zadovoljavanje ograničenja o veličini particija. Uvode se pojmovi spoljašnjih i unutrašnjih veza - odnos između njih definiše da li čvor već pripada odgovarajućoj particiji ili ga treba prebaciti. Ukoliko čvor ima više spoljašnjih veza nego unutrašnjih to jest suseda u drugoj particiji on postaje kandidat za prebacivanje. Kada se nađe čvor u particiji *1* i drugi u particiji 2 (pogodni za prebacivanje) praktično im se zamene mesta. Ažurira se stanje u particijama i traže se drugi kandidati za razmenu. Ukoliko ih više ne nalazimo postupak se prekida.

5. REZULTATI TESTIRANJA

Navedeni kod testiran je na više grafova. Svi grafovi su neusmereni i bestežinski generisani na slučajan način [6]. Rezultati podele prilično zavise od inicijalne podele, samim tim i početnog čvora od kog se deli. Za inicijalnu podelu se koristi Revolver algoritam, te se inicijalna podela vrlo malo ili nikako razlikuje od finalne podele. Što govori u prilog korišćenju samog Revolver algoritma. Na slikama 2 i 3 prikazan je primer podele grafa.



Slika 2: *Graf za podelu*



Slika 3: *Podeljen graf*

6. ZAKLJUČAK

U ovom radu prikazana je teorijska analiza, razvoj aplikacije (implementacija algoritma) i dobijeni rezultati pri pokretanju aplikacije. Algoritam je kodiran u c# programskom jeziku i pri pokretanju daje vrlo dobre rezultate na različitim grafovima koji su korišteni za testiranje. Konkretno, Revolver algoritam se vrlo dobro pokazao pri inicijalnoj podeli koju u nekim slučajevima nije potrebno korigovati. Revolver čuva lokalnu podelu bez žrtvovanja ravnoteže opterećenja.

7. LITERATURA

[1] Alan Turing, Computing Machinery and Intelligence, (1950.)

[2] Mladen Nikolić, Anđelka Zečević, Mašinsko učenje, Beograd (2019.)

[3] Mohammad Hasanzadeh Mofrad, Rami Melhem and Mohammad Hammoud, Partitioning Graphs for the Cloud using - Reinforcement Learning, Pittsburgh USA https://arxiv.org/pdf/1907.06768.pdf (17.07.2019.)

[4] Introduction to Reinforcement Learning with David Silver (https://deepmind.com/learning-resources/-introduction-reinforcement-learning-david-silver), (08.2021.)

[5] Darko Čapko, Optimalna podela velikih modelapodataka u okviru nadzorno-upravljačkih elektroenergetskih sistema, Novi Sad (2012.)

[6] Graph online (https://graphonline.ru/en/) (09.2021.)

Kratka biografija:

Maja Hodžić, rođena 29.05.1987. godine u Dubrovniku, Republika Hrvatska. Diplomski rad (osnovne akademske studije) na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika - usmerenje Automatika i upravljanje sistemima – odbranila je 2018. godine

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.42:004.738.12
DOI: https://doi.org/10.24867/16BE07Antesevic

АНТИФОРЕНЗИКА МОБИЛНИХ УРЕЂАЈА

MOBILE ANTI-FORENSICS

Светлана Антешевић, *Факултет техничких наука, Нови Сад*

Област: ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – *У овом раду обрађена је антифорензика мобилних уређаја. Разматране су и анализиране технике везане за антифорензику мобилних уређаја и представљени су и детаљно описани алати који имплементирају те технике. Акценат је стављен на обраду техника и алата везаних за iOS и Android оперативни систем због њихове доминантности на тржишту.*

Кључне речи: *дигитална форензика, антифорензика, iOS, Android, технике, алати*

Abstract – *This paper addresses the anti-forensics of mobile devices. Techniques related to anti-forensics of mobile devices are considered and analyzed, and tools that implement these techniques are presented and described in detail. Emphasis is placed on the processing of techniques and tools related to the iOS and Android operating systems due to their dominance in the market.*

Keywords: *digital forensics, anti-forensics, Android, iOS, techniques, tools*

1. УВОД

Током протеклих година, сведоци смо све веће распрострањености мобилних уређаја широм света. Не само да број корисника расте, него се и уређаји све више користе за низ свакодневних активности. То указује на све већи број нелегалних радњи који се обављају путем мобилних уређаја. Такође, са жељом да се сакрију, уклоне или униште подаци који могу бити мета или разлог неког напада, спроводе се разне активности како би се докази који су ускладиштени у мобилном уређају уништили, компромитовали или избрисали тј. како би се онемогућила форензичка истрага. Стога, циљ овог рада јесте да се истраже и презентују антифорензичке технике и алати код мобилних уређаја како би се потпомогао процес форензичке истраге мобилних уређаја, дала ближа слика антифорензичких метода над мобилним уређајима и како би се открили или ближе представиле легалне и нелегалне радње над мобилним уређајима.

Друго поглавље овог рада бави се прегледом области из дигиталне форензике, објашњени су основни појмови везани за форензику мобилних уређаја, архитектуру и безбедносне механизме самих мобилних уређаја, појмове везане за дигиталну антифорензику и антифорензику мобилних уређаја. У трећем поглављу размотрене су и детаљно анализиране опште технике код дигиталне антифорензике као и антифорензичке технике код мобилних уређаја са *iOS* и *Android* оперативним системом. Кроз четврто поглавље дат је приказ тренутно коришћених алата који се користе ради примене антифорензичких метода током процеса истраге или ради детектовања спроведених антифорензичких метода над мобилним уређајем.

НАПОМЕНА:
Овај рад проистекао је из мастер рада чији ментор је био др Стеван Гостојић, ванр. проф.

2. ПРЕГЛЕД ОБЛАСТИ

2.1. Дигитална форензика

Дигитална форензика је употреба научно изведених и проверених метода за очување, прикупљање, проверу ваљаности, идентификацију, анализу, тумачење, документацију и презентацију дигиталних доказа изведених из дигиталних извора у сврху олакшавања или унапређења реконструкције догађаја за које се утврди да су кривични или да помажу да предвиди неовлашћене радње за које се показало да ометају планиране операције [1]. Разноликост различитих уређаја, оперативних система и софтвера довела је до потребе да се дигитална форензика подели на више подобласти у зависности од врсте уређаја. Области дигиталне форензике представљају: форензика рачунара, форензика рачунарских мрежа, форензика мобилних уређаја, форензика мултимедијалних записа и остало (нпр. уграђени системи, IoT, облак итд.). Дигитална форензичка истрага се састоји од шест фаза, а то су: идентификација, прикупљање, чување, прегледање, анализа и презентација доказа.

2.2. Форензика мобилних уређаја

Форензика мобилних уређаја је област дигиталне форензике чији предмет су докази ускладиштени на мобилним уређајима или преношени преко целуларне мреже [2]. Целуларна (мобилна) мрежа представља јавну комуникациону мобилну мрежу за пренос говора, текстуалних порука и података.

Мобилни уређаји су незаобилазно средство у свакодневном животу појединаца. Намењени за комуникацију и коришћење разних сервиса и са собом носе мноштво података о корисницима. Као такви, постају главна мета разних злоупотреба и криминалних активности због података који су у њима похрањени. Како на тржишту последњих година преовлађују два оперативна система *iOS* и *Android*, у наставку ће бити разматрана ова два оперативна система.

iOS је мобилни оперативни систем развијен и дистрибуиран од стране *Apple* компаније. Универзалан је оперативни систем за све *Apple* уређаје као што су

Apple TV, iPad, iPod Touch и *iPhone*. На тржишту мобилних уређаја представља удео од отприлике 26% [3]. Управља хардвером уређаја и пружа технологије потребне за имплементацију нативних апликација. iOS архитектура се састоји од четири слоја: *cocoa touch, media layer, core services layer* и *core OS layer.*

Android је оперативни систем за мобилне уређаје. Представља пројекат отвореног изворног кода. На тржишту мобилних уређаја заузима највећи удео, око 70% [3]. Постоји више дистрибуција *Android* платформе од стране различитих произвођача (*LG, Samsung, Huawei* итд.) који користе различите хардверске архитектуре и различите продавнице апликација. *Android* архитектура се састоји од различитих слојева [4]: *Linux Kernel, hardware abstraction layer, android runtime, native C/C++ library u java API framework.*

Мобилна безбедност или тачније безбедност мобилних уређаја, је области информационе безбедности која се бави заштитом паметних телефона, таблета и преносних рачунара од претњи. Посебно забрињава сигурност личних и пословних података који се сада чувају на паметним телефонима. Постоје добре праксе које треба поштовати на свим нивоима, од дизајна до употребе, преко развоја оперативних система, слојева софтвера и апликација за преузимање [5].

iOS је дизајниран са безбедношћу у својој основи и са више слојева сигурности. Хардверске функције ниског нивоа штите од напада злонамерног софтвера, а функције високог нивоа оперативног система спречавају неовлашћену употребу [6]. *Android* је дизајниран са посебним фокусом на безбедност. *Android* као платформа нуди и примењује одређене функције које штите корисничке податке присутне на мобилном уређају кроз вишеслојну сигурност.

Постоје одређене подразумеване поставке безбедности које ће заштитити корисника и одређене понуде које развојна заједница може да искористи за изградњу сигурних апликација.

2.3. Дигитална антифорензика

У овом раду ћемо се служити дефиницијом да је антифорензика процес компромитовања доступности, поузданости и корисности доказа током процеса форензичке истраге [7]. То су покушаји да се негативно утиче на постојање, количину и/или квалитет доказа са места злочина или да се отежа или онемогући прегледање и анализа доказа [8].

Дигитална антифорензика има широк спектар области примене и сврхе. Постоје одређени индикатори који дигиталним форензичарима могу помоћи у откривању таквих намера.

Неки од њих су: шифровани подаци на уређају, докази о брисању артефаката, докази о скривању трагова, присуство антифорензичких алата и слично.

3. ТЕХНИКЕ АНТИФОРЕНЗИКЕ

Антифорензичке технике постају велика препрека за дигиталну форензичку заједницу. Многе од техника представљају значајне изазове и могу онемогућити опоравак података користећи типичне форензичке праксе.

3.1. Антифорензичке технике

Генерално идентификоване антифорензичке технике описане су у наставку овог одељка.

Скривање доказа (енг. *hiding evidence*) представља радње које су преузете ради смањења или чак поништавања доступности доказа током форензичке истраге.

Шифровање је процес кодирања информација у коме се отворени текст трансформише у шифрат тако да само ауторизоване особе, које поседују криптографски кључ, могу да дешифрују шифрат.

Стеганографија је процес скривања једне поруке у другој поруци која може бити у облику текста, фотографије, видеа или аудио снимка.

Контрацепција података укључује антифорензичке технике које не производе доказе или производе мало дигиталних доказа. Оне се односе на посредно позивање системских функција, инјекцију удаљених библиотека, директну манипулацију објектима језгра оперативног система и преносне апликације.

Манипулација чврстим диском односи се на скривање партиција на диску.

Манипулација системом датотека представља скривање датотека или директоријума у систему датотека.

Скривање у меморији је скривање података у радној меморији уређаја.

Скривање података на мрежи укључује антифорензичке технике као што су *proxy* сервери, виртуалне приватне мреже (*virtual private network - VPN*) итд.

Брисање артефаката (енг. *artifact wiping*): Укључује антифорензичке технике које свесно уништавају доказе како би били неупотребљиви током истражног процеса. *Техника размагнетисања диска* је процес демагнетизирања ради уништавања података похрањених на чврстим дисковима и другим магнетним медијумима. *Брисање чврстог диск*а је потпуно брисање свих података са диска. *Уништавање датотека* је процес који укључује сигурно брисање рачунарске датотеке. *Уништавање метаподатака датотека* је брисање информација о другим подацима. *Брисање регистара* подразумева брисање кључа или ставке регистратора. *Брисање лог датотека* представља трајно брисање лог датотека са диска. *Брисање спољашњих медијума* представља брисање података на преносивим дисковима попут USB меморијског стика. *Генеричко брисање података* су технике и алати који могу да изврше више од једног типа брисања података.

Скривање трагова (енг. *trail obfuscation*) представљају антифорензичке технике које намерно дезоријентишу и преусмеравају форензичку истрагу.

Промена backbone мреже представља повезивање основне мреже са различитим мрежама и подмрежама ради размене информација између њих.

Лажирање IP адресе подразумева коришћење неколико *IP* адреса ради прикривања стварне *IP* адресе.

Лажирање MAC адресе је мењање хард кодираног броја *MAC* адресе повезаног са својом мрежном картицом.

Манипулација лог датотекама представља мењање лог датотека, временских линија активности и догађаја који су се догодили.

Коришћење proxy сервера представља. коришћење серверске апликације која служи као посредник између клијената који захтева ресурс и сервера који пружа тај ресурс.

Погрешно усмеравање података је преношење истинитих података преко лажних „тајних“ података.

Тројанске команде представљају злонамеран софтвер који је дизајниран да оштети, омета и уради податке.

Напад на форензичке технике и алате (енг. *attacks on forensics methods and tools*) укључује антифорензичке технике које подразумевају нападе на софтверске алате који се користе у форензичким истрагама и на технике које ти алати имплементирају. У ове нападе спадају: упозорења за употребу форензичких алата, технике против реверзног инжењеринга, напад на интегритет форензичког софтвера, напади на интегритет *hash* вредности, напади на интегритет истражитеља и програмски пакери.

Уклањање извора доказа (енг. *eliminating evidence sources*) представља неутрализацију извора доказа. Ова техника се не тиче уништавања већ спречавања стварања доказа.

Фалсификовање доказа (енг. *counterfeiting evidence*) представља стварање лажне верзије доказа који је правилно направљен да носи погрешне информације како би се преусмерио судски поступак.

3.2. Антифорензичке технике Android-а

Неке од антифорензичких технике које су везане за *Android* платформу су описане у овом одељку.

Техника искоришћавања *Android* функција односи се на управљање различитим дозволама за датотеке на нивоу оперативног система омогућавајући тиме спровођење тих функција.

Уништавање доказа ускладиштених на *Android* уређајима односи се на брисање повезаних база података како би се трајно избрисали одговарајући подаци. Углавном се спроводи путем одговарајућих алата за брисање.

Скривање доказа ускладиштених на *Android* уређајима односи се на чување доказа у приватној фасцикли. Приватна фасцикла је директоријум који је недоступан за било које друге апликације. Може служити за складиштење било које врсте информација, које су невидљиве за крајњег корисника и подаци нису приступачни унутар ње.

3.3. Антифорензичке технике iOS-а

Антифорензичке технике код *iOS* уређаја своде се на прикривање, брисање или уметање података како би се отежала форензичка истрага.

Брисање датотека на iOS-у је примена стандардних техника брисања метаподатака и података како би се отежао процес форензичке истраге.

Техника прикривања која побољшава сигурност незаштићених података који мирују на *iOS* уређајима осмишљена да учини садржај нечитким без одговарајућег 256-битног кључа. Инспирисана је шифром са случајним кључем (енг. *one-time pad - OTP*) коју је дизајнирао *Gilbert Vernam 1917.* [9].

4. АЛАТИ

У оквиру овог одељка представљени су алати који се користе за антифорензичку анализу мобилних уређаја. Алати који су истражени сортирани су по техникама које примењују.

4.1. Алати за скривање доказа

Cyphertop је алат за шифровање података који се користи и на *iOS*-у и на *Android*-у. Користи симетрични систем шифровања по блоковима. Поседује скривена складишта за чување шифрованих података, нуди стеганографске функције скривања података, нуди функције потпуног скривеног разговора.

Orbot Proxy with TOR представља бесплатну *proxy* апликацију која користи *TOR* за шифровање интернет саобраћаја.

K-9 Mail sa APG представља апликацију која шифрује е-пошту на *Android* уређају.

EncriptRIGHT представља алат који врши маскирање података на начин који одговара сваком кориснику који је овлашћен да приступи осетљивим подацима и да их заштити од безбедности ради примене одговарајуће приватности података.

StegDroid Alpha је једноставна апликација која се може користити за скривање текстуалних порука унутар аудио датотека, применом стеганографије.

PixelKnot апликација користи стеганографски алгоритам F5 који имплементира матрично кодирање ради побољшања ефикасности уграђивања и користи пермутације како би равномерно распоредио промене по целом стеганограму.

Da Vinci Secret Image преставља апликацију која се користи ради сакривања тајне поруке унутар слике.

Rootkit програм који је написан за *Android* платформу може се активирати путем телефонских позива или *SMS*-а када се инсталира на телефон, дајући криминалцима прикривен и тешко уочљив алат за пребацивање података са телефона или погрешно усмеравање корисника.

4.2. Алати за брисање доказа

Blancco Mobile Solutions омогућава трајно брисање свих података са паметних телефона и таблета који раде на различитим оперативним системима.

BatchPurifier је алат за уклањање скривених података и метаподатака из више датотека.

Неки од алата за брисање податка који трајно ресетују мобилне уређаје су: *BitRaser, Dr.Fone - Data Eraser (Android), Coolmuster Android Eraser, MobiKin Android Data Eraser.*

Vipre Android Security је алат који даје могућност праћења безбедности *Android* уређаја, праћење где се налази уређај и даљинског брисања података са уређаја.

4.3. Алати за детектовање антифорензичких поступака

XRY је софтверски алат компаније *MSAB* за прикупљање дигиталних доказа из мобилних уређаја. Садржи функционалности које могу да детектују злонамерни софтвер, обрисане податке из облака и резервних копија.

Anti-malware SDK for Android може се користити за скенирање злонамерног софтвера у било којој врсти *Android* датотеке.

The Zimperium Mobile Threat Defense (MTD) платформа пружа континуирано праћење и анализу на уређају за откривање мобилних напада у реалном времену.

Oxygen Forensics Mobile је софтвер који омогућава преузимање података са нових уређаја, приступ сигурнијим подацима и обнављања избрисаних записа. Проналази лозинке за шифроване резервне копије и слике уређаја.

MOBILedit Forensic претражује, испитује и извештава о подацима са *GSM/CDMA/PCS* мобилних телефона. Додатне функције овог алата укључују услугу која омогућава приступ бази података *IMEI* за регистрацију и проверу украдених телефона.

5. ЗАКЉУЧАК

Како би се потпомогао процес форензичке истраге мобилних уређаја, дала ближа слика антифорензичких метода над мобилним уређајима и открило антифорензичке радње над мобилним уређајима, у овом раду су анализиране технике и алати који се користе у сврху компромитовања, брисања и уништавања доказа релевантних за форензичку истрагу.

Да би се превазишли проблеми који се јављају у форензици мобилних уређаја, неопходно је спровести даља истраживања из области антифорензике мобилних уређаја и било би пожељно направити базу података која садржи легалне и нелегалне антифорензичке технике над мобилним уређајима и списак алата који имплементирају те технике.

6. ЛИТЕРАТУРА

[1] André Årnes, *Digital Forensics*, First. John Wiley & Sons Ltd, 2018. [Online]. Available: https://lccn.loc.gov/2017004725

[2] Стеван Гостојић, "06 Форензика мобилних уређаја." 2021.

[3] "Mobile Operating System Market Share Worldwide," *StatCounter Global Stats*. https://gs.statcounter.com/os-market-share/mobile/worldwide (accessed Sep. 16, 2021).

[4] "Platform Architecture," *Android Developers*. https://developer.android.com/guide/platform (accessed Sep. 08, 2021).

[5] "Mobile security," *Wikipedia*. Aug. 18, 2021. Accessed: Sep. 08, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Mobile_security&oldid=1039460744

[6] *Read Practical Mobile Forensics Online by Satish Bommisetty, Rohit Tamma, and Heather Mahalik | Books*. Accessed: Sep. 08, 2021. [Online]. Available: https://www.scribd.com/book/272076743/Practical-Mobile-Forensics

[7] "Digital Forensics Explained," *Routledge & CRC Press*. https://www.routledge.com/Digital-Forensics-Explained/Gogolin/p/book/9780367503437 (accessed Sep. 08, 2021).

[8] Стеван Гостојић, "09 Антифорензика." 2021.

[9] C. DOrazio, A. Ariffin, and K.-K. R. Choo, "iOS Anti-forensics: How Can We Securely Conceal, Delete and Insert Data?," in *2014 47th Hawaii International Conference on System Sciences*, Jan. 2014, pp. 4838–4847. doi: 10.1109/HICSS.2014.594.

Кратка биографија:

Светлана Антешевић рођена је у Новом Саду 1997. године. Завршила је основну школу „Мирослав Антић" у Футогу и средњу школу „Светозар Милетић" у Новом Саду. Дипломирала је 2020. године на Факултету техничких наука у Новом Саду, смер Рачунарство и аутоматика, са темом „Имплементација сервисних модула без серверских конфигурација". Исте године је уписала мастер студије на смеру Рачунарство и аутоматика, модул Примењене рачунарске науке и информатика – Електронско пословање.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007:004.056
DOI: https://doi.org/10.24867/16BE08Zecevic

PRIMENA BEZBEDNOSNIH ELEMENATA AZURE PLATFORME U SISTEMIMA ZA ELEKTRONSKA PLAĆANJA

APPLICATION OF SECURITY CONTROLS OF AZURE PLATFORM IN ELECTRONIC PAYMENT SYSTEMS

Helena Zečević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu predstavljeni su problemi i izazovi sistema koji se bave eletktronskim plaćanjima. Predstavljen je jedan koncept sistema kroz koji su istražene bezbednosne kontrole i zahtevi nekih od regulativa čije implementacije se smatraju obaveznim. Poseban akcenat stavljen je na mehanizme koje pruža i olakšava Microsoft Azure platforma. Istraženo je više servisa koje ova platforma nudi i opisani su prednosti ali i nedostaci istih.*

Ključne reči: *elektronsko plaćanje, informaciona bezbednost, Azure*

Abstract – *In this paper presented are the problems and challenges faced by the electronic payment systems. One concept of such system is presented here, through which security controls have been investigated along with requirements of several regulations which implementations are considered a must in these systems. In this paper highlighted are the mechanisms which are offered and made easier for implementation by Microsoft Azure platform. Multiple services offered by this platform have been explored and both advantages and downsides have been presented in this paper.*

Keywords: *electronic payment, information security, Azure*

1. UVOD

U današnje vreme sistemi bazirani na elektronskom poslovanju su postali značajan udeo tržišta i uveliko zamenjuju tradicionalne načine poslovanja. Sa razvojem internet bankarstva, odlazak u banku postao je izuzetak, a ne pravilo, a sa razvojem *online* trgovine postalo je moguće dobaviti i najosnovnije potrepštine za život iz udobnosti svog doma. Razvoj ovakvih elektronskih sistema, a naročito *online* trgovine, doveo je do toga da se značajan udeo novca razmenjuje na internetu.

Razvoj sistema za elektronsko poslovanje je povukao za sobom i razvoj sistema za elektronsko plaćanje. Elektronsko plaćanje danas je moguće obaviti na više načina: posredstvom sistema razvijenih od strane banaka putem sistema kao što je *PayPal* [1], ali i razmenom kriptovaluta poput *bitcoin*-a [1].

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Goran Sladić, red. prof.

[1] https://www.paypal.com/

Ono što je zajedničko za sve *online* sisteme jeste pažnja koja se pridaje bezbednosti. Sistemi elektronskog plaćanja upravljaju osetljivim korisničkim podacima poput brojeva računa, brojeva kartica, bezbednosnih kodova na karticama, tajnih ključeva za pristup sistemima poput *PayPal*-a, itd.

S obzirom na prirodu ovih osetljivih podataka, rizik ovakvih sistema je visok, stoga je važno posvetiti posebnu pažnju informacionoj bezbednosti. Informaciona bezbednost se definiše kao zaštita implementirana u informacionom sistemu kako bi se očuvali poverljivost (eng. *Confidentiality*), dostupnost (eng. *Availability*) i integritet (eng. *Integrity*) resursa informacionog sistema [2]. Ova tri pojma poznatija su u informacionoj bezbednosti kao CIA trijada i predstavljaju srž računarske bezbednosti [2].

Zbog ekonomske isplativosti, skalabilnosti i strožijih potreba za dostupnosti servisa, veliki broj aplikacija danas se nalazi na *cloud*-u ili je u procesu integracije. Veliki *cloud* provideri, poput *Microsoft Azure*, takođe nude širok opseg već implementiranih bezbednosnih aspekata i jednostavnu integraciju sa njima, ili olakšavaju implementaciju istih.

Postoje brojni standardi koji se tiču bezbednosti u informacionim i *cloud* sistemima, među kojima su neki od poznatijih ISO-27001, GDPR, HIPAA, PCI DSS, itd. [3].

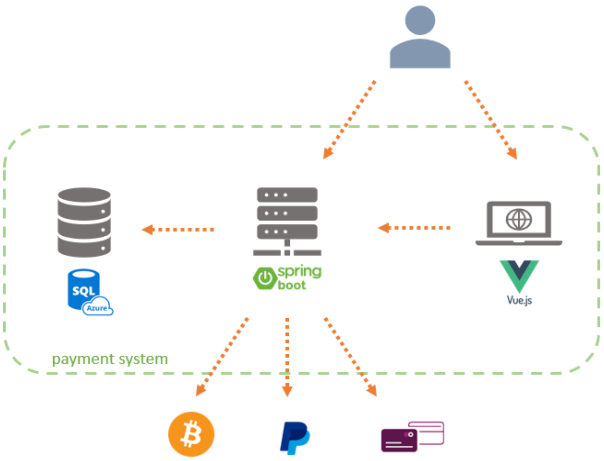
U ovom radu biće izloženi bezbednosni elementi nekih od servisa koje nudi *Azure* platforma, kao i detalji o njihovim prednostima i manama.

2. SPECIFIKACIJA SISTEMA ZA ONLINE PLAĆANJE

Koncept sistema za *online* plaćanje podrazumeva sistem koji korisnicima omogućuje plaćanje posredstvom nekoliko različitih načina plaćanja. Sistem je zamišljen tako da se može proširivati lako novim načinima plaćanja, a u inicijalnoj specifikaciji podrazumeva podršku za plaćanje platnim karticama, putem *PayPal* platforme i *bitcoin* valutom. Sam sistem sastoji se iz više komponenti.

Slika 1 prikazuje arhitekturu i osnovne komponente koncepta sistema za plaćanje koji se sastoji iz: relacione baze podataka, centra sistema za plaćanje, klijentske aplikacije i eksternih servisa za plaćanje.

Korisnici mogu sa sistemom da interaguju direktno koristeći centralnu aplikaciju putem *web* API-a ili posredstvom korisničke *web* aplikacije. Centralna aplikacija izvršenje plaćanja omogućuje putem interakcije sa eksternim servisima za plaćanje, poput *Bitcoin*-a i *PayPal*-a.



Slika 1. *Sistem za plaćanje (uokviren zelenom bojom): relaciona baza podataka (levo), poslovna logika (sredina) i klijentska web aplikacija (desno).*

2.1. Relaciona baza podataka

Podaci koji se koriste u sistemu za plaćanje su po svojoj prirodi strukturirani, stoga se relaciona baza činila kao logičan izbor za skladištenje podataka. Neki od podataka koji se skladište u sistemu su: podaci o prodavcima (eng. *merchants*), podaci o uplatama, detalji o podržanim metodama plaćanja za svakog prodavca, itd.

U finalnoj implementaciji sistema korišćena je *Azure SQL Database* [4] baza podataka, o kojoj će biti više reči kasnije.

2.2. Centar za plaćanje

Centralna aplikacija predstavlja srž sistema za plaćanje. Ona je zadužena za rukovanje svim zahtevima koji pristižu od klijenata koji koriste ovaj sistem za plaćanje.

Korisnici prvog reda ovog sistema za plaćanje zapravo predstavljaju druge sisteme, u najčešćem slučaju *online* trgovine.

Krajnji korisnici ovih trgovina koriste sistem za plaćanje prilikom kupovine proizvoda, međutim njihovi podaci se u ovom sistemu ne skladište i ne postoji potreba za bilo kakvom vrstom registracije ovakvih korisnika.

S druge strane trgovine koje žele da koriste ovaj sistem moraju na neki način da se prijave kako bi bile poznate sistemu, ali i da prilože podatke koji su potrebni za obavljanje transakcija, poput npr. broja bankovnog računa.

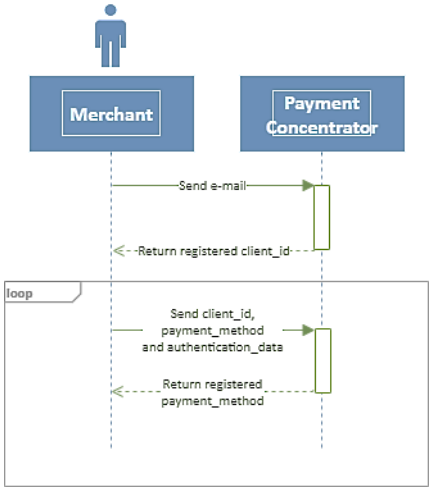
2.2.1 Tok registracije na različite načine plaćanja

Slika 2 prikazuje tok registracije jednog prodavca (eng. *merchant*) na sistem za plaćanje (eng. *payment concentrator*).

Prodavci se najpre registruju na sistem sa *e-mail* adresom, nakon čega dobijaju potvrdu o registraciji u vidu *client_id* oznake.

Nakon toga prodavac je u mogućnosti da se registruje na načine plaćanja koje želi da omogući svojim kupcima, a koji su podržani od strane sistema. Prilikom registracije na željeni način plaćanja potrebno je proslediti svoj *client_id*, odabrani metod, kao i podatke za autentikaciju na željeni metod (npr. API ključ u slučaju *PayPal*-a).

Nakon registracije prodavca, kupci su u mogućnosti da koriste isključivo načine plaćanja koji su podržali prodavci.



Slika 2. *Prikaz toka registracije prodavca na sistem za plaćanje.*

2.2.2 Tok plaćanja

Online plaćanje nije atomična operacija, već je predviđen određen sled koraka, gde u svakom može doći do greške. Sledi opisan primer toka plaćanja putem platnih kartica.

Kupac nakon odabira proizvoda bira način plaćanja od ponuđenih. Podaci se sa sistema prodavca prosleđuju na sistem za plaćanje, među kojima su i URL adrese na koje treba preusmeriti korisnika nakon uspešnog i neuspešnog plaćanja. Podaci se dalje prosleđuju sistemu banke prodavca, koja nakon provere podataka inicijalizuje plaćanje i sistemu vraća URL na koji treba preusmeriti korisnika kako bi se plaćanje realizovalo. Korisnik na novoj stranici unosi podatke iz kartice, koji se prosleđuju banci. Banka proverava podatke i u slučaju da je i kupac u istoj banci ona vrši transfer sredstava, a ukoliko je u drugoj banci, kontaktira asocijaciju koja prosleđuje podatke banci kupca. Nakon izvršenog transfera novca, banka javlja sistemu o rezultatu transakcije, a naš sistem dalje korisnika preusmerava na unapred zadatu stranicu.

2.3. Klijentska aplikacija

Klijentska aplikacija zamišljena je kao *web* aplikacija jednostavnog izgleda. Implementirana je u *Vue.js*[2] okviru za rad. Ovaj okvir za rad je lak za korišćenje i nema velike količine suvišnog koda (eng. *overhead*) kao kod nekih drugih okvira. Aplikacija je vrlo jednostavna i ima mali broj stranica.

3. BEZBEDNOSNI MEHANIZMI AZURE PLATFORME

Jedan od aspekata bezbednosti o kojima posebno treba voditi računa prilikom izrade svakog softverskog sistema jesu osetljivi podaci i njihovo skladištenje. Sistem za plaćanje po svojoj prirodi rukuje osetljivim podacima i gubitak podataka može da dovede do značajne materijalne štete. Stoga je važno posvetiti pažnju očuvanju poverljivosti i integriteta osetljivih podataka i mehanizama koji mogu dovesti do njihovog kompromitovanja.

3.1 GDPR i Azure

GDPR regulativa doneta je kako bi se regulisalo prikupljanje, skladištenje i upotreba ličnih korisničkih podataka,

[2] https://vuejs.org/

sa ciljem njihove zaštite, kao i povećanjem kontrole korisnika nad svojim ličnim podacima. GDPR obuhvata pravila koja predstavljaju dobre prakse u informacionoj bezbednosti kojih bi se trebalo držati bez obzira na primenljivost same regulative.

Microsoft Azure cloud computing platforma posvećuje posebnu pažnju bezbednosti svojih servisa, kao i jednostavnoj integraciji bezbednosnih mehanizama u servise korisnika. Objavljen je dokument [5] koji prikazuje sažetak GDPR regulative i delova koje *Microsoft* izdvaja kao posebno značajne, kao i istaknute principe privatnosti i bezbednosti. U dokumentu se takođe mogu videti primeri *Azure* servisa i funkcionalnosti koje korisnici mogu da upotrebe kako bi na jednostavan način ispoštovali zahteve regulative. Neki od bezbednosnih principa istaknutih u dokumentu su: kontrola pristupa, logovanje i nadgledanje servisa, kriptografija, maskiranje podataka, itd.

3.2 Drugi osetljivi podaci

Pored osetljivih podataka koji spadaju u identifikujuće podatke (eng. PII – *personally identifiable information*) postoje i drugi osetljivi podaci koji nisu identifikujući za korisnike ali su po svojoj prirodi tajni, poput lozinki, API ključeva, brojeva platnih kartica, itd.

3.2.1 Azure SQL Database

Azure SQL Database, drugačije poznata kao SQL DB, omogućuje korisnicima da na jednostavan način, a često i jeftiniji, ispoštuju GDPR propise ali i druge bezbednosne principe. Zabrana suvišnih konekcija ka bazi predstavlja jedan od nivoa zaštite, i implementira se dodeljivanjem prava pristupa eksplicitnim IP adresama.

Autorizacija korisnika je kod SQL DB implementirana u vidu principa minimalnih privilegija. Svaki novi dodati korisnik podrazumevano nema nikakva prava, i ona mu se eksplicitno moraju dodeliti [5]. Preporučuje se kreiranje rola koja obuhvataju određene privilegije, i dodeljivanje rola korisnicima, za razliku od direktnog dodeljivanja privilegija svakom korisniku [5].

Enkripcija podataka je prisutna i u transportu i u skladištu. TLS (eng. *transport layer security*) protokolom zaštićen je sav saobraćaj koji ide ka i od baze podataka i nije potrebno ništa dodatno konfigurisati. Podaci u skladištu su enkriptovani zahvaljujući funkcionalnosti TDE (eng. *transparent data encryption*) [5] i obuhvata podatke iz baze, logove i podatke iz rezerve (eng. *backup*). Podrazumevano je uključena i nema uticaj na aplikacije koje koriste ove podatke, jer se oni enkriptuju i dekriptuju prilikom upisa i čitanja, a o kriptografskim ključevima i njihovoj rotaciji sam servis vodi računa [5].

Kao dodatni način zaštite podataka u upotrebi, SQL DB podržava maskiranje podataka. Maskiranje podrazumeva delimično sakrivanje osetljivih podataka od neprivilegovanih korisnika. Postoje dve vrste maskiranja i obe su podržane: statičko i dinamičko.

3.3 Azure Key Vault

Pored korisničkih podataka u sistemu, skladištenih u relacionoj bazi podataka, postoje i drugi osetljivi (nekorisnički) podaci koje je potrebno zaštititi. To su podaci poput konfiguracionih ključeva, tajni za pristup drugim servisima, konekcionih parametara za bazu, kriptografskih ključeva, itd. Kompromitovanje ovakvih tajni može dovesti do gubitka velike količine ličnih korisničkih podataka i može da ima ozbiljne posledice na sistem.

Azure Key Vault je servis namenjen upravo za ovakve svrhe. Kreiran je za svrhu skladištenja API ključeva, sertifikata, lozinki, itd. [6]. Dodatna prednost korišćenja *Key Vault*-a, u odnosu na druge načine poput skladištenja u sistemske varijable, je što su ključevi skladišteni na centralizovanom mestu. Sistemi koji se sastoje iz više komponenti često imaju potrebu da pristupaju istim ključevima iz različitih servisa. Kompromitovanje ključa u slučaju skladištenja u sistemskim varijablama servisa zahtevalo bi izmenu ključa na svakom mestu, dok je sa *Key Vault*-om dovoljno izmeniti kompromitovanu tajnu na jednom mestu, i promeniti ključeve za pristup samom *Vault*-u na onom mestu gde je došlo do napada. *Key Vault* koristi i softverske i hardverske mehanizme da zaštiti sve skladištene tajne [6].

3.4 Bezbednost servisa

Pored zaštite poverljivosti i integriteta podataka spomenutih u prethodnom poglavlju, drugi aspekt bezbednosti sistema ogleda se u bezbednosti samog servisa. Bezbednost servisa se sastoji iz više segmenata, poput zaštite komunikacije servisa sa ostalim komponentama (bazom podataka, klijentskim aplikacijama, drugim servisima, itd.), sprečavanja pristupa neovlašćenim korisnicima, kao i nadgledanja rada servisa i alarmiranja u slučaju sumnjivih radnji. Pored toga, i dostupnost (eng. *availability*) servisa je jedan od značajnih aspekata bezbednosti.

3.4.1 Deployment servisa

Kako bi aplikacija postala dostupna korisnicima, potrebno je prebaciti aplikaciju na neki od *cloud* servisa. Za one koji ne žele da imaju veliku kontrolu nad samom virtuelnom mašinom na kojoj se izvršava kod aplikacije, preporučuje se *Azure App Service* [7]. Ovaj servis je HTTP bazirani servis za hostovanje *web* aplikacija. Visoka dostupnost servisa je jedna od značajnih karakteristika *App Service*-a. Jedna aplikacija ovog servisa može biti pokrenuta na više instanci (do 30). Povećavanje broja instanci aplikacije povećava nivo dostupnosti, a ove instance opslužuju zahteve upućene servisu u *round-robin* maniru.

Slično kao i kod baze podataka, i *App Service*-u može se ograničiti pristup sa određenog skupa IP adresa, međutim u slučaju platnog sistema ovo nije moguće, imajući u vidu da je u pitanju javni servis.

Saobraćaj svih servisa obezbeđen je TLS protokolom, i za svaku aplikaciju moguće je i preporučljivo podesiti opciju odbijanja svih konekcija putem nebezbednog (HTTP) protokola. Sam *Azure* aplikacijama dodeljuje podrazumevani TLS sertifikat i brine o njegovoj rotaciji, bez potrebe za dodatnim konfigurisanjem ili kupovinom sopstvenog sertifikata.

3.4.2 Čuvanje logova

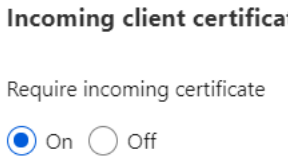
U *cloud* svetu se situacija sa otkrivanjem aplikativnih problema otežava jer nije moguće na isti način pristupati servisu, podacima i greškama kao na ličnom računaru. Iz ovog razloga treba voditi računa o tome kojim se sve informacijama (logovima) može pristupiti, gde se mogu

pronaći greške, itd. Pored toga, logovanje može doprineti boljem razumevanju korisničkih akcija i grešaka, ali i ukazati na sumnjive radnje koje mogu biti posledica potencijalnih napada na servis.

Kako bi se nivo detaljnosti logova *cloud* servisa, pored podrazumevanih logova *App Service*-a, među kojima su npr. logovi o *deployment* procesu i logovi *web* servera, poistovetio sa nivoom detaljnosti logova koje imamo prilikom lokalnog razvoja, *Azure Application Insights* [8] omogućuje integraciju sa *log4j*[3] logovima. *Application Insights* je servis koji omogućuje nadgledanje aplikacija, detekciju anomalija u performansama, agregiranje informacija o zahtevima, brzini odgovora, itd.

3.4.3 Dvosmerna autentifikacija

App Service podržava dvosmernu autentifikaciju i sve što je potrebno da bi se ona uključila jeste odabrati opciju za proveru klijentskog sertifikata (slika 3).



Slika 3. *Opcija za uključenje dvosmerne TLS autentifikacije.*

Dvosmerna autentifikacija kod ovog servisa većinu posla ostavlja samoj aplikaciji, koja sama mora proveriti integritet sertifikata. Jedini zadatak *App Service*-a jeste da proveri da li je klijentski sertifikat zapravo i poslat. Ukoliko nije, *App Service* će vratiti korisniku grešku i zahtev neće ni stići do same aplikacije.

3.4.4 Aplikativni firewall

Azure Web Application Firewall predstavlja centralizovani vid zaštite za *web* aplikacije. Implementira se uz pomoć aplikativnog *gateway*-a (eng. *application gateway*) na regionalnom nivou i jedan isti *firewall* može da opslužuje više aplikacija [9]. Ideja aplikativnog *firewall*-a je da se zaštita prebaci ispred aplikacija, i da se reši problem svih poznatih ranjivosti na jednom mestu, umesto da se bezbednost implementira u kodu svake od aplikacija.

Azure firewall baziran je na OWASP *Core Rule Set* [10] pravilima za detekciju najpoznatijih ranjivost, a između ostalog i ranjivosti sa OWASP *Top Ten* liste. Ovaj projekat kontinuirano radi na otkrivanju novih ranjivosti, i na unapređenju pravila, a *Azure* prati ovaj razvoj i konstantno ažurira svoja pravila kako bi *firewall* bio uvek u toku sa trenutnim ranjivostima.

4. ZAKLJUČAK

U ovom radu predstavljen je koncept sistema za plaćanje koji objedinjuje različite vrste plaćanja. Prikazana je arhitektura sistema koja obuhvata relacionu bazu podataka, klijentsku aplikaciju, i centralni deo sistema koji komunicira sa eksternim servisima za plaćanje poput *PayPal*-a i bankarskih sistema. Glavna motivacija za ovaj rad bila je istraživanje bezbednosnih mehanizama na *Azure* platformi. Stoga je veći deo rada posvećen opisivanju bezbednosnih aspekata ovakvog sistema.

Azure platforma se pokazala kao vrlo pogodna za integrisanje ovakvog platnog sistema, s obzirom da pruža podršku i usaglašenost sa GDPR regulativom, kao i široki spektar servisa i bezbednosnih mehanizama. Takođe, *Azure* nudi odličnu podršku za rad sa *SpringBoot* okvirom za rad u kom je ovaj sistem i implementiran, i naizgled nema velikih razlika i poteškoća u odnosu na podršku za *Microsoft* tehnologije.

Predlog za dalje unapređenje servisa bio bi istraživanje i integracija *Azure DDoS Protection Standard* servisa. Pored toga, dalje istraživanje bi moglo biti posvećeno mogućnostima drugih metoda autentifikacije, poput *Azure Active Directory*.

5. LITERATURA

[1] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, Manubot, 2019.

[2] W. Stallings, Cryptography and network security, Pearson Education India, 2006.

[3] https://itoc.com.au/blog/top-10-cloud-security-standards-and-control-frameworks.

[4] https://docs.microsoft.com/en-us/azure/sql-database/. [Poslednji pristup februar 2020].

[5] https://docs.microsoft.com/en-us/azure/sql-database/sql-database-security-overview. [Poslednji pristup februar 2020].

[6] https://docs.microsoft.com/en-us/azure/key-vault/key-vault-overview. [Poslednji pristup februar 2020].

[7] https://docs.microsoft.com/en-us/azure/app-service/overview. [Poslednji pristup mart 2020].

[8] https://docs.microsoft.com/en-us/azure/azure-monitor/app/app-insights-overview.

[9] https://docs.microsoft.com/en-us/azure/web-application-firewall/ag/ag-overview. [Poslednji pristup mart 2020].

[10] https://owasp.org/www-project-modsecurity-core-rule-set/. [Poslednji pristup mart 2020].

Kratka biografija:

Helena Zečević rođena je u Novom Sadu 1995. god. Osnovne akademske studije završila je na Fakultetu tehničkih nauka 2018. godine. Master rad iz oblasti Elektrotehnike i računarstva – Softversko inženjerstvo i informacione tehnologije odbranila je 2021. god.

kontakt: helena.zecevic@gmail.com

[3] https://logging.apache.org/log4j/2.x/

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.41
DOI: https://doi.org/10.24867/16BE09Kovacevic

ПРИМЕНА *HYPERLEDGER FABRIC BLOCKCHAIN*-А У ПОСЛОВНИМ ПРОЦЕСИМА

APPLICATION OF THE *HYPERLEDGER* FABRIC BLOCKCHAIN IN BUSSINESS PROCESSES

Ивана Ковачевић, *Факултет техничких наука, Нови Сад*

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – *Овај рад наводи теоријске основе на којима је заснована blockchain технологија. Објашњава концепте Hyperledger Fabric blockchain технологије. У раду је описан модел имплементираног решења и представљене су основне функционалности система које се ослањају на blockchain мрежу.*

Кључне речи: *blockchain, Hyperledger, паметни уговор, концезус алгоритам*

Abstract – *This paper presents theoretical basis of blockchain technology. The study includes concepts of Hyperledger Fabric blockchain and explains major considerations for implementation of system for applying for master thesis.*

Keywords: *blockchain, Hyperledger, smart contract, consensus algorithm*

1. УВОД

Пословни процеси имају за циљ оптимизацију послова како би се обезбедиле боље перформансе пословних система, што укључује повећање профита из угла компанија, и веће задовољство пруженим услугама, уз угла потрошача. Критичну тачку у пословним процесима представља верификација и трансформација дигиталних средстава који учествују у комуникацији између компанија без оствареног међусобног поверења.

Како би се постојећи проблем поверења превазишао, пословни процеси могу да се реализују ослањајући се на *blockchain* технологију и њене особине децентрализације и непроменљивости података. Овај рад се базира на примени *blockchain* технологије на пословним процесу у области високог образовања. Ова технологија у себи садржи криптографске примитиве које би, уколико се имплементирају на адекватан начин, могле значајно да допринесу смањењу плагираних или фалсификованих радова и диплома, што представља један од главних проблема са којима се сусреће академска заједница. Особина дистрибуираности *blockchain* мреже олакшава и убзава поступак провере исправности докумената, јер не захтева од заинтересованих страна да се директно обрате централизованом телу.

НАПОМЕНА:
Овај рад је проистекао из мастер рада чији ментор је био др Горан Сладић, ред. проф.

2. *HYPERLEDGER FABRIC*

Hyperledger Fabric [1] је радни оквир за развој дистрибуираних приватних *ledger*-а. Представња основу за развој апликација и решења са модуларном архитектуром. *Hyperledger Fabric* омогућава да компоненте као што су паметни уговори, концензуси или сервис за чланство (енгл. *membership service*) буду плагабилни, односно да их је могуће заменити или додати у систем по потреби.

2.1. *Hyperledger Fabric* модел

Hyperledger Fabric се састоји од асета, концензуса, дељеног *ledger*-а, паметног уговора и сервиса за чланство. *Ledger* чува тренутно стање пословних система у форми дневника трансакција. Физичке компоненте *Hyperledger Fabric-a* су чворови и с обзиром да је мрежа приватна, они могу међусобно да формирају канале за комуникацију скривену од остатка мреже. Приступањем каналу, компоненте прихватају међусобну сарадњу и деле идентичне копије *ledger-a* повезане са тим каналом. Канали су структуре на логичком нивоу, док чворови представљају контролне тачке за приступ и управљање каналима. Унутар *Hyperledger Fabric*-а могуће је приступати средствима (енгл. *asset*) или их мењати користећи трансакције паметних уговора. Средства на мрежи се представљају као колекција кључ-вредност парова, и могу да се чувају у бинарном или JSON формату.

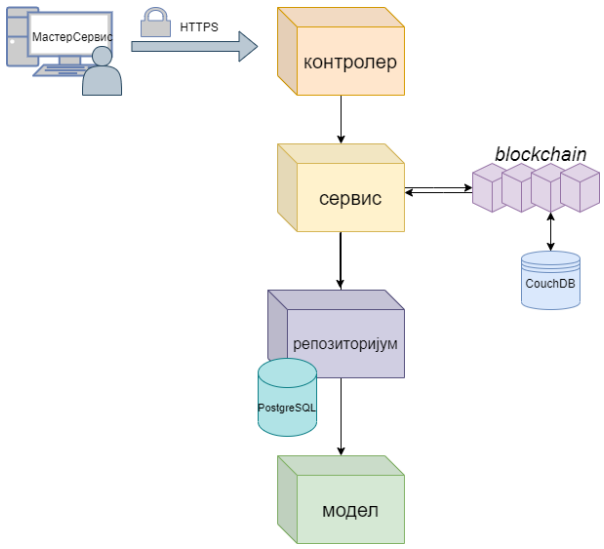
2.2. Паметни уговори и постизање концензуса

Паметни уговори представљају скуп договорених правила која чине пословни модел интеракције између организација [2]. У *Hyperledger Fabric* мрежи, термин који се користи за паметне уговоре је *chaincode*. *Chaincode* представља трансакциону логику која контролише животни век објеката на стању света. Са сваким паметним уговором је повезана и политика одобравања (енгл. e*ndorsement policy*). Политика одобравања дефинише које организације у *blockchain* мрежи морају да потпишу трансакцију како би се трансакција сматрала валидном [3]. *Chaincode* се пише у једном од тренутно подржаних програмских језика, који имплементирају прописани интерфејс. *Chaincode* се извршава у заштићеном *Docker* [4] контејнеру који је изолован од *endorsing* чворова. Стање на *ledger*-у које је креирано и ажурирано од стране једног *chaincode*-а је доступно само том *chaincode-y*. Концензус представља једну или скуп функција које примарно служе да би се постигао договор око редоследа трансакција. У *Hyperledger Fabric* технологији, концензус се постиже

када *orderer* и резултати трансакција једног блока испуњавају све критеријуме провере. Ове провере се одвијају током животног века трансакције, и укључују политику одобрења како би се тачно одредило који чланови мреже одобравају одређени тип трансакције.

3. МОДЕЛ СИСТЕМА

Намена система јесте да обезбеди информациону инфраструктуру за поступак пријаве и одбране завршних мастер радова на факултету. Систем је развијен као *web* апликација, са одвојеним клијентским и серверским делом, који остварују међусобну комуникацију употребом HTTPS протокола. Клијентска апликација се извршава у прегледачу (енгл. *browser*), док је серверски део монолитна апликација која се састоји од пакета, тако да подржава вишеслојну архитектуру. Серверски део комуницира са *Hyperledger Fabric SDK*-ом како би сместио и преузео потребне податке на *blockchain* мрежу. Архитектура система је приказана на слици 1.

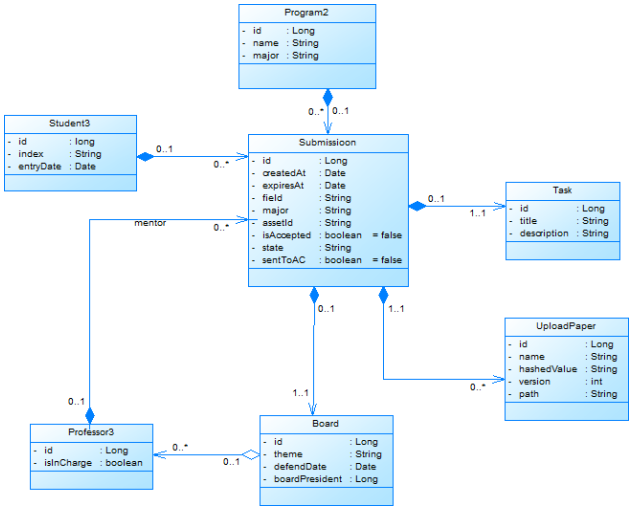


Слика 1. *Архитектура система*

Сви корисници система су део исте факултетске мреже, а систем сам по себи не пружа функционалности које би требало да буду видљиве неаутентификованим корисницима.

Примарни корисници система за предају мастер радова су студенти. Поред студената, апликацију користе још и референти студентске службе, чланови наставно научног већа, руководилац студијског програма и професори. Централно место система представља поступак подношења захтева за израду мастер рада и класе којеу учествују у овом процесу су дате на дијаграму на слици 2. Класа *Submission* моделује захтев који студент попуњава приликом пријаве за израду мастер рада, и поред основних поља за пријаву, садржи и додатна поља која прате статус у ком се пријава тренутно налази.

Пошто се пријава складишти и на *blockchain*-у и у бази података, постоје два јединствена идентификатора у класи, *id* који је идентификатор за базу података, и *assetId*, идентификатор који се користи приликом приступа *blockchain*-у. Идентификатор пријаве на *blockchain*-у ће доћи на систем као одговор приликом успешног додавања нове пријаве. Поред тога, за сваку пријаву се бележи датум када је настала и до кад важи, затим област из које студент жели да пише мастер рад, студијска група/подгрупа којој припада студијски програм на који је студент уписан као и додатни подаци о студенту и професору.



Слика 2. *Класни дијаграм за пријаву мастер рада*

4. ИМПЛЕМЕНТАЦИЈА

Blockchain мрежа је конфигурисана ослањајући се на *test-network* пример који је преузет са званичног git налога *Hyperledger Fabric*-а. Након што се пример преузме, у креираном фолдеру се налазе сви потребни CLI бинарни фајлови и докер слике потребне за конфигурацију и покретање мреже. За потребе система пријаве мастер рада, формиране су три организације, свака са по два чвора и један *ordering* сервис.

У овом решењу организације су подељене тако да прву организацију чине студенти, другу организацију представља студентска служба а трећу организацију чине професори. У решењу постоји један *ordering* сервис који управља трансакцијама, што се може оправдати чињеницом да поступак пријаве мастер рада није високо фреквентна операција, те *ordering* сервис неће бити оптерећен бројем пристиглих трансакција. Мрежа садржи два канала, ради очувања безбедности података који се налазе у трансакцијама. Један канал се користи за комуникацију између студената и студентске службе, а други канал преноси трансакције између студентске службе и професора. За генерисање сертификата се користио *Fabric CA*.

4.1 Регистрација корисника и генерисање сертификата

Прва компонента која се поставља на мрежу је сертификационо тело (CA). Ово тело служи за генерисање и потписивање осталих сертификата у мрежи, као што су сертификати везани за чворове и сертификати који идентификују администраторе и кориснике чворова. За регистрацију и упис администратора и корисника чворова, користи се *Fabric CA* клијент.

Клијент ће прво регистровати чворове, потом кориснике и на крају администратора организације.

Приликом регистровања, дефинише се којој организацији ти учесници припадају. Затим, сваки регистровани учесник има свој идентификатор у оквиру организације и тајну или лозинку. Такође, за сваког корисника се дефинише тип. Дефинисани типови су чвор, клијент и администратор. За TLS комуникацију између организација, наводи се путања до TLS сертификата.

Након што се корисници региструју, потребно је генерисати MSP фолдере за сваки чвор. За генерисање MSP фолдера се користи *enroll* команда при чему је потребно навести на ком порту је организација којој чвор припада подигнута.

4.2 Креирање чворова и канала

Параметри везани за конфигурацију чвора се налазе у `core.yaml` фајлу. Битни параметри који се конфигуришу у овом фајлу су идентификатор и назив чвора, мрежа којој чвор припада, адреса на којој ће чвор да "ослушкује" захтеве, путања до MSP фолдера као и подаци везани за *chaincode*. Сваки чвор комуницира са остатком мреже помоћу *gossip* протокола. Како би се смањио проток саобраћаја кроз канал, *Hyperledger Fabric 2.0* је конфигурисан тако да чворови преузимају нове блокове директно од *ordering* сервиса, уместо од других чворова [5].

У систему који је имплементиран, постоји само један *orderer* при чему се он ставља на располагање организацијама приликом креирања канала. За креирање *orderer*-а важе исти предуслови као и за креирање обичног чвора. Сваки *orderer* треба да има свој сертификат и приватни кључ који ће користити за потписивање трансакција, затим да има TLS сертификате као и MSP фолдер организације.

Канали се креирају тако што се прво креира трансакција за креирање канала у *genesis* блоку, а потом се *genesis* блок шаље *ordering* сервису у захтеву за придруживање каналу [6]. Трансакција за креирање канала специфицира иницијалну конфигурацију канала и креира се уз помоћ *configtxgen* алата.

4.3 Дефинисање и инсталација паметног уговора

Комплетна пословна логика садржана у паметном уговору се налази у *Hyperledger Fabric SDK* апликацији, написаној у *Java* програмском језику и развијеној уз помоћ *gradle* алата. *Java* омогућава употребу анотација како би се обезбедиле информације о паметном уговору и његовој структури. Класа која садржи кôд паметног уговора анотирана је са *@Contract(...)* анотацијом.

Ова анотација пружа могућност дефинисања додатних информација као што су назив, опис, лиценца и подаци о аутору. Да би класа представљала паметни уговор, поред анотација потребно је и да имплементира *ContractInterface*.

На слици 3 приказана је структура паметног уговора. Методе анотиране са *@Transaction* анотацијом означавају део кода који ће се извршавати као трансакциона функција. Методе које се позивају како би се додала нова трансакција, или како би се изменила постојећа трансакција су анотиране са типом *SUBMIT*, док су остале методе анотиране са типом *EVALUATE*.

Слика 3. *Паметни уговор*

Да би се приступило *ledger*-у, потребно је дефинисати контекст који ће бити доступан свим трансакцијама. Контекст се увек наводи као први параметар трансакционих метода. Да би се приступило стању света, контекст нуди *.getStub()* методу. Објекат који је враћен из ове методе служи како би се формирали упити ка бази података. Систем за пријаву мастер радова садржи једну методу за креирање трансакција односно за додавање нове пријаве на *ledger*. Ова метода садржи две главне провере. Прва провера омогућава јединственост трансакција. Уколико пријава прође ову проверу, позива се метода која имплементира предуслове које студент треба да испуни како би пријава била валидна. Услови да би студент могао успешно да поднесе пријаву су да је прошло најмање 6 месеци од датума уписа студијског програма, и да област из које студент жели да ради мастер рад припада области предмета на студијском програму. Ако су оба критеријума испуњена, креира се нови JSON објекат од параметара који чине пријаву. Креирани објекат се серијализује и додаје у стање света, под одговарајућим кључем. Да би се написани *chaincode* извршавао на чворовима, потребно га је прво исталирати на канал. За инсталацију паметног уговора користи се команда наведена на листингу 1.

```
./network.sh down
./network.sh up createChannel -ca
./network.sh deployCC -ccn basic -ccp ../asset-
transfer-basic/chaincode-java -ccl java
```

Листинг 1. *Покретање blockchain мреже*

4.4 Повезивање апликације са *chaincode*-ом

Да би апликација могла да креира трансакције и проверава стање света на *ledger*-у, потребно је успоставити конекцију са *Hyperledger Fabric SDK*-ом. Прво, апликација пријављује администратора. Креденцијале за администратора генерише *Fabric CA* клијент. Да би се апликација успешно повезала са CA клијентом, потребно је поставити путању до сертификата апликације. Ако су сертификати исправни, *Fabric CA* клијент ће креирати сертификат за администратора и вратиће га апликацији. Сви сертификати се смештају у електрнски новчаник (енгл. *wallet*). Након уписа администратора, потребно је регистровати кориснике. Поступак регистрације корисника је сличан. У трећем кораку, потребно је повезати се на мрежу помоћу *gateway*-а. За успостављање конекције користи се претходно регистрован корисник, односно његови креденцијали учитани из *wallet*-а. Ако је остварена конекција, могуће је добити објекат класе *Network* помоћу којег апликација приступа паметном уговору. Након добијања инстанце паметног уговора, над том

инстанцом је могуће позвати било коју трансакциону методу из класе паметног уговора. На листингу 2 је дат пример повезивања на *blockchain*.

```
try (Gateway gateway = connect()) {
        Network network =
        gateway.getNetwork("mychannel");
        Contract contract =
        network.getContract("basic");
        byte[] result =
contract.submitTransaction("CreateSubmissio
n", String.valueOf(dto.getAssetID()),
        dto.getExpiresAt(),
        dto.getMajor(), dto.getField(),
        dto.getIndex(), dto.getProfesor(),
dto.getProgramName(), dto.getEntryDate());
return new String(result);

} catch (Exception e) {
        System.out.println(e);

}
```

Листинг 2. *Позивање паметног уговора*

4.5 Функционалности система

Апликација прати процес пријаве мастер рада што обухвата попуњавање пријаве, одобравање теме и потом формирање комисије, постављање рада, остављање примедби и на крају одобравање рада. Функционалности се ослањају на позиве ка *blockchain* мрежи а сваки од ових корака обавља једна од улога у систему. Процес пријаве започиње студент, на почетној страници, попуњавањем форме за пријаву. Подаци који обухватају име и презиме, број индекса и студијски програм студента се повлаче из базе података и иницијално попуњавају форму а студент на форми треба да попуни област, студијску групу и наставника код ког жели да пише рад. Када се пошаљу поља форме, креира се *Submission* објекат који се шаље на *blockchain* мрежу.

Позивом одговарајуће методе, активира се код паметног уговора, приказан на листингу 2. Метода ће бити позвана ако постоје кориснички креденцијали у *wallet*-у. Након послате пријаве, систему приступа референт. Референту се на почетној страници се приказују све пристигле пријаве. Референт може да прегледа профил студента где му се прилазује историја полагања испита, просечна оцена и студијски програм за појединачног студента. Када референт прегледа пријаву може да упути захтев наставно научном већу, чиме се мења статус пријаве на *ledger*-у. У следећем кораку, систему приступа неко од чланова наставно научног већа, како би се прегледао пристигли захтев и именовала комисија.

Менор рада може да се пријави на систем како би прегледао листу свих одобрених пријава. Из понуђене листе, ментор може да одабере пријаву за коју ће да дефинише задатак. Студент ком је дефинисан задатак, може да прегледа задатак и да на истој страници постави верзију рада. Постављени рад се хешује помоћу *Bcrypt* алгоритма, након чега се тај хеш шаље на *blockchain* мрежу. Након што студент постави рад, професори који су чланови комисије могу на страници за постављене радове да преузму рад, да оставе примедбу или да га одобре. Ако ни један члан комисије нема примедби на рад, ментор дефинише датум и место одбране, а студент може да провери стање на свом профилу.

5. ЗАКЉУЧАК

С обзиром да се у данашње време све више инсистира на системима који гарантују корисницима безбедност, приватност и поверење, *blockchain* може да се употреби како би се ови захтеви пренели са централизованог тела на технологију, где ће бити довољно ослонити се на исправно функционисање одабраних криптографских примитива и правилној конфигурацији мреже и инсталираних паметних уговора. Овај рад се базирао на употреби *Hyperledger Fabric blockchain* технологије отвореног кода, за решавање проблема аутентичности и поверљивости података који учествују у процесу пријаве мастер рада. Предности система су употреба ECC асиметричног алгоритма за генерисање приватних и јавних кључева, потом генерисање сертификата, као и поседовање комплетне историје свих записа на *ledger*-у. Постојеће решење је могуће проширити на начин који би омогућио различитим универзитетима да приступе мрежи, верификују постојеће радове и дипломе, као и да омогуће другим универзитетима увид и верификацију радова њихових студената. На тај начин, повећава се транспарентност у раду академских заједница, док се истовремено гарантује интегритет података доступних на мрежи.

6. ЛИТЕРАТУРА

[1] *Hyperledger Fabric*, https://wiki.hyperledger.org/display/fabric/HyperledgerFabric (приступљено у септембру 2021)

[2] Christian Gorenflo, Stephen Lee, Lukasz Golab, Srinivasan Keshav, *FastFabric: Scaling hyperledger fabric to 20 000 transactions per second*, 2020

[3] Smart contract, https://hyperledger-fabric.readthedocs.io/en/release-2.2/smartcontract/smartcontract.html (приступљено у септембру 2021)

[4] https://www.docker.com/

[5] *Gossip* протокол, https://hyperledger-fabric.readthedocs.io/en/latest/deploypeer/peerchecklist.html#peer-gossip (приступљено у септембру 2021)

[6] *Planning ordering service*, https://hyperledger-fabric.readthedocs.io/en/latest/deployorderer/ordererplan.html (приступљено у септембру 2021)

Кратка биографија:

Ивана Ковачевић рођена је у Новом Саду 1997. године. Мастер рад на Факултету техничких наука из области Електротехнике и рачунарства – Електрнско пословање одбранила је 2021. год.
контакт: kovacevic.ivana@uns.ac.rs

Зборник радова Факултета техничких наука, Нови Сад

УДК: 004.738.1:766
ДОИ: https://doi.org/10.24867/16BE10Kevac

СТЕГАНОГРАФИЈА И СТЕГОАНАЛИЗА

STEGANOGRAPHY AND STEGANALYSIS

Елена Кевац, *Факултет техничких наука, Нови Сад*

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – *У овом раду описана је стеганографија као техника скривеног записа. Детаљно су објашњени типови стеганографије, као и алати који се користе да би се ова техника спровела у дјело. Такође, у наставку рада описана је стегоанализа која представља метод откривања стеганографије, као и алати који се користе у те сврхе.*

Кључне речи: *стеганографија, стегоанализа, дигитална форензика*

Abstract – *This paper describes steganography as a technique of hidden notation. The types of steganography are explained in detail, as well as the tools used to put those technique into use. Also, in the continuation of the paper, steganalysis is described as a method of detecting steganography, as well as the tools used for that purpose.*

Keywords: *steganography, steganalysis, digital forensics*

1. УВОД

Један од најважнијих фактора информационих технологија и комуникација јесте безбједност информација. Криптографија је настала као техника осигуравања тајности комуникације. Развијене су многе методе шифровања и дешифровања порука, али понекад није довољно да садржај поруке буде тајан, већ и да постојање поруке буде тајна за неауторизоване кориснике. Развијена је научна дисциплина чији је основни задатак управо то, а њен назив јесте стеганографија.

Назив потиче од грчке ријечи *steganós* (στεγανός) што значи прикривен или тајни, и *–graphia* (γραφή) што значи писање. Стеганографија има врло широке могућности примјене, од прикривене размјене података у приватне и пословне сврхе па до заштите ауторских права у облику воденог жига. Ипак, због свог темељног принципа „невидљивости" често се користи и у илегалне сврхе.

За стегоанализу се може рећи да је она за стеганографију оно што је криптоанализа за криптографију или антивирусни софтвер за рачунарски вирус. То је у ствари процес откривања малих промјена у очекиваном шаблону структуре мултимедијалне датотеке. Дигитална форензика се фокусира на очување и анализу дигиталних доказа. Дефиниција дигиталне форензике је: „Употреба научно изведених и доказаних метода за очување, прикупљање, валидацију, идентификацију, анализу, тумачење, документацију и представљање дигиталних доказа изведених из дигиталних извора у сврху олакшавања или унапређења реконструкција догађаја за које се утврди да су кривични или помаже у предвиђању неовлаштених случајева акције за које се показало да ремете планиране операције" [1].

НАПОМЕНА:
Овај рад је проистекао из мастер рада чији ментор је био др Стеван Гостојић, ванр. проф.

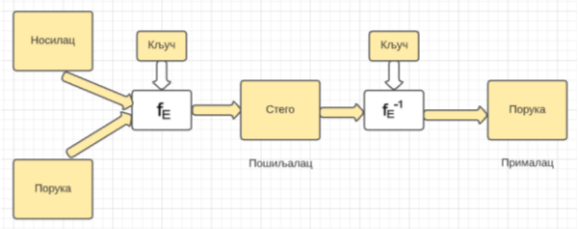
2. СТЕГАНОГРАФИЈА

2.1. Појам и класификација стеганографије

Стеганографија подразумијева прикривање тајне поруке, али не и чињенице да двије стране комуницирају међусобно. Стога, процес стеганографије обично укључује уметање тајне поруке унутар неког преносног медија који се назива носилац и има улогу прикривања постојања тајне поруке. Носилац треба да буде такав скуп података који је саставни дио свакодневне комуникације те као такав не привлачи посебну пажњу на себе. Најчешћи примјери су:

- Слике (.bmp, .gif, .jpeg и сл.)
- Видео записи (.avi, .mpg, .vob и сл.)
- Звучни записи (.mp3, .midi, .wav, .wma и сл.)
- Датотеке (.doc, .xls, .ppt, .txt и сл.)

Пошиљалац кориштењем кључа и стеганографске функције (f_E) утискује поруку у носиоца. Као резултат добија се стеганографски објекат (стего). Стего је комбинација поруке и носиоца (и евентуално кључа). Носилац ће бити видљив свима и неће изазвати сумњу у постојање тајне поруке. Кориштењем стего кључа, пошиљалац маскира поруку у носиоца. Истим кључем прималац издваја поруку из носиоца. Стего кључ се може појавити у више различитих облика. Он може да буде обична лозинка или може да буде позиција у носиоцу. Нападач има могућност пресретања поруке и манипулације стего објектом. Откривањем тајне поруке, нападач је може уништити, измијенити или издвојити.



Слика 1. *Стеганографски систем*

На слици бр. 1 приказан је начин функционисања стеганографског система, гдје је:

- Носилац – медиј унутар којег се сакрива тајна порука
- Порука – тајна порука која треба бити сакривена
- Кључ – стеганографски кључ, параметар функције f_E
- f_E – стеганографска функција „ уграђивање”
- Стего – стеганографска датотека
- ${f_E}^{-1}$ – стеганографска функција „издвајање”

Исти носилац никада не би требало да се користи два пута, јер нападач који има приступ двјема верзијама једне те исте слике може лако детектовати и евентуално реконструисати поруку. Да би се избјегла случајна поновна употреба, и пошиљалац и прималац би требало да униште носиоце које су већ користили за пренос информација.

2.3 Класификација техника стеганографије

Стеганографске технике се у глобалу дијеле на техничку и лингвистичку.
Техничка стеганографија користи научни приступ да би се сакрили подаци. Ова техника се огледа у употреби специјалних уређаја, инструмената и метода у скривању поруке. Техничкој стеганографији припадају сљедеће технике: невидљиво мастило, скривена мјеста, микрофотографије као и дигитална стеганографија. Због своје опширности и фокуса у овом раду, дигитална стеганографија обрађена је у сљедећем одјељку.

2.3.1 Дигитална стеганографија

Код дигиталне стеганографије порука се сакрива у неком од дигиталних медијума попут слике, видеа, звука, текстуалног документа или унутар мрежних пакета. Централни концепт сликовне стеганографије је поступак скривања поруке унутар слике тако да она буде невидљива оку у оригиналној слици. У звучној стеганографији звучна датотека се користи као носилац за прикривање повјерљивог садржаја уз помоћ људског слушног система.

Видео стеганографија је проширење стеганографије слика. Видео снимци нуде нове могућности за скривања података, попут скривања поруке у компонентама покрета. Звучна компонента видео садржаја се такође може искористити за сакривање поруке. Текстуална стеганографија се сматра врло често и најтежом због недостатка сувишних података у текстуалној датотеци. Када је у питању мрежна стеганографија, она се односи на технике уградње скривених информација унутар порука које се шаљу преко мреже унутар TCP/IP заглавља. У наставку текста детаљно су објашњене поменуте технике.

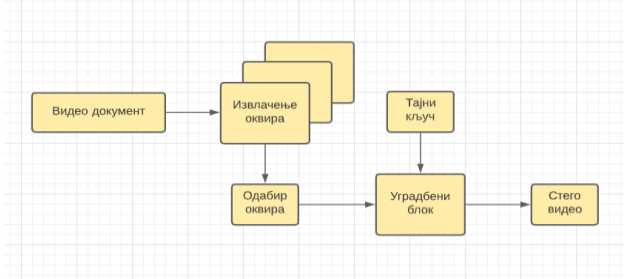
2.3.1.1 Звучна стеганографија

Технике скривања порука у звучној датотеци ослањају се на два корака. Први корак је означавање битова који се понављају у звучној датотеци. Други корак је уметање повјерљиве поруке замјеном ових битова са битовима поруке. Најчешће су три технике звучне стеганографије:

1. Фазно кодирање (енг. *phase encoding*)
2. Ширење спектра (енг. *spread spectrum*)
3. Сакривање одјека (енг. *echo data hiding*).

2.3.1.2 Видео стеганографија

Видео датотека која садржи различите оквире слика користи се као носач за покривање података. Велика количина тајних података може се сакрити у видео датотекама као што су MPEG, MP4, AVI итд.



Слика 2. Основни блок дијаграм за видео стеганографију

Потребни кораци изведени у видео стеганографији су сљедећи:

- Одабир одређеног видео записа у који желимо да уградимо поруку
- Подјела видеа у мале оквире
- Одабир одређене структуре у коју желимо да се тајни подаци убацују
- Тајни кључ је постављен за уграђивање са одређеним оквиром, а затим се стего видео шаље пошиљаоцу

2.3.1.3 Мрежна стеганографија

Односи се на технике уградње скривених информација унутар порука које се шаљу преко мреже унутар *TCP/IP* заглавља. *TCP/IP (*енг. *transmission control protocol/internet protocol).*

3. СЛИКОВНА СТЕГАНОГРАФИЈА

Код овог типа стеганографије, датотеке са екстензијама JPEG, GIF, BMP, PNG итд. користе се за складиштење битова тајне поруке. Најпознатије технике сликовне стеганографије су технике просторног домена, технике фреквентног домена и кориштење формата слике.

3.1 Кориштење формата слике

Најједноставнији метод за скривање информације јесте употреба формата слике код којег се на крај датотеке са сликом убацује текстуална датотека. Порука се додаје након EOF-а. EOF је ознака за крај датотеке. Када се слика отвори у неком програму за преглед фотографија чита се само онај дио до EOF-а, а остатак се занемарује. Кориштењем овог метода не смањује се квалитет слике, али је поруку врло лако открити, односно довољно је отворити слику у било којем уређивачу текста.

3.2 Технике просторног домена

У технике просторног домена спадају замјена бита најмање важности, филтрирање и маскирање, сортирање палета и деградација слике.

3.2.1 Замјена бита најмање важности (LSB алгоритам)

LSB алгоритам (енг. *least significant bit substitution*) је најпопуларнија и најчешће кориштена техника сликовне стеганографије. Идеја ове технике се заснива на

растављању оригиналне поруке у битове, који се потом убацују на позицију бита најмање важности. Појам бит најмање важности се односи на нумеричку вриједност бита у октету, односно на његову тежинску вриједност. Октет заједно са садржавајућим битовима и њиховим тежинским вредностима приказан је на слици 3.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	1	0	0	1	1

Слика 3. *Октет*

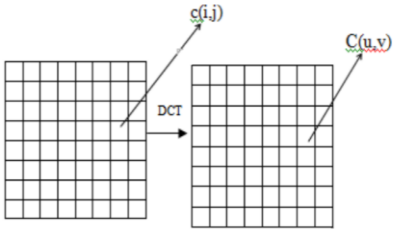
Из претходног објашњења лако је закључити да промјена бита најмање вриједности има најмањи утицај на промјену укупне вриједности октета. Из овога се закључује да промјене бита на најмањим тежинским позицијама у октетима изазивају најмањи утицај на промјену изгледа оригиналне слике.

3.3 Технике фреквентног домена

Технике фреквентног домена користе алгоритме и трансформације, односно математичке функције које се користе и код техника компресије. Најпознатији су алгоритми дискретне косинусне трансформације и дискретне таласне трансформације.

3.3.1 Дискретна косинусна трансформација

Основна улога дискретне косинусне трансформације (енг. *discrete cosine transform – DCT)* је да трансформише сигнал или слику из просторног у фреквентни домен као што је приказано на слици 6. Корисна је при дијељењу слике на различите дијелове различитог значаја, у односу на квалитет слике. Слика се дијели у високофреквентне, средњефреквентне или нискофреквентне компоненте.



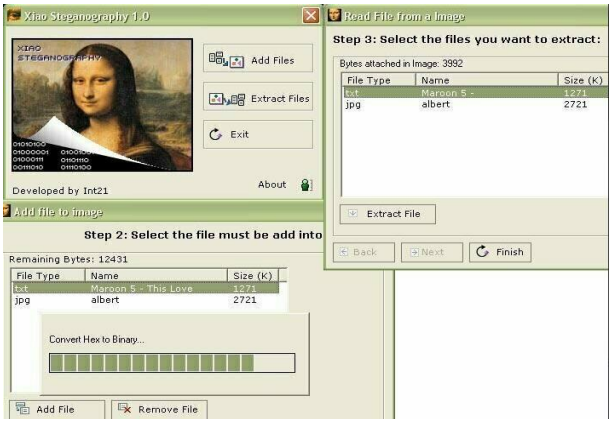
Слика 4. *Дискретна косинусна трансформација слике [3]*

4. АЛАТИ ЗА СТЕГАНОГРАФИЈУ

Постоје многи алати који нуде могућност стеганографије. Неки од њих нуде само стеганографију, док други нуде криптографију прије сакривања података. Примјер алата који се користи у ове сврхе је Xiao Steganography.

4.1 Xiao Steganography

Xiao Steganography је бесплатан софтвер који се може користити за скривање тајних датотека у *BMP* сликама или *WAV* датотекама. Коришћење алата се заснива на томе да се учита било која *BMP* или *WAV* датотека, а затим се дода датотека која треба да буде сакривена. Овај алат подржава и шифровање. Даје могућност бирања алгоритма за шифровање као што су: *RC4, Triple DES, DES, Triple DES 112, RC2* и *hashing SHA, MD4, MD2* и *MD5.* Изабере се било који од њих и сачува се циљна датотека. Да би се прочитала скривена порука из ове датотеке, користи се поново исти софтвер. Овај софтвер ће прочитати датотеку и декодирати скривену датотеку из ње. *Xiao Steganography* има интуитиван изглед који олакшава коришћење као што је приказано на слици 5.



Слика 5. *Изглед алата Xiao Steganography [2]*

5. СТЕГОАНАЛИЗА

5.1 Дефиниција

Стегоанализа (енг. *steganalysis*) је процес детектовања стеганографског садржаја који се заснива на проучавању варијација узорака битова и необично великих датотека.

5.2 Циљеви Стегоанализе

Стегоанализа је обрнути процес у односу на стеганографију. Док је код стеганографије циљ сакрити податке у носиоца, код стегоанализе циљеви су:

- идентификација сумњивих скупова података (сигнали или датотеке), унутар којих се потенцијално налазе скривене поруке,
- утврђивање да ли су подаци уметнути у носиоца шифровани прије процеса стеганографије,
- утврђивање постојања шума или небитних података унутар сигнала или датотеке и
- издвајање и дешифровање уметнуте поруке из стего објекта.

5.3 Облици Стегоанализе

Напади и анализе скривених података које изводи аналитичар укључују различите активности попут детекције, издвајања, онемогућавања или уништавања скривених података. Врста напада који ће бити изведен зависи од информација које су доступне аналитичару. У зависности од тога, постоје облици напада у којима је познато сљедеће

- само стеганографска датотека (енг. *steganography-only attack*) – доступна је само стеганографска датотека над којом се потом изводе различите анализе,
- познати носилац (енг. *known-carrier attack*) – доступна је стеганографска датотека и носилац, а упоређивањем двају датотека се долази до скривеног садржаја. Најбољи примјер је употреба *LSB* технике за скривање тајне поруке у

логоу *Google*-а. Пошто је носилац познат, а доступан је нападачу, он врло лако упоређивањем добијене и оригиналне фотографије може да екстрахује скривене информације,

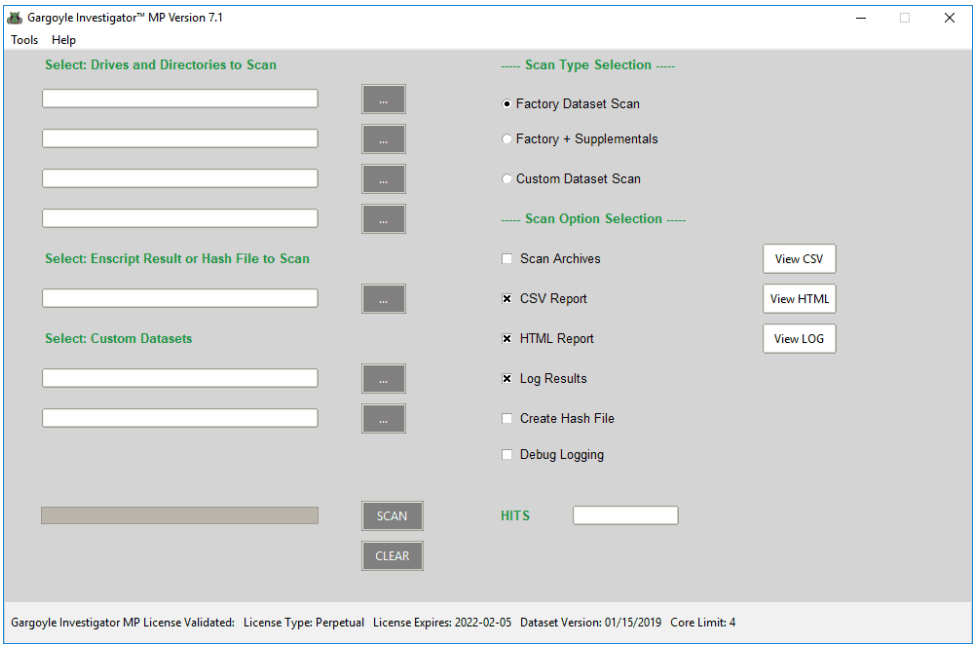
- позната порука (енг. *known-message attack*) – код овог напада доступна је тајна порука,
- изабрана стеганографска техника (енг. *choosen-steganography attack*) – позната је и стеганографска датотека и стеганографска техника, односно алгоритам који је кориштен за уметање поруке,
- изабрана порука (енг. *chosen-message attack*) – позната је порука и стеганографски алгоритам кориштен за креирање стеганографске датотеке која ће се користити за будућу анализу и упоређивање,
- познати носилац и стеганографска техника (енг. *known-steganography attack*) – расположива је стеганографска датотека, стеганографски носилац, као и алгоритам кориштен за уметање тајне поруке. Сврха овог напада јесте утврђивање одговарајућих узорака у стеганографској датотеци који могу указати на кориштење одређеног стеганографског алгоритма.

6. АЛАТИ ЗА СТЕГОАНАЛИЗУ

Постоје многи алати који нуде могућност стегоанализе. У овом одјељку дат је примјер једног алата који се користи у те сврхе, а то је Gargoyle.

6.1 Gargoyle

Gargoyle је програм развијен од стране *WetStone technologies* 2004. године (раније *StegoDetect*), који се може користити за детекцију присуства стеганографских садржаја. Овај алат користи заштићени скуп података свих датотека из познатих стеганографских алата, упоређујући их са помијешаним датотекама предмета који се истражује. *Gargoyle* скуп података може бити кориштен и за детекцију присуства криптографије, хитних порука, кључа за записивање у оперативни регистар, тројанских коња и других злонамјерних софтвера. Изглед алата Gargoyle приказан је на слици 6.



Слика 6. *Изглед алата Gargoyle [3]*

7. ЗАКЉУЧАК

Иако се стеганографски алати могу користити за легитимне примене као што је заштита тајности пословних или приватних информација, постали су предмет интересовања форензичких истраживача који се баве њиховом злонамјерном и незаконитом употребом. Како поменути алати постају све доступнији и лакши за употребу, заштита од злонамјерне употребе захтијева све већу пажњу. Такође, равнотежа између заштите од незаконите употребе и мијешање у легитимну употребу појављују се као нови изазов.

8. ЛИТЕРАТУРА

[1] Sadhana Rathore, "Steganography: Basics and digital forensics", *International Journal of Science, Engineering and Technology Research (IJSETR),* Volume 4, Issue 7

[2] XiaoSteganography (https://xiao-steganography.en.softonic.com/)

[3] Gargoyle (https://www.wetstonetech.com/products/gargoyle-malware-detection-dfir/)

Кратка биографија:

Елена Кевац рођена је 10.05.1994. године у Бањалуци. Мастер рад на Факултету техничких наука из области Рачунарство и аутоматика – Софтверско инжењерство одбранила је 2021.год. контакт: elenakevac@gmail.com

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.3.049
DOI: https://doi.org/10.24867/16BE11Vincan

MEMRISTIVNA IMPLEMENTACIJA PRIRODNIH NEURONA I SINAPSI

MEMRISTIVE IMPLEMENTATION OF SPIKING NEURONS AND SYNAPSES

Vladimir Vincan, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTOTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu je opisana softverska implementacija Moris-Lekarovog memristivnog modela prirodnog neurona i Zamarenovog memristivnog modela sinapse. U Matlab-u i Simulink-u su implementirani matematički modeli memristora koji su iskorišćeni za simulaciju kalcijumovih i kalijumovih kanala Moris-Lekarovog modela neurona. Na kraju, pokazano je kako se može implementirati STDP pravilo učenja na Zamarenovom memristivnom modelu sinapse.*

Ključne reči: *neuromorfno računarstvo, prirodni neuron, sinapsa, STDP pravilo učenja, memristor*

Abstract – *In this paper, software implementations of the memristive Morris-Lecar model of the spiking neuron and the memristive Zamarreno model of the synapse have been developed. Matlab and Simulink models of memristors have been used for the simulation of calcium and potassium channels of the Morris-Lecar neuron model. It has been shown, how the STDP learning rule can be implemented on the memristive Zamarreno model of the synapse.*

Keywords: *neuromorphic computing, spiking neuron, synapse, STDP learning rule, memristor*

1. UVOD

Istraživanja i razvoj veštačkih neuronskih mreža datiraju još od četrdesetih godina prošlog veka [1]. Da bi jednog dana veštačke neuronske mreže mogle dostići, pa čak i prevazići sposobnosti prirodnih neuronskih mreža, odnosno mozga, neophodno je verno modelovati njihove najvažnije gradivne elemente – neurone i sinapse.

Memristori omogućavaju novi i izuzetno pogodan način modelovanja određenih osobina neurona i sinapsi, zbog sposobnosti promene i zadržavanja tekuće vrednosti svoje otpornosti. Memristor je prvi put opisan 1971. godine od strane Leona Čue [2] i predstavlja četvrti osnovni dvokrajni pasivni element u svojoj osnovnoj verziji sa stanovišta teorije električnih kola, kojim su povezani fluks (integral napona po vremenu) i količina naelektrisanja (integral struje po vremenu).

Istraživači iz HP laboratorije uspeli su da naprave prvi memristor baziran na nanometarskim filmovima titanijum dioksida [3], 37 godina nakon što je teorijski predviđeno njegovo postojanje.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Staniša Dautović, vanr. prof.

Relativno skorašnja pojava memristora i drugih mem-elemenata rezultat je značajno povećanog interesovanja istraživačke zajednice, motivisane brojnim potencijalnim primenama memristivnih elemenata. Među ovim primenama istaknuto mesto zauzima oblast neuromorfnog računarstva [4] [5].

Ovaj rad sastoji se iz pet delova. Nakon prvog, uvodnog dela, u drugom delu je data matematička definicija memristora i opisan je Zamarenov model memristora [6]. U trećem delu je opisan Moris-Lekarov memristivni model neurona [7], kao i najčešće korišćeni LIF (engl. *Leaky Integrate and Fire*) nememristivni model prirodnog neurona. U četvrtom delu je definisano STDP (engl. *Spike Time Dependent Plasticity*) pravilo učenja na sinapsama. U petom delu je iznesen zaključak.

2. ZAMARENOV MODEL MEMRISTORA

Opšti oblik generičkog naponom kontrolisanog memristora je definisan sledećim konstitutivnim relacijama [8]:

$$i_{MR} = G(\boldsymbol{x}) \cdot v_{MR}, \tag{1}$$

$$\frac{d\boldsymbol{x}}{dt} = g(\boldsymbol{x}, v_{MR}). \tag{2}$$

Dualno, opšti oblik generičkog strujom kontrolisanog memristora je definisan sledećim konstitutivnim relacijama:

$$v_{MR} = R(\boldsymbol{x}) \cdot i_{MR}, \tag{3}$$

$$\frac{d\boldsymbol{x}}{dt} = f(\boldsymbol{x}, i_{MR}). \tag{4}$$

Funkcije $G(\boldsymbol{x})$ i $R(\boldsymbol{x})$ predstavljaju memduktansu (provodnost) i memristansu (otpornost), $\boldsymbol{x}$ predstavlja vektor promenljivih stanja memristora, dok (2) i (4) predstavljaju jednačine stanja. Funkcije $g(\boldsymbol{x}, \mathrm{v_{MR}})$ i $f(\boldsymbol{x}, i_{MR})$ se tipično razlikuju za različite modele memristora. U ovom radu je u Matlab-u i Simulink-u implementiran Zamarenov model memristora [6], koji spada u klasu generičkih memristora [8] i čije su konkretne konstitutivne relacije:

$$R(x) = k_R \cdot (x + x_0), \tag{5}$$

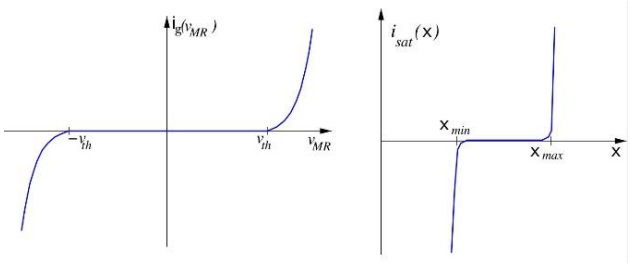
$$C_{MR}\dot{x} = i_g(v_{MR}) - i_{SAT}(x), \tag{6}$$

$$i_g(v_{MR}) = \begin{cases} I_0 \cdot sign(v_{MR}) \cdot \left[e^{|v_{MR}|/v_0} - e^{v_{th}/v_0}\right], |v_{MR}| > v_{th} \\ 0, u\ suprotnom \end{cases}. \tag{7}$$

Inverzna sigmoidalna funkcija struje i_g, jednačina (7), dominantno oblikuje desnu stranu jednačine stanja (6). Promena vrednosti promenljive stanja je moguća tek ukoliko apsolutna vrednost napona na memristoru $|v_{MR}|$ pređe naponski prag $|v_{th}|$, Slika 1a. Jedan primer prirodnog sistema koji može biti modelovan na ovaj način su sinapse između neurona. Potrebno je da se dostigne određen naponski prag na membrani neurona kako bi se otvorio odgovarajući jonski kanal i omogućio protok pozitivno i negativno naelektrisanih jona.

Struja i_{SAT} je saturaciona funkcija koja onemogućava prekoračenje opsega promenljive stanja. Njena vrednost je zanemariva sve dok se promenljiva stanja ne približi graničnoj vrednosti. Idealizovan oblik funkcije i_{SAT} je prikazan na Slici 1b i u ovom radu je odabrano da data funkcija bude definisana na sledeći način:

$$i_{SAT} = 0.005 \cdot tan\left(\frac{\pi}{2} \cdot \frac{x}{x_{max}}\right). \tag{8}$$



Slika 1. *Oblici funkcija (a)* $i_g(v_{MR})$ *i (b)* $i_{SAT}(x)$ [6]

Parametri modelovanog Zamarenovog memristora su prikazani u Tabeli 1.

$x_{min}(V)$	-10	$k_R^{-1}(nA)$	222	$I_0(\mu A)$	10
$x_{max}(V)$	10	$C_{MR}(mF)$	10	$v_0(V)$	0.1
$x_0(V)$	0			$v_{th}(V)$	1

Tabela 1. *Parametri realizovanog Zamarenovog memristora*

3. MORIS-LEKAROV I LIF MODEL NEURONA

Neuron šalje informacije susednim ćelijama elektrohemijski, odnosno preko naelektrisanih jona. Dok je membrana zatvorena, nema protoka jona i postoji konstantna razlika potencijala između dva kraja ćelijske membrane, koja se zove potencijal mirovanja (engl. *resting potential*). Prilikom kretanja jona, razlika potencijala poprimi oblik šiljka (engl. *spike*) koji se zove akcioni potencijal (engl. *action potential*). Akcioni potencijal ima značajnu ulogu jer biološkim sinapsama omogućava promenu težine sinaptičkih veza [9].

Modeli neuronskih ćelija se dele po svojoj složenosti na tri kategorije – eksplicitni modeli, koji su biološki najtačniji, ali zahtevaju najviše računarskih operacija i simulacije su uglavnom spore; generalni modeli, koji opisuju biološko ponašanje ali izostavljaju neke fizičke karakteristike koje uzrokuju takvo ponašanje; i generički modeli koji okidaju impuls nakon što je pređen neki prag [1]. Generički modeli su najjednostavniji i brzo se simuliraju, ali nisu toliko precizni.

Moris-Lekarov (ML) model [7] predstavlja eksplicitni model neuronske ćelije i zasnovan je na mišićnim vlaknima infraklase morskih rakova, vitičarima. Pokazano je da za električnu interpretaciju jonskih kanala ML modela neurona potrebno koristiti memristore, umesto vremenski promenljivih otpornika [10], što je prikazano na Slici 2a. Konstitutivne relacije ML modela neurona su definisane na sledeći način :

$$I = C_M \frac{dV}{dt} + I_{ionic}, \tag{9}$$

$$I_{ionic} = I_{Ca} + I_K + I_L, \tag{10}$$

$$I_{Ca} = \bar{g}_{Ca} \cdot M \cdot (V - E_{Na}), \tag{11}$$

$$I_K = \bar{g}_K \cdot N \cdot (V - E_K), \tag{12}$$

$$I_L = g_L \cdot (V - E_L), \tag{13}$$

$$\frac{dM}{dt} = \lambda_M(V)[M_\infty(V) - M], \tag{14}$$

$$\frac{dN}{dt} = \lambda_N(V)[N_\infty(V) - N], \tag{15}$$

gde su:

$$M_\infty(V) = \frac{1}{2}\left\{1 + tanh\left[\frac{V - V_1}{V_2}\right]\right\}, \tag{16}$$

$$\lambda_M(V) = \bar{\lambda}_M cosh\left[\frac{V - V_1}{2V_2}\right], \tag{17}$$

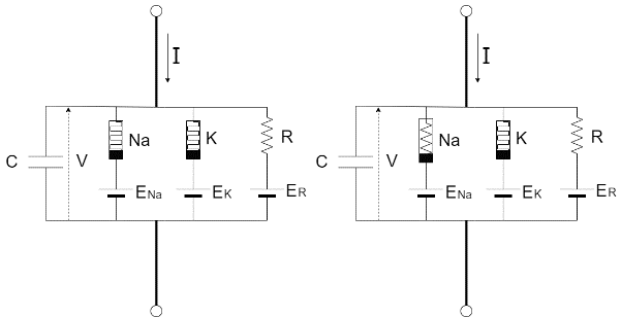
$$N_\infty(V) = \frac{1}{2}\left\{1 + tanh\left[\frac{V - V_3}{V_4}\right]\right\}, \tag{18}$$

$$\lambda_N(V) = \bar{\lambda}_N cosh\left[\frac{V - V_3}{2V_4}\right]. \tag{19}$$

Jednačine (11) i (14) se mogu pojednostaviti ukoliko se pretpostavi da je kalcijumov jonski kanal značajno brži nego kalijumov, pa će važiti $\frac{dM}{dt} = 0$. Time, jednačina (11) postaje:

$$I_{Ca} = \bar{g}_{Ca} \cdot M_\infty(V) \cdot (V - E_{Na}). \tag{20}$$

Redukovani izraz pojednostavljuje modelovanje kalcijumovog kanala pomoću nelinearnog otpornika, Slika 2b. Model poseduje parametre $\bar{g}_{Na}$, $\bar{g}_K$, g_L, E_{Ca}, E_K, E_L, C_M, V_1, V_2, V_3, V_4 i $\bar{\lambda}_N$. Vrednosti parametara datog redukovanog ML neurona su prikazane u Tabeli 2.



Slika 2. (a) Memristivni model ML neurona, (b) Redukovani memristivni model ML neurona

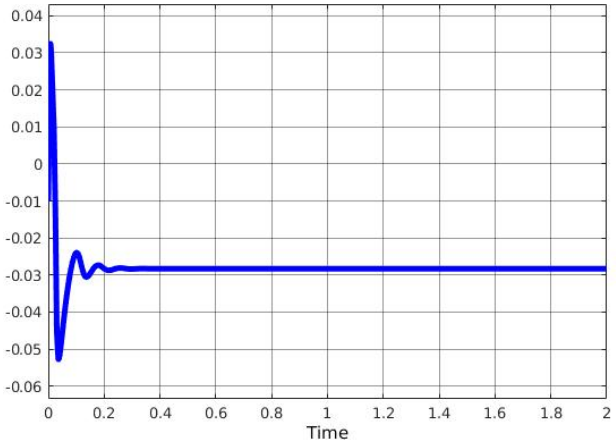
$\bar{g}_{Ca}$ $(mScm^{-2})$	4.4	E_{Ca} (mV)	120	C_M (μF)	20	V_1 (mV)	-1.2
$\bar{g}_K$ $(mScm^{-2})$	8	E_K (mV)	-84	$\bar{\lambda}_N$ (ms^{-1})	0.04	V_2 (mV)	18
g_L $(mScm^{-2})$	2	E_L (mV)	-60			V_3 (mV)	2
						V_4 (mV)	30

Tabela 2. Parametri realizovanog redukovanog ML neurona

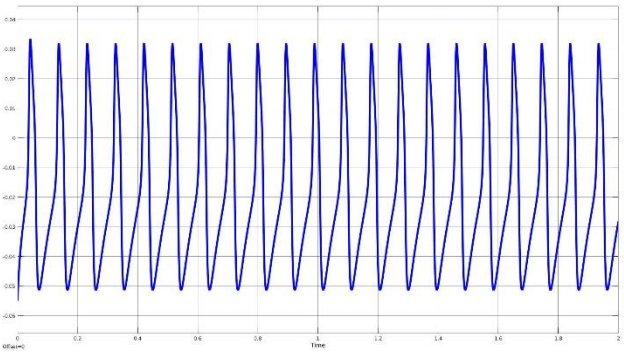
ML neuron se može naći u tri stanja ukoliko je pobuđen jediničnom pobudom

$$i(t) = I \cdot h(t), \tag{21}$$

pri čemu je $h(t)$ Hevisajdova funkcija – stabilnom, bistabilnom i nestabilnom stanju, zavisno od vrednosti konstante I i početnog stanja neurona [10]. U stabilnom stanju, napon se približava radnoj tački određenoj vrednošću $I = 85\mu A$, Slika 3. U bistabilnom stanju, napon, odnosno akcioni potencijal periodično osciluje pri vrednosti $I = 93\mu A$, Slika 4. U oba slučaja svi parametri su kao u Tabeli 2.



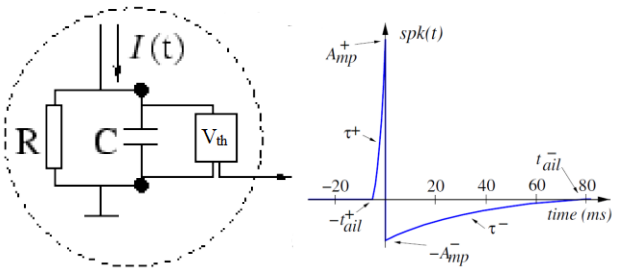
Slika 3. *Talasni oblik vremenskog signala napona strujom kontrolisanog ML neurona u stabilnom režimu*



Slika 4. *Talasni oblik vremenskog signala napona strujom kontrolisanog ML neurona u bistabilnom režimu*

LIF neuron (engl. *Leaky Integrate and Fire*) predstavlja generički (najjednostavniji i najrasprostranjeniji) model neurona i sastoji se iz RC integratora i okidačkog kola, Slika 5. Kada napon na kondenzatoru pređe određeni prag, V_{th} , okidačko kolo na svom izlazu generiše specifičnu funkciju napona koja je fitovana da odgovara neuronskom akcionom potencijalu [6]:

$$v(t) = \begin{cases} A_{mp}^{+} \dfrac{e^{t/\tau_{ail}^{+}} - e^{t_{ail}^{+}/\tau_{ail}^{+}}}{1 - e^{t_{ail}^{+}/\tau_{ail}^{+}}}, -t_{ail}^{+} < t < 0 \\ -A_{mp}^{-} \dfrac{e^{-t/\tau_{ail}^{-}} - e^{-t_{ail}^{-}/\tau_{ail}^{-}}}{1 - e^{-t_{ail}^{-}/\tau_{ail}^{-}}}, 0 < t < t_{ail}^{-} \\ 0, \quad inače \end{cases}. \tag{22}$$



Slika 5. *(a) LIF neuron* [11]
(b) funkcija akcionog potencijala [6]

Nakon okidanja, resetuje se ulazni napon na početnu vrednost. Parametri odabrani za funkciju akcionog potencijala su prikazani u Tabeli 3.

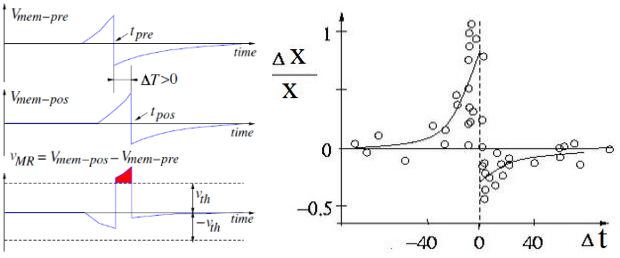
$A_{mp}^{+}(V)$	1	$t_{ail}^{+}(ms)$	5	$\tau_{ail}^{+}(ms)$	40	$V_{th}(V)$	0.9
$A_{mp}^{-}(V)$	0.25	$t_{ail}^{-}(ms)$	75	$\tau_{ail}^{-}(ms)$	3		

Tabela 3. *Parametri realizovanog LIF neurona*

4. IMPLEMENTACIJA STDP PRAVILA UČENJA NA MEMRISTIVNOJ SINAPSI

Sinapsa predstavlja strukturu koja omogućava protok električnih i hemijskih signala od jednog neurona do drugog. U ljudskom mozgu, postoji između 10^3 i 10^4 sinapsi po neuronu [6]. Sposobnost biološkog učenja predstavlja sposobnost promene (jačanja ili slabljenja) težine sinaptičkih veza između neurona [12]. Jedan od električnih elemenata koji se koristi za modelovanje sinapsi je memristor [13], gde se težina sinaptičkih veza implementira nekom promenljivom stanja, a promena vrednosti promenljive stanja postiže dovođenjem odgovarajućeg signala.

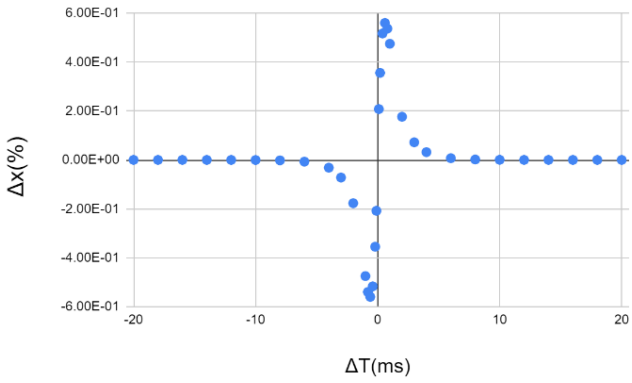
STDP (engl. *Spike Time Dependent Plasticity*) predstavlja mehanizam učenja bioloških sinapsi, na osnovu razlike u vremenu između pojave akcionih potencijala presinaptičkog i postsinaptičkog neurona [9]. Detaljnije, što je manja apsolutna vrednost razlike u vremenima okidanja $|\Delta T|$, to je veća promena težine sinapse po apsolutnoj vrednosti i obratno.



Slika 6. *(a) STDP mehanizam* [6],
(b) Procentualna promena težine u zavisnosti od ΔT [9]

Na Slici 6 se može primetiti način funkcionisanja STDP mehanizma. Što je razlika u vremenu okidanja kraća između neurona, to je promena težine sinaptičkih veza veća, odnosno povećava se površina osenčena crvenom bojom i obratno.

U *Simulink*-u je iskorišćen akcioni potencijal LIF neurona za demonstraciju funkcionisanja STDP pravila na Zamarenovom memristivnom modelu sinapse. Dobijeni rezultati su prikazani na Slici 7.



Slika 7. *STDP funkcija kod Zamarenovog memristora*

5. ZAKLJUČAK

U ovom radu je opisana softverska implementacija Moris-Lekarovog modela neurona, Zamarenovog memristivnog modela sinapse i implementacija STDP pravila učenja kod Zamarenovog memristora. Implementirani modeli se mogu koristiti u brzoj simulaciji ponašanja složenijih bioloških struktura, kao što su prirodne neuronske mreže. U master radu, iz kog je proistekao rad za ovaj zbornik, su u Matlab-u, Simulink-u i Simscape-u implementirani i drugi modeli memristora i neurona, kao što su Hodžkin-Hakslijev model neurona [14] [15], VTEAM model memristora [16] i HP model memristora [3], koji ovde nisu mogli biti prikazani zbog ograničenja dužine teksta.

6. LITERATURA

[1] M. Johnson and S. Chartier, "Spike neural models (part I): The Hodgkin-Huxley model," *The Quantitative Methods for Psychology*, vol. 13, no. 2, pp. 105-119.

[2] L. Chua, "Memristor - The Missing Circuit Element," *IEEE Transactions on Circuit Theory*, vol. 18, 5 September 1971.

[3] D. B. Strukov, G. S. Snider, D. R. Stewart and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80-83, 1 May 2008.

[4] O. Krestinskaya, A. P. James and L. O. Chua, "Neuromemristive circuits for edge computing: A review," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 1, pp. 4-23, 2019.

[5] M. Zidan, J. Strachan and W. Lu, "The future of electronics based on memristive systems," *Nature Electronics*, vol. 1, p. 22–29, 2018.

[6] C. Zamarreño-Ramos, L. A. Camuñas-Mesa, J. A. Pérez-Carrasco, T. Masquelier, T. Serrano-Gotarredona and B. Linares-Barranco, "On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex," *Frontiers in Neuroscience*, vol. 5, 17 March 2011.

[7] C. Morris and H. Lecar, "Voltage oscillations in the barnacle giant muscle fiber," *Biophysical Journal*, vol. 35, no. 1, pp. 193-213, 1 July 1982.

[8] L. Chua, "Everything You Wish To Know About Memristors But Are Afraid To Ask," *Radioengineering*, vol. 24, no. 2, pp. 319-368, 2 June 2015.

[9] J. Sjöström and W. Gerstner, "Spike-timing dependent plasticity," *Scholarpedia*, vol. 5, 2010.

[10] M. P. Sah, H. Kim, A. Eroglu and L. Chua, "Memristive Model of the Barnacle Giant Muscle Fibers," *International Journal of Bifurcation and Chaos*, vol. 26, no. 1, p. 1630001, 2016.

[11] W. Gerstner and W. M. Kistler, Spiking Neuron Models, Cambridge University Press, 2002.

[12] M. M. Adnan, S. Sayyaparaju, G. S. Rose, C. D. Schuman, B. W. Ku and S. K. Lim, "A Twin Memristor Synapse for Spike Timing Dependent Learning in Neuromorphic Systems," in *2018 31st IEEE International System-on-Chip Conference (SOCC)*, 2018 .

[13] M. P. Sah, H. Kim and L. O. Chua, "Brains Are Made of Memristors," *IEEE Circuits and Systems Magazine*, vol. 14, no. 1, pp. 12-36, 2014.

[14] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of physiology*, vol. 117, no. 4, pp. 500-544, 28 August 1952.

[15] L. Chua, V. Sbitnev and H. Kim, "Hodkin-Huxley Axon Is Made Of Memristors," *International Journal of Bifurcation and Chaos*, vol. 22, no. 3, pp. 123011--, 2012.

[16] S. Kvatinsky, M. Ramadan, E. G. Friedman and A. Kolodny, "VTEAM - A General Model for Voltage Controlled Memristors," *IEEE Transactions on Circuits and Systems*, vol. 62, no. 8, 2015.

Kratka biografija:

Vladimir Vincan je rođen i odrastao u Novom Sadu. Završio je gimnaziju „Jovan Jovanović Zmaj" i 2015. godine upisao Fakultet tehničkih nauka, smer „Energetika, elektronika i telekomunikacije". 2020. godine je odbranio diplomski rad i iste godine upisao master studije iz oblasti embeded sistema i algoritama.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/16BE12Kovacevic

INTEGRACIJA VEB I MOBILNIH APLIKACIJA UPOTREBOM SISTEMA ZA RAZMENU PORUKA

INTEGRATION OF WEB AND MOBILE APPLICATIONS USING MESSAGING SYSTEMS

Milica Kovačević, Milan Vidaković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – *Rad prikazuje razvoj veb i mobilne aplikacije uz podršku slanja notifikacija korisnicima na različitim platformama, koje je realizovano pomoću Firebase Cloud Messaging sistema. Kao primer razvijena je aplikacija HealthZone, fitnes aplikacija koja nudi korisnicima online saradnju. Ideja aplikacije je da trener kreira plan treninga za svoje klijente. Za implementaciju serverskog dela korišćen je Express framework, dok je za implementaciju klijentskog dela veb aplikacije korišćen Angular framewrok. Mobilna aplikacija implementirana je kao aplikacija za mobilne uređaje sa Android operativnim sistemom. Za skladištenje podataka korišćena je nerelaciona MongoDB baza podataka.*

Ključne reči: *JavaScript, Angular, Express, Android, MongoDB, Mongoose, FCM*

Abstract – *The aim of these paper is to develop a web and mobile application with the support of sending notifications to users on different platforms, which was realized using the Firebase Cloud Messaging system. As an example, the HealthZone application was developed, a fitness application that offers users online collaboration. The idea of the application is for the trainer to create a training plan for his clients. The Express framework was used to implement the server part, while the Angular framework was used to implement the client part of the web application. The mobile application is implemented as an application for mobile devices with the Android operating system. A non-relational MongoDB database was used for data storage.*

Keywords: *JavaScript, Angular, Express, Android, MongoDB, Mongoose, FCM*

1. UVOD

Ovaj rad će prikazati razvoj veb i mobilne aplikacije uz podršku slanja notifikacija korisnicima na različitim platformama, koje je realizovano pomoću Firebase Cloud Messaging sistema (FCM). FCM je besplatna usluga koja šalje poruke u realnom vremenu. Kao primer razvijena je aplikacija HealthZone koja spada u grupu fitnes aplikacija i nudi korisnicima online saradnju. Ideja aplikacije je da trener kreira plan treninga za svoje klijente. Svaki trening treba da bude prilagođen potrebama klijenta, njegovoj formi i rezultatima koje klijent želi da postigne. Trening sadrži detaljna uputstva sa spiskom vežbi, brojem serija, brojem ponavljanja i opciono definisanim opterećenjem koje je potrebno uključiti kako bi se postigli određeni ciljevi i vremenskim intervalom u kom se vežba izvodi ali i ono najvažnije, slike za pravilno izvođenje vežbi.

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

2. TEHNOLOGIJE

U ovom poglavlju opisane su tehnologije pomoću kojih su implementirane veb i mobilna aplikacija. Za svaku tehnologiju navedene su samo osnovne karakteristike potrebne za dalje razumevanje projektnog zadatka.

2.1. Angular

Angular je besplatan, open-source JavaScript framework [1]. Koristi se za ravoj dinamičkih veb i mobilnih aplikacija. Omogućuje kreiranje aplikacija sa jednom stranom SPA – Single page Application. Angular aplikacije pišu se u programskom jeziku TypeScript. Angular aplikacija sastoji se iz modula. Modul predstavlja način grupisanja komponenti, servisa, direktiva i drugih fajlova koji su povezani, na takav način da predstavljaju logičku celinu.

Komponenta predstavlja glavni gradivni element Angular aplikacije. Svaka Angular aplikacija sadrži najmanje jednu komponentu. Komponentu možemo smatrati posebnim pogledom aplikacije sa sopstvenom logikom i podacima. Komponenta se definiše u okviru modula u kom se nalazi, i zbog toga se pre ključne reči class koristi ključna reč export. Dekoratoru @Component() prosleđuje se objekat sa sledećim svojstvima (listing 2):

- **selector** predstavlja naziv, na osnovu koga će komponenta biti prikazana u okviru neke druge komponente.
- **templateUrl** predstavlja putanju do HTML šablon koji definiše šta će biti prikazano na stranici.
- **stylesUrl** predstavlja putanju do CSS fajla u kom se definišu stilovi koji će biti primenjeni u HTML-u.

```
import { Component } from '@angular/core';
@Component({
        selector: 'app-component',
        templateUrl:'./app.component.html',
```

```
        styleUrls: ['./app.component.css'] })

export class AppComponent {
        text = 'Hello World!';
}
```

Listing 1 – Primer root komponenete

2.2. Bootstrap

Bootstrap je najpopularniji framework otvorenog koda za izradu web sajtova i mobilnih aplikacija [2]. Baziran je na HTML i CSS šablonima za tipografiju, kreiranju formulara, dugmadi, navigacionim i ostalim komponentama, kao i opcionim JavaScript dodacima. Za razliku od Angulara, Bootstrap je fokusiran na vizuelni deo i kao takav koristi se kao dodatak, radi poboljšanja korisničkog intefejsa i bržeg razvoja aplikacije.

2.3. PrimeNG

PrimeNG sadrži bogatu kolekciju komponenata korisničkog intefejsa za Angular, koje zadovoljavalju većinu korisničkih zahteva kao što su tabele, padajući meniji, dugmad, kalendar, poruke, obaveštenja i slično [3]. PrimeNG komponente podržavaju šablone pomoću kojih je moguće prilagoditi sadržaj komponente.

2.4. Express

Express je besplatan, open-source NodeJS framework koji se koristi za brzo i jednostavno kreiranje serverskih aplikacija [5] . Pruža mehanizam za jednostavno upravljanje HTTP zahteva koje klijentska aplikacija šalje. Podržava MVC (Model-View-Controller), veoma čestu arhitekturu za dizajn veb aplikacija. Podržan je na svim platformama (eng. cross-platform) i nije ograničen na određeni operativni sistem. Express aplikacija ne dolazi sa strogom strukturom fajlova i foldera, već daje slobodu da sami organizujemo strukturu aplikacije. Struktuiranje aplikacije na domenski povezane delove je najbolji pristup, jer na taj način inkapsuliramo delove aplikacije što olakšava dalji razvoj u budućnosti.

2.5. MongoDB Atlas baza podataka

MongoDB je vodeća NoSQL baza podataka koja čuva podatke kao JSON dokumente sa dinamičkim šemama. JSON (JavaScript Object Notation) je otvoreni standard zasnovan na tekstu, osmišljen za razmenu podataka koji su pogodni za čitanje ljudima [6]. Dokument baza podataka omogućuje da podaci mogu biti smešteni u ugnježdenom obliku do proizvoljne dubine. Dokument u Mongo bazi podataka se može posmatrati kao jedan red u tabeli relacione baze podataka.

Mongoose je objektni dokument maper (eng. ODM - Object Document Mapper) koji olakšava korišćenje MongoDB-a, prevođenjem dokumenata iz MongoDB baze podataka u objekte koji se koriste u kodu. Mongoose koristi šeme za modelovanje podataka. Mongoose šema definiše strukturu podataka, predefinisane vrednosti, validatore i sl. Mongoose model je wrapper oko šeme, koji pruža intefejs bazi podataka za kreiranje, brisanje, ažuriranje, postavljanje upita i slično.

2.6. Firebase

Firebase je kompanija iz San Franciska koja pruža bekend usluge i usluge računarstva u oblaku (eng. cloud computing) [8]. Kompanija pruža veliki broj usluga za programere mobilnih i veb aplikacija, kao što su usluge baze podataka za rad u realnom vremenu i bekend usluge, hoisting, firebase autentifikacija i druge. Firebase Cloud Messaging sistem omogućuje besplatnu razmenu poruka na različitim platformama u realnom vremenu.

2.7. Android

Za programiranje Android aplikacija koristi se programski jezik Java i Android SDK. Android SDK (Software Development Kit) je biblioteka koja sadrži skup razvojnih alata za izradu aplikacije, kao što su emulator Android uređaja i alati za testiranje i debagovanje. Okruženje koje se koristi za razvoj je Android Studio. Za kreiranje korisničkog interfejsa i kreiranje konfiguracionih fajlova koristi se XML jezik.

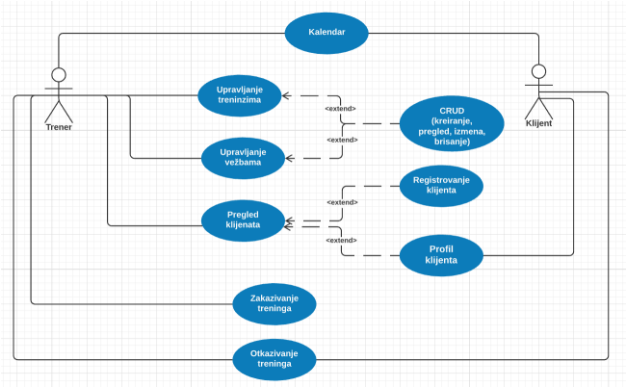
Android aplikacije se pišu u programskom jeziku Java. Prevedeni Java kod, zajedno sa svim datotekama, zapakovan je u datoteku sa sufiksom .apk. Ova datoteka služi za distribuiranje i instaliranje aplikacije na mobilnim uređajima. Aplikacijom se smatra sav kod koji se nalazi u .apk datoteci.

Prednosti Android aplikacije su te što pružaju kvalitetniji korisnički interfejs, poseduju bolje performanse i mogu upravljati resursima uređaja, za razliku od veb aplikacija koje se mogu izvršiti u veb pretraživačima uređaja.

3. SPECIFIKACIJA ZADATKA

HealthZone aplikacija spada u grupu fitnes aplikacija i nudi korisnicima online saradnju. Namenjena je personalnim trenerima i njihovim klijentima. Online trening podrazumeva da korisnik na svom uređaju ima sve podatke koji su potrebni za treniranje. Ideja aplikacije je da trener kreira plan treninga za svoje klijente. Svaki trening treba da bude prilagođen potrebama klijenta, njegovoj formi i rezultatima koje klijent želi da postigne. Trening sadrži detaljna uputstva sa spiskom vežbi, brojem serija, brojem ponavljanja i opciono definisanim opterećenjem koje je potrebno uključiti kako bi se postigli određeni ciljevi i vremenskim intervalom u kom se vežba izvodi ali i ono najvažnije, slike za pravilno izvođenje vežbi.

Na slici 1 prikazan je dijagram slučajeva korišćenja za prijavljenog trenera i klijenta.



Slika 1 – *Dijagram slučajeva korišćenja prijavljenog korisnika*

4. IMPLEMENTACIJA

Realizovani sistem obuhvata veb i mobilnu aplikaciju U narednim poglavljima biće ukratko opisane funkcionalnosti aplikacija iz ugla različitih tipova korisnika.

4.1. Veb aplikacija

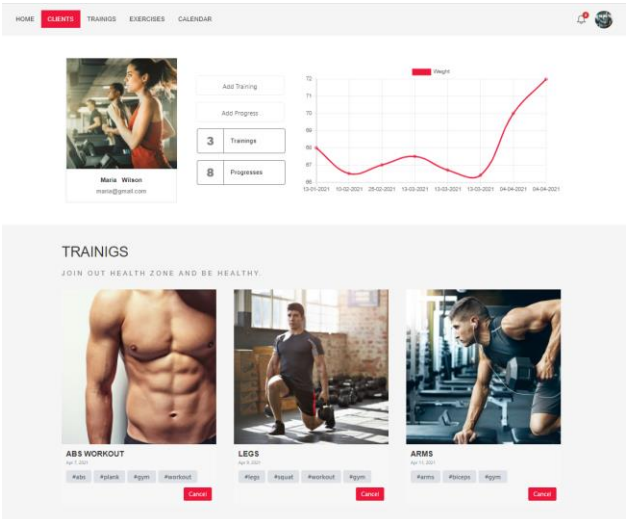
Veb aplikacija je aplikacija kojoj korisnici pristupaju preko internet mreže, koristeći veb pregledač (eng. browser). Funkcionišu tako što korisnik šalje zahteve veb serveru, putem interneta kroz veb pregledač. Veb server prosleđuje zahtev aplikaciji na serveru, koja izvršava taj zadatak, generiše rezultat i vraća odgovor nazad serveru i na kraju server prosleđuje rezultat korisniku u veb pregledaču.

HealthZone aplikacija namenjena je trenerima i kljentima. Korišćenje aplikacije omogućeno je samo registrovanim korisnicima. U zavisnosti od toga koji je tip korisnika prijavljen (trener ili klijent), navigacioni meni će sadržati različite linkove ka odgovarajućim stranicama. Trener ima opcije pregleda početne stranice ulogovanog klijenta, liste klijenata, liste treninga, liste vežbi i kalendara, dok klijent ima mogućnost da vidi početnu stranicu, svoj profil i kalendar.

Prilikom prijave na sistem serverskoj aplikaciji se šalju email adresa i lozinka. Ukoliko su uneti podaci ispravni klijentskoj aplikaciji se vraća odgovor koji sadrži token. Klijentska aplikacija token čuva u localstorage-u. Sve metode na serveru su zaštićene i nije im moguće pristupiti ako korisnik prethodno nije prijavljen na sistem, stoga je potrebno poslati token (dobijen prilikom prijave na sistem) prilikom svakog zahteva upućenog serverskoj aplikaciji.

Profil klijenta prikazan je na slici 2. Profil klijenta sadrži osnovne informacije o klijentu, opcije za zakazivanje treninga i dodavanje parametara napretka, prikaz broja zakazanih treninga, grafički prikaz parametra napretka gde su na x-osi prikazani datumi, a na y-osi numeričke vrednosti, i listu zakazanih treninga. Treninzi su prikazani u vidu kartica.

Trening sadrži osnovne informacije, naziv, sliku, datum kada se održava, tagove i dugme za opciju otkazivanja. Opcija otkazivanja treninga omogućena je klijentu i treneru. Klikom na dugme otkaži (eng. cancel), trening će biti otkazan i korisniku će se prikazati obaveštenje o uspešnosti.



Slika 2 – *Profil klijenta*

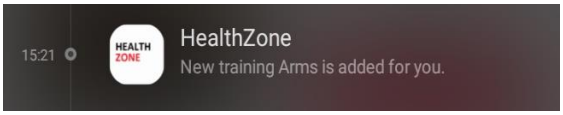
Dodavanje treninga za klijenta vrši se putem odgovarajućeg dijaloga, gde je potrebno izabrati datum i vreme, kao i trening iz padajućeg menija, koji sadrži listu prethodno kreiranih treninga za prijavljenog trenera. Ova opcija omogućena je samo treneru. Klikom na dugme potvrdi (eng. confirm), zahtev se prosleđuje serverskoj aplikaciji. Serverska aplikacija obrađuje zahtev, kreira i čuva trening u bazi podataka i nakon toga kreira notifikaciju za korisnika (slika 3).

Kreiraju se tri tipa notifikacija za klijenta za koga je trening zakazan.

Prva notifikacija obaveštava klijenta da je novi trening dodat, druga notifikacija šalje se klijentu sat vremena pred početak treninga kao podsetnik, dok se treća notifikacija šalje klijentu ukoliko je trening otkazan.

Slanje notifikacija implementirano je koristeći Firebase Cloud Messaging (FCM) koji pruža konekciju između serverske aplikacije i veb aplikacije ili mobilnog uređaja. FCM obaveštenja ili poruke može slati pojedinačnom uređaju ili grupi uređaja koji su pretplaćeni na primanje obaveštenja ili poruka.

Cloud Messaging funkcioniše tako što koristi Cloud Messaging Token, koji se čuva u bazi podataka a kreira se prilikom reistracije korisnika. Ovaj token se kasnije koristi za slanje poruka ka različitim uređajima. Brzina slanja poruka je veoma zadovoljavajuća kao i verovatnoća da će poruka biti isporučena.



Slika 3 – *Notifikacija o zakazanom treningu*

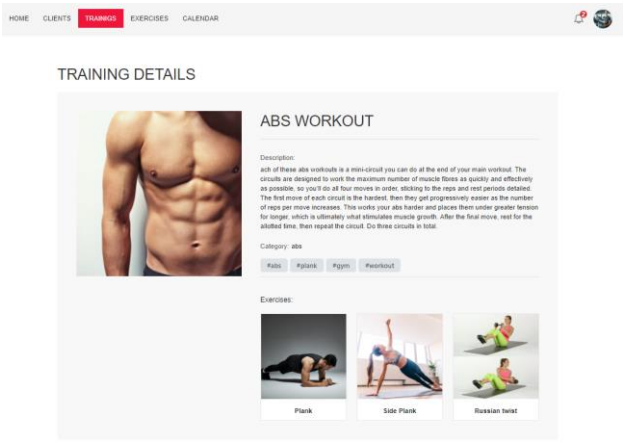
```
const notification = {
  token: registrationToken,
  title: "HealthZone",
  body: "New training "+training.name+" is added for
you."
};
firebaseAdmin.sendNotification(notification);
```

Listing 2 Deo koda koji prikazuje definisanje i slanje notifikacije nakon dodavanja treninga za klijenta

Kartica za prikaz treninga implementirana je kao generička komponenta. Koristi se jos za prikaz vežbi i rezervisanih treninga. Generička komponenta omogućuje da na osnovu ulaznih parametara koji joj se proslede prikaže različit sadržaj na ekranu. Na ovaj način prikazivanje podataka smešteno je u komponenti koja ne vodi računa o logici. Pored ulaznih parametara komponenta emituje događaje (eng. event) na klik određenog dugmeta i prosleđuje ga nazad komponenti koja je pozvala, koja dalje obrađuje taj događaj. Na taj način cela logika smeštena je u roditeljskoj komponenti.

Stranica za prikaz detalja odabranog treninga prikaza je na slici 4. Na ovoj stranici su prikazani svi detalji treninga kao i lista vežbi, koje su dodate u okviru treninga. Ova stranica dostupna je klijentima i trenerima. Klikom na karticu sa vežbom, koje su prikazane ispod osnovnih

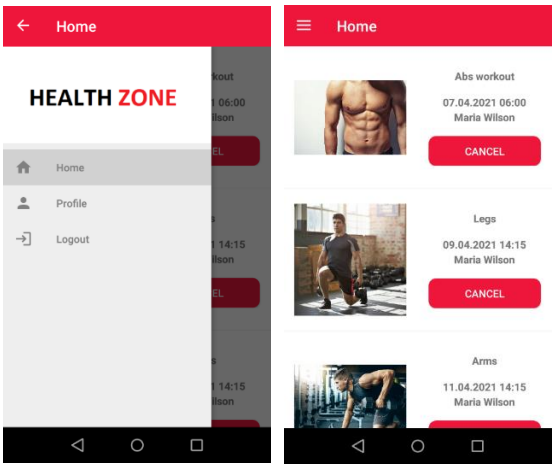
detalja o treningu, klijent će biti preusmeren na stranicu sa detaljima izabrane vežbe.



Slika 4 *Prikaz detalja treninga*

4.1. Android aplikacija

Android aplikacija namenjena je mobilnim uređajima sa ekranom osetljivim na dodir kao što su mobilni telefoni i tablet uređaji. Izvršava se na Android operativnom sistemu. HealthZone spada u grupu android aplikacija koje se opciono mogu instaliraati na mobilnom uređaju. Pristup aplikaciji omogućen je samo registrovanim korisnicima. Registrovanje korisnika može se izvršiti samo putem veb aplikacije. Android aplikacija podržava samo prijavu korisnika na sistem. Za razliku od veb aplikacije, android aplikacija podržava samo neke od funkcionalnosti sistema. Na slici 5 prikazan je navigacioni meni i početni ekran aplikacije. Navigacioni meni olakšava kretanje kroz aplikaciju. Sastoji se iz tri opcije: početni ekran, profil korisnika, i opcija za odjavu korisnika. Na početnom ekranu prikazana je lista zakazanih treninga za ulogovanog korisnika. Svaka kartica sarži samo osnovne informacije o treningu, naziv, datum i vreme održavanja i ime klijenta. Pored osnovnih informacija o treningu, postoji dugme za otkazivanje treninga.



Slika 5 Meni i početna stran

5. ZAKLJUČAK

Cilj ovog rada bio je kreiranje veb i mobilne aplikacije u okviru kojih je implementirano slanje različitih notifikacija korisnicima. Slanje notifikacija u realnom vremenu implementirano je korišćenjem Firebase Cloud Messaging sistema. Aplikacija nudi različite funkcionalnosti registrovanim korisnicima sistema za podršku online saradnje trenera i klijenta. Proširivanje i nadogradnja ove aplikacije podrazumevala bi dodavanje novih funkcionalnosti koje bi mogle unaprediti, olakšati i poboljšati saradnju između korisnika.

Aplikacije su razvijene u trenutno aktuelnim tehnologijama koje omogućuju laku i jednostavnu nadogradnju kao i održavanje. FCM sistem koji je korišćen za razmenu poruka na različitim platformama predstavlja trenutno najefikasnije rešenje za tu namenu. Stalni razvoj i održavanje pružaju mogućnost daljeg razvoja i nadogradnje kako veb tako i mobilne aplikacije.

6. LITERATURA

[1] Angular https://angular.io/docs
[2] Bootstrap https://sr.wikipedia.org/sr-el/Bootstrap
[3] PrimeNG https://www.primefaces.org/primeng
[4] Express https://expressjs.com
[5] Express/Node uvod https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction
[6] MongoDB https://sr.wikipedia.org/wiki/MongoDB
[7] Firebase https://firebase.google.com/docs
[8] Firebase-vikipedija https://sr.wikipedia.org/sr-el/Fajerbejs
[9] Android – vikipedija https://sh.wikipedia.org/wiki/Android_(operativni_sistem)
[10] Android https://developer.android.com/docs

Kratka biografija:

Milica Kovačević rođena je 29. juna 1995. godine u Novom Sadu. Završila je prirodno-matematički smer gimnazije „Jovan Jovanović Zmaj“ u Novom Sadu. Po završetku srednje škole upisuje 2014.godine Fakultet tehničih nauka u Novom Sadu, smer Računarstvo i automatika i na trećoj godini se opredeljuje za smer Primenjene računarske nauke i informatika. Diplomirala je 2018. godine i nakon toga upisuje master akademske studije na istom fakultetu na smeru za Računarstvo i automatiku, modul Elektronsko poslovanje. Položila je sve ispite predviđene planom i programom.

Milan Vidaković rođen je u Novom Sadu 1971. godine. Na Fakultetu tehničkih nauka u Novom Sadu završio je doktorske studije 2003. godine. Na istom fakultetu je 2014. godine izabran za redovnog profesora iz oblasti *Primenjene računarske nauke i informatika*.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.42:004.738.12
DOI: https://doi.org/10.24867/16BE13Jevtovic

SISTEM ZA SINHRONIZOVANU REPRODUKCIJU AUDIO SADRŽAJA GRUPAMA KORISNIKA

THE SYSTEM FOR SYNCHRONIZED PLAYBACK OF AUDIO CONTENT TO USER GROUPS

Marko Jevtović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu je predstavljen sistem za sinhronizovanu grupnu reprodukciju audio sadržaja. Aplikaciju čine server i klijent aplikacije. Server vrši audio striming ka klijentima preko RTSP protokola. Komunikacija između klijenata i servera se odvija i upotrebom REST i WebSocket standarda. Integracija pomenutih tehnologija je izvršena uz pomoć biblioteka otvorenog izvornog koda. Sprovedena je evaluacija ove aplikacije i performanse su upoređene sa sličnim rješenjima. Na osnovu toga su izvučena zaključna razmatranja ovog rada.*

Ključne reči: *sinhronizacija, reprodukcija, audio sadržaj, RTSP*

Abstract – *This paper presents the system for synchronized group playback of audio content. The application consists of the server application and the client application. The server performs audio streaming to the clients via RTSP protocol. Communication between clients and the server also takes place using the REST and WebSocket standards. The integration of these technologies is performed using the open source libraries. Evaluation of this application is conducted and performance is compared to similar solutions. Based on the evaluation results concluding considerations of this paper are drawn.*

Keywords: *synchronization, playback, audio content, RTSP*

1. UVOD

Aplikacije koje rade sa multimedijalnim sadržajem u većini slučajeva podržavaju samo pojedinačnog korisnika i podređuju sve funkcionalnosti njemu. Aplikacija koja je opisana u ovom radu i njoj slične omogućavaju da u jednom trenutku bude prisutno više korisnika kojima je potrebno pružiti iste usluge – slično kao kod radio stanica, gdje korisnici koji slušaju istu stanicu slušaju i isti sadržaj u isto vrijeme. Osim toga, u rješenju predstavljenom u ovom radu korisnici mogu direktno da utiču na sadržaj koji se reprodukuje, pa i da oni sami budu autori pjesama i plejliste. Sinhronizacija reprodukcije u realnom vremenu predstavlja izazov jer je potrebno podržati više korisnika, tačnije pružiti im isto iskustvo bez obzira u kom trenutku reprodukcije su se priključili. Rješavanje ovog problema stvara doživljaj da više osoba zajedno sluša muziku, sa različitih lokacija.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Marko Marković, docent.

2. PREGLED SLIČNIH APLIKACIJA

Postojeća rješenja u ovom domenu uglavnom predstavljaju aplikacije za mobilne uređaje. Njima se postiže da više osoba sluša isti audio sadržaj istovremeno, dok neke od njih imaju podršku i za video.

JQBX [1] je desktop i mobilna aplikacija. Oslanja se na platformu Spotify, s obzirom da koristi njen API što je dobra strana budući da je u pitanju najpoznatiji muzički servis ali sa druge strane znači da ne postoji druga opcija za reprodukciju. Radi na principu muzičkih soba koje mogu biti javne ili privatne. Unutar sobe postoje i dodatne opcije kao što su pravljenje plejliste, dopisivanje korisnika, ocjenjivanje trenutne pjesme i druge.

SoundSeeder [2] je Android aplikacija koja reprodukuje zvuk na više uređaja povezanih na istu mrežu. Podržava HTTP stream, DLNA, UPnP, ali i lokalne fajlove. Takođe, postoji opcija da se sluša neka od preko 25 000 radio stanica širom svijeta. Ova aplikacija koristi jedan uređaj kao orkestrator, i sa njega emituje sadržaj prema drugim uređajima. Na taj način se stvara sistem u kojem se glavni uređaj ponaša kao virtuelni server, a ostali uređaji kao zvučnici. Putem glavnog uređaja je moguće kontrolisati reprodukciju ali i jačinu zvuka na svim ostalim. Ova aplikacija se može pokrenuti i na desktop platformama, ali samo u režimu zvučnika, dakle bez opcije da takav sistem ima ulogu orkestratora.

AmpMe [3] je desktop i mobilna aplikacija koja je kao kombinacija dvije prethodno navedene, s obzirom na to da omogućava reprodukciju putem Spotify servisa, lokalnih fajlova ali ima podršku i za YouTube. Samim tim, pored audia, dostupan je i video sadržaj. Moguće je pridružiti se sobama u blizini zavisno od lokacije uređaja, kao i drugim javno dostupnim sobama u čitavom svijetu. Postoji i opcija povezivanja naloga sa socijalnih mreža, kako bi se olakšalo pronalaženje drugih osoba, čije profile je moguće zapratiti i tako dobijati obavještenja kada neko od njih napravi sobu.

Rave [4] je aplikacija za više platformi namjenjena korisnicima koji žele da zajedno slušaju ili gledaju sadržaj. Podržava različite striming servise, od kojih su najpopularniji YouTube, Netflix i Vimeo. Moguće je

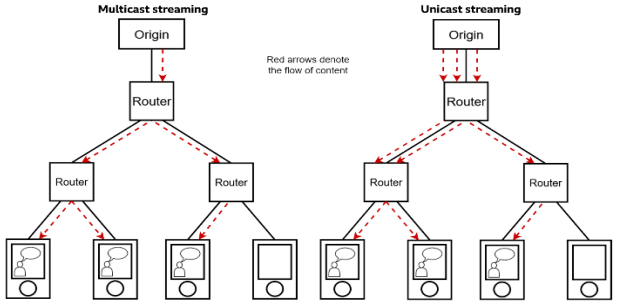
pristupiti i reprodukovati sadržaj sa DropBox ili Google Drive servisa za skladištenje. Korisnik koji napravi sobu, može da podesi da ona bude dostupna javno, samo korisnicima u blizini, samo prijateljima ili samo korisnicima koji uđu preko pozivnice tj. linka. Takođe, postoji opcija za način izbora sljedeće stavke za reprodukciju. Moguće je da korisnici o tome glasaju, može da bude automatski, ili da vlasnik sam izabere.

3. PREGLED KORIŠĆENIH TEHNOLOGIJA

RTP (Real Time Transport Protocol) [5] je transportni protokol dizajniran za saobraćaj u realnom vremenu, npr. za audio i video, prvi put objavljen 1996. godine. Obimno se koristi u sistemima za komunikaciju i zabavu koji uključuju multimedijalni striming, video pozive i konferencije, VoIP, itd. Oslanja se na UDP protokol, pa samim tim bolje toleriše manje gubitke paketa nego njihovo kašnjenje. Može da se koristi i sa TCP protokolom ali takvi slučajevi nisu česti budući da TCP mehanizmi za kontrolu grešaka mogu da uspore saobraćaj i utiču na vrijeme isporuke paketa. Može da se koristi zajedno sa RTCP protokolom.

RTCP (Real-time Transport Control Protocol) [6] je kontrolni protokol koji nadgleda strim i pruža povratne informacije o njegovom kvalitetu. Dok RTP prenosi pakete sa podacima, RTCP šalje kontrolne pakete učesnicima komunikacije. Oni obuhvataju transportne statistike i informacije kao što su brojači paketa, devijacija signala, vrijeme putovanja, i druge. Aplikacija može da koristi navedene informacije i na osnovu njih promijeni parametre strima ako je potrebno. RTCP ne obezbjeđuje enkripciju ni autentikaciju, ali je to moguće implementirati upotrebom sigurnog RTCP protokola (SRTCP), odnosno, kad je u pitanju RTP, njemu ekvivaletnog sigurnog RTP (SRTP) protokola.

RTSP (Real Time Streaming Protocol) [7] je striming protokol čija je uloga da kontroliše audio ili video strim bez potrebe za preuzimanjem fajla. Podržava operacije kao što su: Options, Setup, Teardown, Play, Pause, Record, i druge. Pritom, klijent može prije slanja neke od funkcionalnih operacija da pošalje upit serveru kako bi utvrdio koje od njih su moguće. Koristi kombinaciju protokola kao što su TCP, UDP i RTP da bi uspostavio i održao sesiju između klijenta i servera, vršio prenos podataka, itd. Može da se upotrebljava za unicast striming gdje se prenos podataka odvija između jednog klijenta i servera, kao i za multicast striming gdje se paketi sa servera propagiraju do više klijenta, kao na slici 1.



Slika 1. *Multicast i unicast striming [8]*

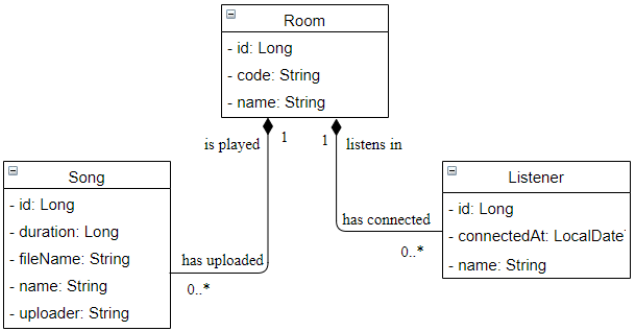
VLC (VideoLAN Client) [9] je program otvorenog koda za prikazivanje i emitovanje multimedijalnog sadržaja. Podržava veliki broj formata i protokola. Započet je kao studentski projekat 1996. godine i u početku je bio samo klijent za reprodukciju sadržaja. Danas posjeduje i značajne serverske mogućnosti kada je u pitanju striming u realnom vremenu. Formiran je od velikog broja modula, što čini lakšim dodavanje novih interfejsa, striming opcija i drugih funkcionalnosti. Podržava veliki broj operativnih sistema, na desktop i mobilnim uređajima.

VLCJ [10] je frejmvork otvorenog koda, koji omogućava integrisanje Java aplikacije sa VLC programom. Upotrebljiv je i za klijent i za server aplikacije. Nudi pristup većini VLC funkcionalnosti, od reprodukcije lokalnih fajlova do potpunog VOD servera. Enkapsulira rukovanje osnovnom bibliotekom, pri čemu štiti korisnika od nepravilne upotrebe komponenti, a i omogućuje mu proširivanje funkcionalnosti zavisno od potrebe. Npr. kod rukovanja asinhronim događajima, moguće je registrovati nove obrađivače koji će izvršiti korisničku logiku.

LibVLC [11] je biblioteka koja sadrži VLC funkcionalnosti podjeljene po modulima. Predstavlja jezgro i interfejs za multimedijalnu platformu na kojoj se VLC bazira. Postoji veliki broj verzija za desktop i mobilne platforme, pa je moguće integrisati je u različite aplikacije.

4. SPECIFIKACIJA SISTEMA

Na slici 2 je prikazan dijagram klasa koji prikazuje odnos između entiteta u implementiranom sistemu. Klasa *Room* je apstrakcija sobe i njenih podataka. Sadrži jake veze ka druge dvije klase, što je predstavljeno kompozicijom, s obzirom na to da instance ovih klasa ne mogu da postoje bez sobe. Klasa *Song* predstavlja pjesmu unutar plejliste sobe. Može da bude reprodukovana u samo jednoj sobi, dok u jednoj sobi može da bude nula ili više pjesama. Klasa *Listener* je reprezentacija korisnika, tj. slušaoca u sobi. U jednom trenutku može da bude prisutan u samo jednoj sobi, dok jedna soba ne mora da ima, ali i može, više konektovanih slušalaca.



Slika 2. *Dijagram klasa*

Svi entiteti imaju polje za jedinstveni identifikator. U osnovne podatke sobe spadaju njen kôd i ime. Slušalac takođe ima svoje ime ali i vrijeme konektovanja u sobu. Svaka pjesma pored imena ima trajanje i ime fajla u lokalnoj memoriji servera, ali i ime slušaoca koji ju je aploudovao. U ovu svrhu nije upotrebljena direktna veza na entitet *Listener* da bi se izbjegla ciklična veza između entiteta.

5. IMPLEMENTACIJA SISTEMA

U ovom poglavlju je predstavljena implementacija sistema za sinhronizovano slušanje audio fajlova na više uređaja. Čine ga klijentska i serverska aplikacija, tako da su one i njihovi moduli odvojeno objašnjeni.

5.1 Server

Ovaj dio sistema čini Spring Boot aplikacija, koja uz pomoć različitih tehnologija i protokola ispunjava zahtjeve navedene u specifikaciji sistema. Osim standardnih Spring biblioteka za veb aplikaciju, koristi još i VLCJ. Server je napisan u Kotlin programskom jeziku koji je dosta sličan programskom jeziku Java i takođe se može izvršavati pomoću JVM, ali sadrži brojna unaprijeđenja. Za čuvanje podataka se koristi relaciona baza podataka, u ovom slučaju MySQL.

REST interfejs servera omogućava neke od funkcionalnosti za klijente kao npr. dobijanje osnovnih informacija o muzičkoj sobi kojoj žele da se priključe, kreiranje nove sobe, aploud fajla na server, itd.

Kada je u pitanju WebSocket komunikacija, u sistemu postoje dva tipa poruka koje se razmjenjuju. Prvi, koji predstavlja stvarne poruke korisnika u dijelu aplikacije za dopisivanje koje iniciraju klijenti i drugi koji označava događaje u realnom vremenu, koje inicira server. Za implementaciju je iskorišćena Spring WebSocket podrška koja se bazira na STOMP protokolu.

U cilju ispunjavanja svoje glavne funkcionalnosti, sistem je u mogućnosti da fajlove koje mu pošalju klijenti strimuje ka svima koji se nalaze u datoj sobi. Pošto je moguće da postoji više soba, to znači da u jednom trenutku više strimova može da bude aktivno. Implementacija to podržava na način da se svaki strim veže za sobu, tačnije za njen jedinstven kod pa je tako i njegov URL jedinstven. Takođe je potrebno voditi računa o plejlisti za svaku sobu i o objektima koji učestvuju u strimingu. Kao osnova za audio strim je upotrebljen VLC media player, tačnije njegova implementacija u Javi u okviru VLCJ paketa.

5.2 Klijent

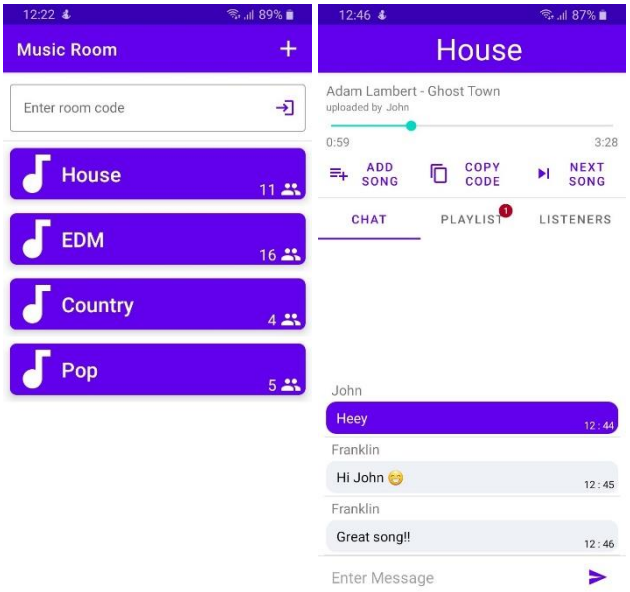
Klijentska strana je implementirana kao Android mobilna aplikacija. Takođe je, kao i server, napisana u Kotlin programskom jeziku. Neke od njegovih prednosti su povećana produktivnost, sigurniji i stabilniji kod, manje šablonskog koda, kompatibilnost sa Javom, i druge. Minimalna Android SDK verzija koju aplikacija podržava je 26, odnosno verzija sistema 8.0 Oreo što po trenutno aktuelnoj zvaničnoj statistici obuhvata oko 60% mobilnih uređaja.

Za potrebe poziva na REST API servera koristi se Retrofit klijent. On omogućava da se deklarišu metode unutar interfejsa sa posebnim anotacijama, koje pri pozivu šalju zahtjev na server. Moguće ih je izvršiti sinhrono, tj. u blokirajućem načinu tako da se čeka na odgovor, a moguće je i podesiti logiku koja će se izvršiti kada server odgovori, dakle asinhrono. Za podešavanje klijenta je potrebno napraviti Retrofit instancu koja ima definisan URL servera i koja za zadati interfejs kreira njegovu implementaciju.

Za WebSocket komunikaciju upotrebljena je STOMP Android biblioteka. Slično kao i za Retrofit i ovde je potrebno napraviti instancu klijenta koja će se koristiti za slanje i primanje poruka. Parametri koji joj se proslijeđuju su naziv klase koja služi za konekciju i URL servera.

Za konektovanje na RTSP strim i reprodukciju pjesama upotrebljena je LibVLC Android biblioteka. Komponente koje je potrebno napraviti su slične kao i za VLCJ na serveru, samo što ovde ne postoji pojam plejliste, već samo jedan strim na koji se klijent konektuje, a sadržaj koji dobije od njega reprodukuje preko zvučnika uređaja.

Na slici 3 se vidi kako izgleda početni ekran aplikacije i ekran kada se korisnik nalazi u sobi.



Slika 3. Izgled ekrana aplikacije

6. EVALUACIJA

Da bi se performanse ove i sličnih aplikacija uporedile u realnom okruženju, sprovedeno je testiranje na dva fizička Android uređaja, Samsung Galaxy A52 i Samsung Galaxy J6. Od navedenih aplikacija u drugom poglavlju, u obzir su uzete SoundSeeder i AmpMe. Evaluacija performansi Music Room i SoundSeeder aplikacija je sprovedena reprodukcijom fajlova sa uređaja, dok je za AmpMe u pitanju bio sadržaj sa YouTube platforme, pošto ista nije omogućavala korišćenje lokalnih fajlova. U nastavku je objašnjeno poređenje rada ovih aplikacija na navedenim uređajima u smislu sinhronizacije i kvaliteta reprodukcije sadržaja i količine generisanog mrežnog saobraćaja.

Sihronizacija je testirana analiziranjem zvuka koji uređaji emituju dok su konektovani na strim, prilikom reprodukcije kratkog audio snimka koji ima jednako raspoređene signale. Zvuk je zabilježen trećim uređajem, u ovom slučaju laptop računarom koji posjeduje mikrofon, a uz pomoć Audacity softvera.

Proces je tekao tako da se prvo na jednom uređaju zvuk pojača a na drugom utiša, a onda se u toku snimka izvor zvuka obrne tako što se na prvom uređaju utiša a na drugom pojača.

Na kraju se na snimku mjeri rastojanje od posljednjeg signala sa prvog uređaja do prvog signala sa drugog

uređaja. To rastojanje se zatim uporedi sa fiksnim rastojanjem koje je poznato iz originalnog snimka. Da bi se reprodukcija smatrala sinhronizovanom, potrebno je da rastojanja budu približno podudarna. Što je manja razlika između njih, to je bolja sinhronizacija. Rezultati mjerenja su prikazani u tabeli 1.

Tabela 1. Odstupanje od sinhronizacije po aplikacijama

Aplikacija	**Odstupanje u sinhronizaciji (ms)**
Music Room	78,33
SoundSeeder	31,25
AmpMe	17,67

Usaglašenost kvaliteta reprodukcije je izmjerena poređenjem zvučnih talasa na audio izlazima iz uređaja. Dakle, uređaji se konektuju na strim i svaki uređaj snima svoj zvuk. Kada se strim završi, audio zapisi se uporede tako da se vidi koji uređaj je kakav zvuk reprodukovao u kom trenutku. Rezultati mjerenja su prikazani u tabeli 2.

Tabela 2. Razlika reprodukovanog zvuka po aplikacijama

Aplikacija	**Razlika u zvuku sa uređaja (%)**
Music Room	7,71
SoundSeeder	10,19
AmpMe	5,22

Za mjerenje mrežnog saobraćaja koji aplikacija koristi na uređaju sprovedeni su testovi sa istom pjesmom u tri prethodno evaluirane aplikacije. Mjerenje protoka na mreži je specifično kada postoji više konektovanih uređaja, jer je teško izdvojiti samo podatke od interesa. Android aplikacije i servisi često u pozadini vrše određene operacije koje zahtjevaju pristup internetu, tako da je u ovom slučaju upotrebljena GlassWire aplikacija koja mjeri potrošnju za svaku aplikaciju posebno, u određenim vremenskim intervalima. Na taj način je obezbjeđeno da se prikupljaju podaci samo od aplikacija koje su predmet ove evaluacije, i to u momentu reprodukcije pjesme. Rezultati mjerenja su prikazani u tabeli 3.

Tabela 3. Količina mrežnog saobraćaja po aplikacijama

Aplikacija	**Dolazni Saobraćaj (MB)**	**Odlazni Saobraćaj (MB)**	**Ukupno (MB)**
Music Room	11,8	0,0122	11,8
SoundSeeder	35,8	1,1	36,8
AmpMe	11,2	2,3	13,5

7. ZAKLJUČAK

U ovom radu je predstavljen primjer implementacije sistema za sinhronizovano slušanje pjesama preko više uređaja koji ne moraju biti na istoj mreži. Prednosti ovakvog rješenja su što pruža brz i jednostavan pristup funkcionalnostima sistema. Nije potrebna registracija ili prijavljivanje unutar aplikacije, niti povezivanje na naloge eksternih sistema. Samim tim, ne postoje posebni preduslovi za upotrebu. U poređenju sa srodnim rješenjima, dobra strana ovog rješenja je što ne postoje ograničenja u vidu broja korisnika ili trajanja reprodukcije, kao i manja količina mrežnog saobraćaja uz kvalitet reprodukcije koji je konkurentan komercijalnim aplikacijama. Same performanse sistema zavise od hardverskih kapaciteta servera i mobilnih uređaja, ali i kvaliteta komunikacione mreže koju koriste.

Kao mane se mogu pripisati relativno zastarjeli način reprodukcije muzike kao jedina opcija, odnosno nedostatak drugih izvora sadržaja. Danas se sve više koriste muzički servisi kao što su Spotify, YouTube, Deezer, SoundCloud i drugi, koji nude bogatu kolekciju pjesama uz dosta pratećih podataka i raznih funkcionalnosti.

Reprodukcija audio fajlova polako postaje prošlost uzimajući u obzir dostupnost interneta i sadržaja na njemu. Osim toga, neka od sličnih postojećih rješenja za razliku od ovog nude i raznovrsnost u vidu protokola i tehnologija preko kojih se odvija strim, kao i platformi za koje su aplikacije dostupne.

8. LITERATURA

[1] JQBX, https://www.jqbx.fm (pristupljeno u avgustu 2021.)
[2] SoundSeeder, https://soundseeder.com (pristupljeno u avgustu 2021.)
[3] AmpMe, https://www.ampme.com (pristupljeno u avgustu 2021.)
[4] Rave, https://rave.io (pristupljeno u avgustu 2021.)
[5] What is RTP?, https://www.3cx.com/pbx/rtp (pristupljeno u avgustu 2021.)
[6] What is RTCP?, https://www.3cx.com/pbx/rtcp (pristupljeno u avgustu 2021.)
[7] Real Time Streaming Protocol, https://www.avsillc.com/real-time-streaming-protocol (pristupljeno u avgustu 2021.)
[8] Multicast and unicast streaming, https://www.bbc.co.uk/rd/blog/2019-09-forecaster-5g-mobile-interactive-content-experience (pristupljeno u avgustu 2021.)
[9] VLC media player, https://wiki.videolan.org/VLC_media_player (pristupljeno u avgustu 2021.)
[10] VLCJ, https://capricasoftware.co.uk/projects/vlcj (pristupljeno u avgustu 2021.)
[11] LibVLC, https://www.videolan.org/vlc/libvlc.html (pristupljeno u avgustu 2021.)

Kratka biografija:

Marko Jevtović rođen je u Istočnom Sarajevu 1996. god. Osnovne akademske studije završio je 2019. god. na Fakultetu tehničkih nauka na smjeru Računarstvo i automatika. Iste godine upisuje master akademske studije na programu Primenjene računarske nauke i informatika - Elektronsko poslovanje. kontakt: markojevtovic25@yahoo.com

UDK: 004.9
DOI: https://doi.org/10.24867/16BE14Batic

DUBOKO UČENJE ZA DELINEACIJU POLJOPRIVREDNIH PARCELA

DEEP LEARNING FOR FARM BOUNDARIES DELINEATION

Đorđe Batić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Najčešće korišćen izvor za dobavljanje podataka o poljoprivrednim granicama je katastar. Međutim, proces ekstrakcije katastarskih podataka je i dalje značajno manuelan proces, što neizbežno dovodi do povećane verovatnoće za grešku u merenju, kao i povećanih troškova. Jedan od mogućih načina prevazilaženja spomenutih problema je automatizacija procesa delineacije granica parcela korišćenjem satelitskih slika i metoda dubokog učenja. U ovom radu predstavljena je konvolutivna neuronska mreža za detekciju (tj. delineaciju) poljoprivrednih granica, gde se kao ulaz modela koriste multi-spektralne satelitske slike. Istrenirani model uspešno prepoznaje granice poljoprivrednih parcela i ostvaruje rezultate bolje od autora originalnog skupa podataka. Iako su ostvareni zadovoljavajući rezultati, predloženi su koraci za poboljšanje predloženog rešenja koji bi mogli dovesti do ostvarivanja robusnijih rezultata.*

Ključne reči: *CNN, Delineacija, Daljinsko istraživanje, Satelitski snimci*

Abstract – *The most common source of field boundary data is cadaster. Unfortunately, the process of extracting cadastral information has remained highly manual, which inevitably leads to proneness to errors and high costs. One of the possible ways to improve generating and updating cadastral information is automating the delineation of boundaries through satellite imagery and deep learning-based methods. This paper presents a CNN-based architecture for delineating agricultural boundaries, where multispectral satellite images are used as an input of the model. The trained model performs accurate boundary detection and achieves better results than the authors of the original dataset. Even though satisfactory results were achieved, we proposed steps to improve the proposed solution, which could increase robustness.*

Keywords: *CNN, Delineation, Remote Sensing, Satellite Imagery*

1. UVOD

Cambridge English Dictionary definiše granicu kao realnu ili imaginarnu liniju koja označava ivicu nečega. Ova definicija može biti proširena u agrikulturalnom domenu tvrdnjom da je granica lokacije gde se događa promena u tipu poljoprivredne kulture ili gde su dve slične poljoprivredne kulture odvojene prirodnom disrupcijom u terenu [1].

Informacije o opsegu granica su zabeležene i definisane u katastru. Katastar (katastarska mapa) je obiman katalog zemljišnih parcela koje predstavljaju kontinualan prostor koji je identifikovan jedinstvenim skupom homogenih imovinskih prava i geometrijskim opisom zemljišnih parcela. Literatura prepoznaje dva tipa poljoprivrednih granica: „fiksne“ i „generalne“. Fiksne granice sadrže precizno određene dimenzije, dok generalne granice predstavljaju aproksimaciju koja je podložna prostornim i vremenskim promenama. Održanje integriteta fiksnih i generalnih granica je krucijalan zadatak za pouzdanost katastarskih informacija. Međutim, kreiranje i ažuriranje katastarskih podataka je često spor, skup i proces podložan greškama, što je posledica oslanjanja na direktne ili *ground-based* tehnike merenja [2]. Ova činjenica često dovodi do nedovoljno kvalitetnog geometrijskog kvaliteta katastarskih mapa i nedoslednosti između registrovanih podataka i stvarnog stanja na terenu.

Motivisani spomenutim izazovima, interesovanje za automatske pristupe detekciji poljoprivrednih granica (tj. delineaciji) doživelo je značajan rast u naučnoj zajednici [3, 4, 5]. Pouzdana automatska detekcija poljoprivrednih granica korišćenjem satelitskih slika omogućava poverenje u tačnost katastarskih informacija i značajno smanjuje cenu definisanja i ažuriranja granica. Ovaj rad se fokusira isključivo na prirodne granice koje mogu biti detektovane sa satelitskih slika, a to uključuje najčešće: zgrade, ograde, žive ograde, zidovi, putevi, staze, određeni usevi, reke i kanali, slika 1.

Slika 1. *Primer slike iz skupa podataka (levo) i primer slike iz skupa referentnih (engl. ground-truth) podataka*

U poslednje vreme, konvolucione neuronske mreže (engl. *convolutional neural networks*, CNNs), algoritmi koji su zasnovani na dubokom učenju, beleže značajan uspeh u oblasti rešavanju problema mašinske vizije (engl. *computer vision*) vezanih za klasifikaciju slika [6]. Njihov uspeh je omogućen sposobnošću da pod uslovom skupa podataka sa dovoljnim nivoom varijabilnosti, efikasno nauče atribute (engl. *features*) različitih semantičkih nivoa. Skorija istraživanja pokazuju da povećanje dubine (broja slojeva) mreže obogaćuje hijerarhijsku semantiku naučenih atributa [7]. Međutim, u praksi, dodavanje novih slojeva na odgovarajuću duboku mrežu često dovodi do degradacije tačnosti prilikom treniranja. Kao odgovor na ovaj problem, autori u [7] predlažu tehniku rezidualnog učenja. Autori napominju da ova degradacija nije uslovljena problemima vezanim za nestajanje ili eksploziju gradijenata, koji su rešeni uvođenjem normaliza-

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

cionih slojeva, kao ni pretreniranjem mreže (engl. *overfitting*).

Motivisan rešenjem predloženim u [7], ovaj rad predstavlja arhitekturu koja primenjuje koncept rezidualnog učenja u domenu detekcije poljoprivrednih granica. Predloženo rešenje ostvaruje zadovoljavajuće rezultate i pokazuje sposobnost generalizacije nad oblastima koje sadrže poljoprivredne parcele različitog oblika, veličine i tipa.

U narednom poglavlju detaljno su predstavljena postojeća rešenja u oblasti delineacije poljoprivrednih parcela, kao i poređenje sa našim rešenjem. Arhitektura modela i korišćene tehnike opisane su u poglavlju 3, a poglavlje 4 sadrži analizu dobijenih rezultata i mogućih poboljšanja. Poglavlje 5 predstavlja zaključak rada.

2. PRETHODNA REŠENJA

U poslednje vreme, interesovanje za primenu metoda dubokog učenja u domenu delineacije poljoprivrednih parcela primećuje značajan rast [3, 4, 5]. Autori u [5] predlažu algoritam dubokog učenja za detekciju granica u okviru poljoprivrednih polja Navare regije u severnoj Španiji.

Satelitske slike su sakupljene korišćenjem SIGPAC sistema[1]. Predložena je U-Net [8] arhitektura bazirana na isečcima slika (engl. *patch-based architecture*), gde su enkoder i dekoder kreirani modifikacijom pretrenirane VGG-16 [9] mreže. Ulazi slika su preklopljeni isečci originalne RGB slike koja klasifikuje svaki piksel slike kao granicu ili ne-granicu. Kako bi obezbedili robustnost, autori predlažu korak post-procesiranja koji izvršava razne nasumične transformacije ulaznog isečka slike i vraća finalnu klasu segmentirane slike mekim glasanjem (engl. *soft voting*).

Persello i drugi [4] predlažu algoritam za delineaciju malih njiva (engl. *smallholder farms*) na području Nigerije i Malija uz pomoć satelitskih slika, korišćenjem konvolucionih mreža i kombinatornog grupisanja (engl. *combinatorial grouping*). Male njive su poljoprivredne parcele malih dimenzija koje su često karakterisane učešćem porodice u obrađivanju zemljišta. Problem detekcije granica parcele u ovom kontekstu je naročito kompleksan jer uključuje parcele malih dimenzija, neregularnog oblika i pomešanih useva, što čini da granice često budu nejasno definisane. Kako bi detektovali komplekse prostorne atribute sa satelitskih slika, autori koriste enkoder-dekoder baziranu arhitekturu i vrše predikciju granica na multi-spektralnim WorldView-2/3 slikama prikupljenim na području Nigerije i Malija.

Konačno, Masoud i drugi [3] predlažu VGG-16 arhitekturu zasnovanu na dilatacionim konvolucijama (engl. *dilated convolutions*) predloženim u [10]. Dilatacione konvolucije omogućavaju eksponencijalnu ekspanziju receptivnog polja (veličine filtera) bez gubljenja rezolucije i omogućavaju detektovanje udaljenih zavisnosti između piksela bez povećanja broja parametara. Mreža izvršava delineaciju granice na Sentinel-2 slikama 10m rezolucije slikanim nad Flevoland provincije.

[1] sigpac.mapama.gob.es

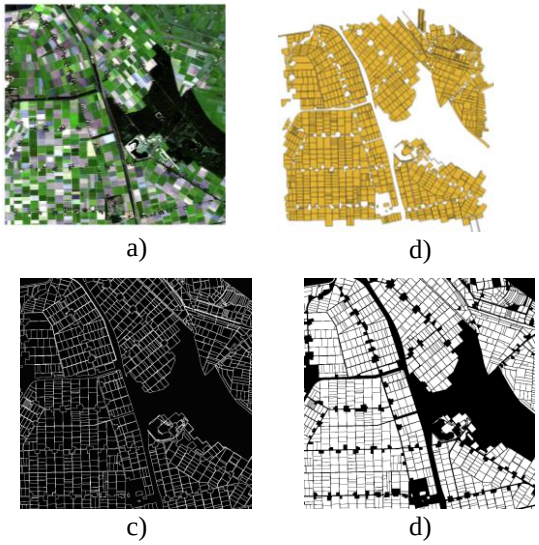
Rešenje predloženo u ovom radu oslanja se na tehnike učenja sa rezidualnim konekcijama [7]. Ulaz mreže predstavljaju multi-spektralni satelitski snimci, dok izlaz predstavlja binarna mapa koja labelira svaki piksel jednom od dve moguće klase: granica i ne-granica.

3. METOD

U narednim poglavljima izloženi su skup podataka, arhitektura i trening modela.

3.1. Skup podataka

Za treniranje i evaluaciju predloženog rešenja, korišćeni su podaci kreirani od strane autora iz [3]. Sakupljeni Sentinel-2 satelitski snimci i referentni (engl. *ground-truth*) podaci koji sadrže mapu granica parcela su sakupljeni iznad Flevoland oblasti u Holandiji. Originalne Sentinel-2. Slike sadrže 10 spektralnih kanala, četiri kanala na 10 m rezolucije (2, 3, 4, i 8 koji predstavljaju plavi, crveni, zeleni i blizu-infracrveni kanal), šest kanala na 20 m rezolucije (5, 6, 7, i 8A koji predstavljaju vegetacione kanale; i 11 i 12 SWIR kanale). Poslednja tri kanala na 60 m rezolucije (10, 11, i 12 koji predstavljaju kanale za detekciju aerosola, vodene pare i cirusa) su izostavljeni jer ne pružaju informacije relevantne za detekciju granica parcela. Ukupno, definisano je 10 poljoprivrednih oblasti (engl. *tiles*) sa parcelama, od kojih se jedna polovina koristi za treniranje, a druga za testiranje modela. Svaka oblast zauzima dimenzije od 800 x 800 piksela i površinu od 8 x 8 km^2. Primer jedne oblasti istraživanja i referentnih podataka dat je u donjem redu slike 2.



Slika 2. *Oblast istraživanja TR1: a) RGB Sentinel-2 slika oblasti b) Referentni skup podataka za oblast c) ground-truth podaci ivica parcela d) ground-truth podaci parcela*

3.2. Arhitektura

Predloženo rešenje se oslanja na arhitekture zasnovane na rezidualnom učenju [7] i ima za cilj da ekstrahuje hijerarhijske prostorne atribute multi-spektralnih ulaznih slika i transformiše ih u binarnu izlaznu sliku koja mapira granice poljoprivrednih parcela. Ulazi u mrežu predstavljaju isečke dimenzija 56 x 56 piksela, gde je za svaku od pet oblasti istraživanja (dimenzija 800 x 800 piksela) iz trening skupa generisano 1000 isečaka. Ukupno, skup podataka za obučavanje sadrži 5,000

primeraka. Ulazne slike su potom prosleđene neuralnoj mreži koja se sastoji od dva dela: enkodera i dekodera poljoprivrednih granica. Vrste operacija koje predložena enkoder arhitektura sadrži su sledeće:

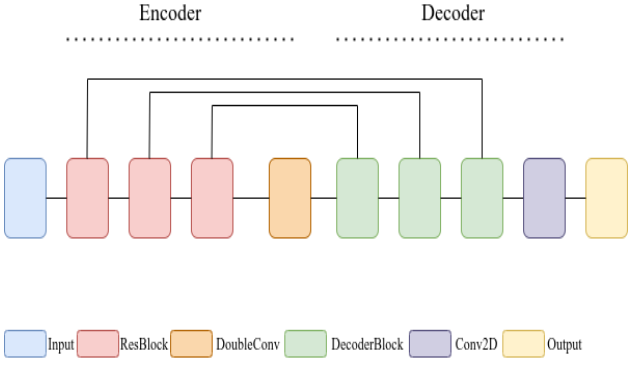
1) **Conv2D** konvolucioni sloj sa filterom veličine 3 (engl. *kernel-size = 3*) i ispunjavanjem veličine 1 (engl. *padding = 1*)
2) **Residual** ulazni podaci su obrađeni tako da generišu rezidual primenom 2-D konvolucije sa filterom veličine 1 i normalizacijom grupe (engl. *batch normalization*)
3) **DoubleConv** nad ulaznim podacima su dva puta ponovljene „Conv2D“ operacije praćene normalizacijom gupe i ReLU aktivacinom funkcijom.
4) **ResBlock** inspirisani idejom koja je predstavljena u [7], nad ulaznim podacima se primenjuje „Residual“ i „DoubleConv“ operacije čiji se izlazi sabiraju i prosleđuju ReLU aktivacionoj funkciji. Ovaj rezultat predstavlja jedan od izlaza bloka, dok je drugi izlaz rezultat „Residual“ operacije

Arhitektura enkoder dela mreže se sastoji od tri povezana „ResBlock“ bloka. Kao završni korak, nad izlazom je primenjena „DoubleConv“ operacija. Pored toga, enkoder prosleđuje i tri rezidualne reprezentacije sadržane u izlazima rezidualnih blokova (slika 3).

Naredne komponente predložene arhitekture predstavljaju dekoder koji obavlja sledeće operacije:

1) **DecoderBlock** obavlja „DoubleConv“ operaciju nad ulaznom reprezentacijom koja je konkatenirana sa izlazima odgovarajućeg rezidualnog bloka.
2) **Conv2D** generiše izlaz dekodera primenom 2-D konvolucije sa dva izlazna kanala koji predstavljaju dve moguće klase: granica i ne-granica, nad kojima je potom izvršena logaritamska *SoftMax* aktivaciona funkcija (engl. *Log-SoftMax*).

Vizuelni prikaz predložene arhitekture je prikazan na slici 3. Primene operacija smanjenja rezolucije u "ResBlock" blokovima nije dovela do povećanja tačnosti, štaviše, uzrokovala je smanjenje performansi rešenja.



Slika 3. *Vizuelni primer predložene arhitekture rešenja*

Mogući razlog ove pojave jeste da ulazni podaci ne sadrže dovoljno bogatu prostornu kompleksnost (zbog male veličine ulaznih isečaka) da bi ova tehnika bila uspešna. U finalnom koraku susedni, ravnomerno raspoređeni i neprekloplјeni isečci koji predstavljaju izlaze mreže su spojeni u jednu sliku koja odgovara oblasti istraživanja dimenzija 800 x 800 piksela.

3.3. Trening

Prilikom treniranja mreža, sve vrednosti piksela su normalizovane na vrednosti u opsegu [0, 1] kako bismo ubrzali vreme treniranje modela. Iskorišćen je Adam optimizator sa preporučenim parametrima α = 0.001, β_1 = 0.9, β_2 = 0.999. i $\varepsilon = 10^{-8}$.

Vrednost broja uzoraka (engl. *batch size*) od 32 se pokazala kao najpogodnija. Veće vrednosti stepena učenja (engl. *learning rate*) su dovele do preprilagođavanja tako da je model treniran u 20 epoha sa vrednošću stepena učenja od $3e^{-4}$.

Korišćena je *Softmax* aktivaciona funkcija nad kojom je primenjena logaritamska funkcija (*Log-Softmax*).

3.3. Procena tačnosti

Kako bismo izmerili pouzdanost rešenja za detekciju granica, korišćena je F-skor mera, dok su rezultati upoređeni sa rezultatima ostvarenim od strane autora rada [3].

Za svaku izlaznu mapu, kako bismo stekli detaljniji uvid u performanse modela, pored F-skora izračunate su i vrednosti preciznosti (engl. *precision*) i odziva (engl. *recall*).

Preciznost je definisana kao broj stvarno pozitivnih piksela koji su detektovani od strane mreže, podeljeni sa brojem svih pozitivnih detektovanih piksela:

$$p = \frac{T_{pos}}{T_{pos} + F_{pos}}$$

Odziv je mera koja je definisana brojem stvarno pozitivnih detektovanih piksela, podeljeni sa ukupnim brojem pozitivnih piksela:

$$r = \frac{T_{pos}}{T_{pos} + F_{neg}}$$

Dok je F-skor je definisan sledećom jednačinom:

$$F = 2 \cdot \frac{p \cdot r}{p + r}$$

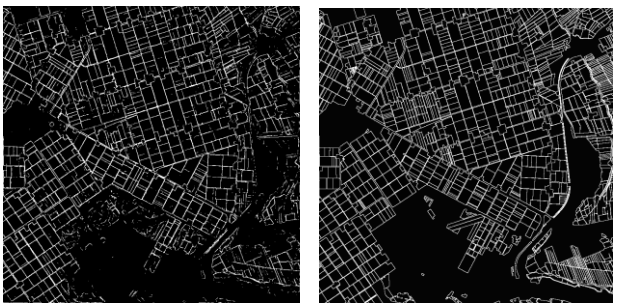
4. REZULTATI I DISKUSIJA

Evaluacijom modela utvrđeno je da dostiže veću srednju vrednost F-skor mere u odnosu na rezultate prijavljene u [3]. Vizuelni primer generisanih granica poljoprivredne parcele je prikazan na slici 4. Ostvareni rezultati su prikazani u tabeli 1.

Predloženo rešenje je sposobno da odredi poljoprivredne granice sa visokom tačnošću i pokazuje sposobnost generalizovanja kroz prepoznavanje poljoprivrednih granica nad više oblasti koje sadrže parcele različitih oblika, veličine i orijentacije.

Tabela 1. *Ostvareni rezultati modela nad pet oblasti (TS1-5) iz skupa za testiranje*

Model	F-skor za TS1	F-skor za TS2	F-skor za TS3	F-skor za TS4	F-skor za TS5	Prosečan F-skor
Masoud i drugi [3]	0.65	**0.6**	0.63	0.67	0.62	0.63
Naš model	**0.68**	**0.6**	**0.64**	**0.68**	**0.63**	**0.65**



Slika 4. *Primer izlaza iz mreže (levo) i slike iz skupa referentnih (engl. ground-thruth) podataka*

Iako rešenje predstavljeno u ovom radu ostvaruje *state-of-the-art* rezultate prilikom delineacije parcela, postoji nekolicina potencijalno značajnih poboljšanja koji mogu dovesti do povećanja tačnosti i robusnijih rezultata. Jedan od mogućih unapređenja je detekcija nad slikama iz više različitih, vremenski bliskih perioda.

Predikcija nad slikama iz više različitih datuma nam omogućuje da kreiramo konsenzus određivanjem prosečne vrednosti predikcija što ima potencijal da unapredi robusnost sistema. Sa druge strane, kvalitet ulaznih podataka se može unaprediti generisanjem kompozitnih slika na nivou meseca, čime obezbeđujemo uklanjanje oblaka i veću konzistentnost podataka.

5. ZAKLJUČAK

U ovom radu predstavljen je model koji omogućava određivanje granica (tj. delineaciju) poljoprivrednih parcela. U svrhu evaluacije pristupa korišćeni su podaci kreirani od strane autora rada [3]. Arhitektura modela bazirana je na metodama dubokog rezidualnog učenja [7]. Evaluacijom je utvrđeno da model pokazuje sposobnost uspešnog određivanja poljoprivrednih granica i generalizacije nad više oblasti koje sadrže poljoprivredne parcele različitih karakteristika. Konačno, predloženi su koraci koji mogu da omoguće dodatnu robusnost i unaprede performanse predložene arhitekture.

6. LITERATURA

[1] Rydberg A. And Borgefors G., 2001, Integrated method for boundary delineation of agricultural fields in multispectral satellite images, IEEE Trans. Geosci. Remote Sens., vol. 39, no. 11 (pp. 2514–2520)

[2] Enemark, S., Bell, K.C., Lemmen, C. and McLaren, R., 2014, Fit-For-Purpose Land Administration, International Federation of Surveyors (FIG): Copenhagen, Denmark

[3] Masoud, K.M., Persello, C. and Tolpekin, V.A. Delineation of Agricultural Field Boundaries from Sentinel-2 Images Using a Novel Super-Resolution Contour Detector Based on Fully Convolutional Networks. Remote Sens. 2020

[4] Persello C., Tolpekin V.A., Bergado J.R. and de By R.A., 2019, Delineation of agricultural fields in smallholder farms from satellite images using fully convolutional networks and combinatorial grouping, Remote Sensing of Environment, Volume 231

[5] García-Pedrero A., Lillo-Saavedra M., Rodríguez-Esparragón D. and Gonzalo-Martín C., 2019, Deep Learning for Automatic Outlining Agricultural Parcels: Exploiting the Land Parcel Identification System, IEEE Access, vol. 7 (pp. 158223-158236)

[6] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. ArXiv:1409.0575, 2014

[7] Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V. and He, Kaiming & Zhang, Xiangyu & Ren, Shaoqing & Sun, Jian., 2016, Deep Residual Learning for Image Recognition. 770-778. 10.1109/CVPR.2016.90.

[8] Ronneberger O., Fischer P., Brox T., 2015, U-Net: Convolutional Networks for Biomedical Image Segmentation, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015

[9] Simonyan, K. & Zisserman, A., 2014, Very Deep Convolutional Networks for Large-Scale Image Recognition

[10] Fisher Y. and Vladlen K., 2016, Multi-Scale Context Aggregation by Dilated Convolutions, International Conference on Learning Representations (ICLR)

Kratka biografija:

Đorđe Batić rođen je 1997. godine u Loznici. Osnovne akademske studije završio je 2020. godine na Fakultetu tehničkih nauka, na kom brani i master rad 2021. godine iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika.

kontakt: djordjebatic@gmail.com

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007:004.056
DOI: https://doi.org/10.24867/16BE17Sarenac

IDENTIFIKACIJA KOHEZIVNIH CJELINA KLASE

IDENTIFYING COHESIVE PARTS OF A CLASS

Balša Šarenac, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – *U radu je opisan algoritam pronalaska kohezivnih cjelina unutar klase. Algoritam nudi korisniku moguće načine refaktorisanja klase na dvije manje visoko kohezivne klase.*

Ključne reči: *OOP, kohezija čist kod, refaktorisanje*

Abstract – *The paper describes an algorithm that finds highly cohesive parts of a class. As a result, it offers user possible ways of refactoring given class to create two smaller more cohesive classes.*

Keywords: *OOP, cohesion, clean code, refactoring*

1. UVOD

Čist kod je jedna od najbitnijih stvari koja čine kvalitetne softverske projekte. Čist kod se može definisati kao jednostavan, čitak, razumljiv i lak za testiranje. Iako se lako može prepoznati kad je neki kod čist, pisanje čistog koda nije lako, pogotovo ne za početnike, jer je to vještina koja se uči s vremenom [1]. Upravo to je razlog zašto bi alati koji pomažu pri pisanju čistog koda bili korisni početnicima.

Jedan od načina za mjerenje kvaliteta koda je kohezija. Kohezija označava stepen povezanosti komponenti unutar nekog modula. Moduli sa visokom kohezijom su obično lakši za razvoj, održavanje, testiranje i ponovnu upotrebu [2]. U kontekstu objektno-orjentisanog programiranja, kohezija se može posmatrati na nivou klase. Tada se kohezija ogleda kao stepen povezanosti atributa i metoda. Tako je klasa visoko kohezivna ako sve metode interaguju sa svim atributima.

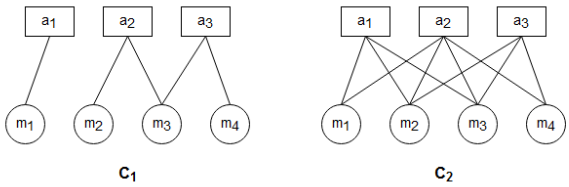
Ovaj rad predlaže rješenje koje će identifikovati kohezivne cjeline unutar neke klase. Dato rješenje nudi programeru više mogućnosti podjele klase na dvije kohezivne cjeline sa svim potrebnim informacijama za refaktorisanje.

U poglavlju 2 je kohezija detaljno opisana i neki ključni koncepti koji su usko povezani sa kohezijom. U poglavlju 3 je opisan dizajn rješenja. U poglavlju 4 je odrađena analiza implementacije, a u poglavlju 5 je zaključak.

2. KOHEZIJA

Fokus ovog rada je na analizi i računanju kohezije klasa u objektno-orjentisanom programiranju (OOP). Kohezija se može posmatrati na različitim nivoima apstrakcije. Tako se u OOP, pored kohezije klasa, može posmatrati kohezija klasa i fajlova unutar paketa.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Nikola Luburić, docent.



Slika 2.1 *Primjeri klasa predstavljenih kao grafovi*

Na primjer klase c_1 i c_2 prikazane na slici 2.1. Klase su predstavljene u obliku grafa, gdje su pravougaonici sa oznakama a_i atributi, a krugovi sa oznakama m_j su metode. Linija koja povezuje metodu sa atributom predstavlja neku interakciju (čitanje ili pisanje) metode i atributa. Na prvi pogled je jasno da je klasa c_2 kohezivnija od klase c_1.

Jedan od razloga zbog kojeg su moduli sa visokom kohezijom poželjni jeste što oni obično zadovoljavaju princip jedne odgovornosti (engl. *Single Responsibility Principle* – SRP). Jedna od definicija ovog principa je: „Okupi stvari koje se mijenjaju iz istih razloga. Razdvoji stvari koje se mijenjaju iz različitih razloga.“, što se može posmatrati i kao drugi način da se opiše kohezija. Stvari koje se mijenjaju iz istih razloga treba da imaju visoku koheziju.

Da bi se procjena stepena kohezije nekog modula automatizovala i učinila objektivnijom, pojavila se potreba za kohezivnim metrikama.

2.1. Metrike za mjerenje kohezije klase

Postoji veliki broj ovih metrika. Kako mjere kohezivnost na različite načine, dijele se po u kom mogu da računaju koheziju:

- Metrike bazirane na interfejsu – koheziju računaju tokom faze dizajna. Ove metrike koriste podatke iz definicija metoda (npr. broj parametara i tipovi parametara).
- Metrike bazirane na kodu – koheziju računaju u fazi razvoja. Ove metrike koriste podatke iz koda da bi računali koheziju (npr. pristupe metoda atributima, sadržaj komentara, itd.). Imaju dvije podgrupe:
  - Semantičke metrike – računaju koheziju na osnovu nekih koncepata izvornog koda (npr. imena identifikatora i sadržaj komentara).
  - Strukturalne metrike – koriste strukturalne podatke (npr. pristupe metoda atributima).

Generalno gledano bolje rezultate daju metrike bazirane na kodu (konkretno strukturalne metrike), jer imaju više informacija i meta-podataka na raspolaganju [3].

2.2. Poželjna svojstva

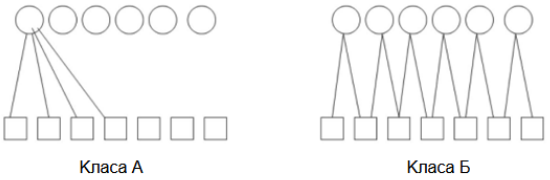
Sa velikim brojem metrika javila se i potreba za načinom da se mjeri kako teorijska tako i empirijska validnost metrika. Četiri poželjna svojstva koja su definisali Briand at al. [4] su postala standard za teorijsku validaciju:

1. Normalizacija i nenegativnost – metrike treba da budu normalizovane na interval od 0 do 1.
2. *Null* i maksimalna vrijednost – metrika treba da ima definisanu maksimalnu i *null* vrijednost.
3. Monotonost – dodavanjem konekcija unutar modula ne bi trebalo da smanji koheziju.
4. Kohezivni moduli – Kohezija modula dobijenog spajanjem dva nepovezana modula ne bi trebalo da bude veća od kohezije dva originalna modula.

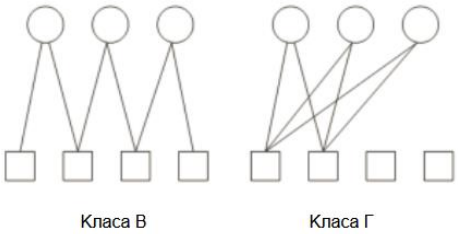
Metrike koje zadovoljavaju ova svojstva su dobro definisane [3]. Zadovoljavanje ovih svojstava je prvi korak ka definisanju kvalitetnih metrika.

2.3. LDA problem

Pored četiri poželjna svojstva, ne uzimanje u obzir interakcija unutar modula (engl. *Lack of discrimination anomaly* – LDA [5]) je još jedno svojstvo koje je bitno za kohezivne metrike. Ovo svojstvo je nepoželjno u smislu da metrike ne treba da ga ispoljavaju. LDA se ispoljava kod metrika koje ne uzimaju u obzir interakcije metoda i atributa unutar klase ili zanemaruju njihovu raspodjelu. Na slikama 2.2. i 2.3 su primjeri klasa za koje metrike koje ispoljavaju LDA ne mogu tačno odrediti koheziju.



Slika 2.2. K*lase za koje metrike ne mogu tačno odrediti koheziju ukoliko ne uzimaju u obzir veze između metoda i atributa*



Slika 2.3 *Klase za koje metrike ne mogu tačno odrediti koheziju ukoliko ne uzimaju u obzir raspored veza*

2.4 Diskusija

Metrike koje zadovoljavaju poželjna svojstva i ne ispoljavaju LDA problem generalno daju bolje rezultate od metrika koje ih ne zadovoljavaju.

Metrike za računanje kohezije su zgodne jer za kratko vrijeme daju povratnu informaciju o kodu. Međutim, velika mana strukturalnih metrika je što ne koriste i ne uračunavaju semantiku neke klase. Semantičke metrike pokušavaju da uračunaju semantiku, međutim da bi ove metrike dale dobre rezultate potrebno je da imena identifikatora budu smislena i pažljivo odabrana, tj. da su komentari smisleno napisani. Postoje neki radovi koji su pokušali da kombinuju semantičke i strukturalne metrike da dobiju bolje rezultate [6].

3. DIZAJN RJEŠENJA

Osnovna ideja je da se pomoću kohezivnih metrika odrede kohezivne cjeline unutar klase i ponude korisniku kao opcije za refaktorisanje.

Od istraženih metrika ICMBC [2] se izdvojio kao dobar kandidat. Ova metrika rekurzivno uklanja interakcije između metoda i atributa da bi dekomponovala klasu. Na osnovu [3] je utvrđeno da ICBMC zadovoljava sva 4 poželjna svojstva i ne ispoljava LDA problem. Iz navedenih razloga je ova metrika uzeta kao početna tačka implementacije.

Za razumijevanje ICBMC metrike potrebno je razumjeti metriku na osnovu koje je definisana ICBMC, a to je CBMC [7].

3.1. CBMC

Za početak će se uvesti par definicija koje će se referencirati u ostatku rada.

Definicija 3.1 Metoda klase C je **specijalna** ako je ili metoda pristupa (engl. *accessor*), metoda delegacije, konstruktor ili destruktor. Metode koje nisu specijalne su **normalne**.

Definicija 3.2 **Referentni graf** za klasu C je usmjereni graf $G = (N, A)$ gdje je:

- $N = Nv \cup Nm$, gdje su Nv i Nm skup svih atributa i skup svih normalnih metoda klase C
- $A = \{(n, v) \mid m\ čita\ ili\ piše\ v, m \in Nm, v \in N\_v\}$

Definicija 3.3 Skup ljepljivih metoda je minimalan podskup metoda bez kojih je referentni graf diskonektovan.

Definicija 3.4 Kohezija referentnog grafa predstavlja umnožak faktora povezanosti i faktora strukture. Faktor povezanosti je količnik broja ljepljivih metoda i ukupnog broja metoda. Faktor strukture je srednja vrijednost kohezije podgrafova nastalih dekompozicijom referentnog grafa.

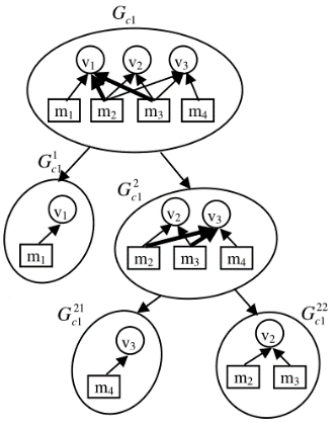
Definicija 3.5 Elementarne komponente su grafovi sa minimalno dva čvora, gdje je maksimalno može biti tačno jedana metoda ili jedan atribut.

3.2 ICBMC

ICBMC (*Improved* CBMC) metrika dekompoziciju klase vrši uklanjanjem veza između metoda i atributa, kao što slika 3.4 ilustruje.

Definicija 3.6 Skup za sječenje (*Cut set*) referentnog grafa je podskup skupa veza između metoda i atributa takav da se njihovim uklanjanjem referentni graf može podijeliti na dva manja grafa i da je taj set je minimalan (svaki njegov nadskup ne zadovoljava prvi uslov).

Definicija 3.7 Po ICBMC kohezija za referentni graf se računa kao proizvod odnosa broja veza u skupu za sječenje i ukupnog broja veza sa prosječnom vrijednošću kohezije podgrafova.



Slika 3.4. *Primjer dekompozicije klase po ICBMC*

3.2.1 Diskusija

Iako ICBMC ispravlja neke nedostatke CBMC i sama ima neke nedostatke. Naime, tokom implementacije se ispostavilo da ICBMC ipak ne zadovoljava svojstvo monotonosti [8].

Vrijeme izvršavanja je dugačko jer se na svakom nivou drveta prolazi kroz sve kombinacije veza bez ponavljanja da bi se odredio podskup veza za sječenja. Dakle to je rekurzivni $O(2^n)$ algoritam. Algoritam staje kada se pronađu prvi minimalan skup veza za sječenje, međutim dekompozicija se vrši sve do elementarnih komponenti.

Zbog svega navedenog je odlučeno da se urade određene modifikacije.

Odbačena je rekurzivna podjela grafa na podgrafove. Zadržana je osnovna ideja da se klasa pokuša izdijeliti uklanjanjem veza između atributa i metoda.

Coh [8] je uzeta kao kohezivna metrika za određivanje kohezije podgrafova, jer je za izračunavanje ICBMC potreban rekurzivni silazak.

Dalje je smanjen broj maksimalnog podskupa veza za sječenje. Inicijalno su pravljene sve kombinacije bez ponavljanja svih veza u grafu. Empirijski je utvrđeno da većina tih podskupova daju nevalidne klase ili su nadskupovi nekog drugog rješenja. Iz tog razloga je maksimalan broj veza za sječenje ograničen na: $\frac{n}{2}+1$, gdje je n ukupan broj veza.

Sa ovim modifikacijama kompleksnost algoritma je spala na $O(2^{\frac{n}{2}+1})$.

ICBMC i CBMC izbacuju sve specijalne metode iz računice, jer one interaguju sa ograničenim brojem atributa. Da taj ograničeni broj atributa ne bi narušio koheziju, specijalne metode se ne uzimaju u obzir prilikom računanja. U radu [9] je zaključeno da uključivanje specijalnih metoda prilikom predikcije klasa kojima je potrebno refaktorisanje znatno smanjuje rezultate predikcije i predloženo je da se metode pristupa i delegacije izbace, a konstruktori zadrže. Suprotno ovoj preporuci, odlučeno je da se konstruktori izbace. Konstruktori služe za inicijalizaciju atributa, samim tim nema potrebe da se uklanjaju interakcije konstruktora sa atributima. Nakon obavljenog refaktorisanja, početni konstruktor se može podijeliti na dva djela ukoliko za tim ima potrebe. Metode pristupa (čitanja i pisanja) su izbačene, a metode delegacije su ubačene. Razlog zbog kog su metode delegacije zadržane je jer nije bilo adekvatnog načina da se utvrdi da li je metoda delegator ili metoda od jedne linije koja radi neke bitne kalkulacije.

Urađeno je i poređenje vremena izvršavanja između algoritma koji koristi ICBMC i modifikovanog algoritma. Poređenje je rađeno na mašini sa Intel i7-4702Q, 2.2GHz i 16GB RAM. Vrijeme izvršavanja starog algoritma je oko 3.9 sekundi na 100 izvršavanja, dok je modifikovani algoritam trajao oko 850 milisekundi, što je 80% brže. Test je odrađen na klasi od 3 polja i 4 metode, sa 10 interakcija.

4. DIZAJN ALGORITMA

Implementacija predloženog rješenja je odrađena u sklopu platforme Clean CaDET. Ova platforma je osmišljena da pomogne programerima da pišu kvalitetniji softver. Algoritam će se koristiti prilikom evaluaciji klasa, ali kao i pomoć pri pisanju čistijeg koda.

4.1. Specifičnosti implementacije

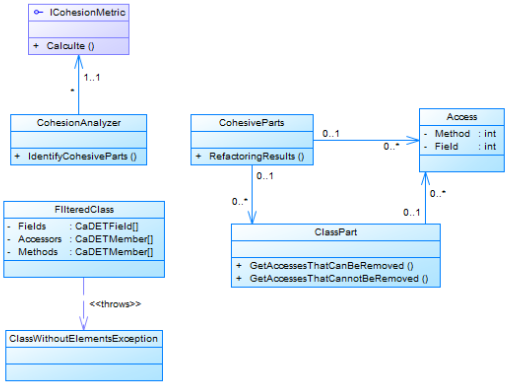
Implementacija je rađena u C# programskom jeziku. C# je specifičan jer pored klasičnih polja klasa može da sadrži i polja koja ujedno definišu u metoda pristupa i mutacije (engl. *Accessors*). Pritom je moguće odvojeno definisati polje i *accessor*-a za to polje. Zato je odlučeno da se uključuje samo *accessor*-i koji definišu osnovne metode pristupa i manipulacije. U nastavku će se pod pojmom „polja“ smatrati unija polja i ovih *accessor*-a.

4.2. Opis strukture rješenja

Na dijagramu na slici 4.1 je prikazan dijagram klasa predloženog rješenja.

Sa dijagrama na slici 4.1 se mogu uočiti iduće klase:

- *Access* – klasa koja prestavlja interakciju (čitanje ili pisanje) metode i atributa
- *ClassPart* – klasa koja je omotač za skup interakcija. Predstavlja apstrakciju klase kao skupa interakcija koje se nalaze u njoj.
- *CohesiveParts* –klasa u koju se smještaju rezultati. Sadrži listu interakcija koje treba da se izbace da bi se klasa podijelila, kao i *ClassPart* apstrakcije klasa koje će se dobiti.
- *ICohesionMetric* – Interfejs za računanje kohezivnosti nekog dijela klase (*ClassPart*-a).
- *FilteredClass* – filtrira i validira polja i metoda početne klase.
- *CohesionAnalyzer* – pronalazi kohezivne cjeline.



Slika 4.1 *Dijagram klasa predloženog rješenja*

4.3. Opis ponašanja

Algoritam prvo kreiranja *FilteredClass* instancu na osnovu *CaDETClass* instance. Klasa je validna ako ima barem jedno polje i barem jednu metodu. Idući korak je izbacivanje svih polja/metoda koja se ne koriste od strane ni jedne metode/polja. Isfiltrirana polja i metode se pretvaraju u nizove. Na osnovu njih se kasnije lako mogu namapirati interakcije na odgovarajuće metode i polja. Na osnovu *FilteredClass* instance se formira početni *ClassPart* koji sadrži sve interakcije metoda i polja.

Dalje se na osnovu početnog skupa interakcija prave manji podskupovi interakcija koji se mogu izbaciti iz klase. Oni predstavljaju kandidate pomoću kojih će se početna klasa dekomponovati.

Za svaki od podskupova se pokuša dekomponovati klasa i pronaći kohezivni dijelovi. Pronalazak kohezvinih dijelova se radi tako što se izbaci podskup interakcija i pronalaze dijelovi klase. Dijelovi se traže tako što krene od jedne interakcije i dodaju se sve interakcije povezane sa njom. Pošto interakciju predstavlja veza metode i polja, povezane interakcije su one koje sadrže ili istu metodu ili isto polje. Ukoliko se pokupe sve interakcije klasa nije diskonektovana. Ukoliko se ne pokupe sve interakcije, iz početnog skupa se izbace sve pronađene i stave se u listu za povratnu vrijednost i algoritam se opet izvršava. Pošto je ograničeno da se klasa dijeli na tačno dvije manje klase, uzimaju se samo rezultati koji su pronašli tačno dva dijela klase.

Iz rješenja se izbace sva ona koja su nadskupovi nekog drugog rješenja. Razlog je što će i svaki od tih nadskupova da podjeli klasu, ali će pri tom da dodatno ukloni još neke veze. Nakon što se isprobaju svi podskupovi kao rezultat se vrati lista svih mogućih uspješnih podjela klase. Svaku podjelu čini lista interakcija koje je potrebno ukloniti i lista novodobijenih klasa u obliku *ClassPart*-a. Lista svih uspješnih podjela klase se dodatno isfiltrira tako da u njoj ostanu samo rezultati koji sadrže visoko kohezivne cjeline.

5. ZAKLJUČAK

U ovom radu je opisan algoritam pronalaska kohezivnih cjelina klase. Dat je uvod u kohezivne metrike, kao i svojstva koja bi te metrike trebalo da zadovolje. Dalje je opisan dizajn algoritma od odabira metrike do svih modifikacija koje su primjenjene da se optimizuje vrijeme izvršavanja i kvalitet rezultata. Na kraju je kratko opisano ponašanja algoritma.

Tačke proširenja uključuju eksperimentisanje sa raznim kohezivnim metrikama ili grupama metrika koje bi dale bolje rezultate pri selekciji. Uključivanje semantičkih metrika potencijalno može povećati kvalitet rezultata.

6. LITERATURA

[1] R. C. Martin, Ур., *Clean code: a handbook of agile software craftsmanship*. Upper Saddle River, NJ: Prentice Hall, 2009.

[2] Yuming Zhou, Baowen Xu, Jianjun Zhao, и Hongji Yang, „ICBMC: an improved cohesion measure for classes“, у *International Conference on Software Maintenance, 2002. Proceedings.*, Montreal, Que., Canada, 2002, doi: 10.1109/ICSM.2002.1167746.

[3] H. Izadkhah и M. Hooshyar, „Class Cohesion Metrics for Software Engineering: A Critical Review“,

[4] L. C. Briand, S. Morasca, и V. R. Basili, „Property-based software engineering measurement“, *IEEE Trans. Softw. Eng.*, doi: 10.1109/32.481535.

[5] J. Al Dallal, „Measuring the Discriminative Power of Object-Oriented Class Cohesion Metrics“, *IEEE Trans. Softw. Eng.*, Нов. 2011, doi: 10.1109/TSE.2010.97.

[6] A. Marcus, D. Poshyvanyk, и R. Ferenc, „Using the Conceptual Cohesion of Classes for Fault Prediction in Object-Oriented Systems“, *IEEE Trans. Softw. Eng.*, Март 2008, doi: 10.1109/TSE.2007.70768.

[7] Heung Seok Chae и Yong Rae Kwon, „A cohesion measure for classes in object-oriented systems“, у *Proceedings Fifth International Software Metrics Symposium. Metrics (Cat. No.98TB100262)*, Bethesda, MD, USA, 1998, doi: 10.1109/METRIC.1998.731241.

[8] L. C. Briand, J. W. Daly, и J. Wust, „A unified framework for cohesion measurement in object-oriented systems“, у *Proceedings Fourth International Software Metrics Symposium*, Albuquerque, NM, USA, 1997, doi: 10.1109/METRIC.1997.637164.

[9] J. Al Dallal, „The impact of accounting for special methods in the measurement of object-oriented class cohesion on refactoring and fault prediction activities“, *J. Syst. Softw.*, Maj 2012, doi: 10.1016/j.jss.2011.12.006.

Kratka biografija:

Balša Šarenac je rođen 7. juna 1997. godine u Trebinju, Republika Srpska. Godine 2016. upisao je Fakultet Tehničkih nauka u Novom Sadu smer Računarstvo i automatika. 2020. godine je diplomirao sa osnovnih akademskih studija i iste godine upisuje Master studije na smeru Računarstvo i automatika.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.41
DOI: https://doi.org/10.24867/16BE18Vunic

PRIMENA TEHNOLOGIJA RAČUNARSTVA VISOKIH PERFORMANSI U ISTRAŽIVANJU BEZBEDNOSTI SAOBRAĆAJA

APPLICATION OF TECHNOLOGIES OF HIGH PERFORMANCE COMPUTING IN RESEARCH OF ROAD TRAFFIC SAFETY

Jovan Vunić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Ovaj rad opisuje postupak primene tehnologija računarstva visokih performansi u analizi faktora koji utiču na stepen ozbiljnosti saobraćajnih nesreća. U radu su navedeni upotrebljene tehnologije i algoritmi mašinskog učenja, skupovi podataka sa kojima je rađeno i primenjena metodologija uz prikazane rezultate istraživanja. Rešenja su implementirana uz pomoć programskog jezika R i sistema za distribuiranu obradu podataka Apache Spark.*

Ključne reči: *Saobraćajne nesreće, Računarstvo visokih performansi, Mašinsko učenje, Apache Spark*

Abstract – *The paper presents how technologies of high performance computing are applied in an analysis of factors affecting the severity level of road traffic accidents. The paper specifies the employed machine learning technologies and algorithms, used data sets and applied methodology with presented results of the research. The solutions are implemented using R programming language and Apache Spark engine for distributed data processing.*

Keywords: *Road traffic accidents, High performance computing, Machine learning, Apache Spark*

1. UVOD

Saobraćajne nesreće predstavljaju fenomen koji utiče ne samo na zdravstvene aspekte jedne države, već i na oblikovanje ekonomske slike društva. Bezbednost u saobraćaju predstavlja oblast koja pripada različitim naučnim krugovima, krećući se od sociologije, preko medicine, fizike i mehanike, sve do ekonomije i prava. Glavni cilj date oblasti je obezbeđivanje najvišeg mogućeg stepena sigurnosti učesnika u saobraćaju. Prema podacima *Svetske zdravstvene organizacije* [1], saobraćajne nezgode zauzimaju prvo mesto na lestvici uzroka smrtnih slučajeva među populacijom između 5 i 29 godina starosti.

Visok broj saobraćajnih nesreća često se vezuje za ekonomsko stanje države, što je usko povezano sa kvalitetom infrastrukture saobraćajnica. Sa druge strane, sve češće se javljaju sumnje i tvrdnje o povezanosti pojačanog događanja saobraćajnih nesreća sa socijalnim stanjem društva, specifičnim delovima kolovoza poput raskrsnica ili delova gde je prisutna železnička pruga, ali i prirodnim fenomenima, poput različitih vremenskih uslova, godišnjih doba i toga koji period dana je u pitanju.

Mašinsko učenje, u kombinaciji sa tehnologijama računarstva visokih performansi, pokazalo se kao grana nauke kojom se, uz pomoć odgovarajućih tehnologija, može ispitati veza između saobraćajnih nesreća sa težim posledicama po učesnike i potencijalnih uzroka takvih ishoda. U ovu svrhu korišćeni su algoritmi bazirani na stablima odlučivanja, čije osobine omogućavaju sagledavanje šire slike u kontekstu važnosti različitih atributa u odnosu na ponašanje obučenog modela, kao i algoritam *logistička regresija*. Sa obzirom na količinu korišćenih podataka, svi procesi koji se tiču mašinskog učenja obavljeni su uz pomoć odgovarajućih tehnologija računarstva visokih performansi.

2. PRETHODNA ISTRAŽIVANJA

U radu [2] diskutuje se o mogućnosti predviđanja saobraćajnih nesreća u realnom vremenu uz oslonac na različite grupe atributa, gde postoji uticaj svakoga od njih na mogućnost izazivanja nezgode u okvirima drumskog saobraćaja. Korišćeni su podaci koji se odnose na saobraćajne nesreće registrovane na teritoriji Sjedinjenih Američkih Država [2,3]. Rešenja su formirana upotrebom modela zasnovanog na dubokoj neuronskoj mreži. Kao uticajni faktori uzete su u obzir različite grupacije parametara, poput saobraćajnih elemenata, vremenskih uslova, interesnih tačaka i datumsko-vremenskih atributa. Mera F1 je upotrebljena za evaluaciju formiranih prediktivnih modela. Na osnovu predviđanja vršenog za šest odabranih gradova i geografske regione razumno velikih površina u toku kratkih vremenskih intervala, došlo se do zaključka da podaci o stanju toka saobraćaja u realnom vremenu, uz informacije o interesnim tačkama u blizini vozila, imaju uticaj na predviđanje događanja saobraćajnih nesreća u realnom vremenu.

Rad [4] bavi se evaluacijom performansi različitih modela mašinskog učenja, formiranih na osnovu algoritama *stablo odlučivanja* (engl. *Decision Tree*) i *nasumične šume* (engl. *Random Forests*) u cilju rešavanja problema predviđanja saobraćajnih nezgoda na teritoriji države Kalifornije, koja ulazi u sastav Sjedinjenih Američkih Država. Rađeno je sa skupom podataka koji obuhvata nesreće zabeležene širom Sjedinjenih Američkih Država [2,3]. Glavni cilj je bilo ispitivanje uticaja vremenskih uslova na izazivanje saobraćajne nesreće. Kao glavna mera evaluacije performansi formiranih modela

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Vladimir Ivančević.

upotrebljena je mera F1. Na osnovu dobijenih rezultata izveden je zaključak da postoji čvrsta veza između vremenskih uslova i mogućnosti događanja saobraćajnih nesreća.

3. SKUPOVI PODATAKA

Za potrebe ovog rada korišćeni su podaci sakupljeni na teritorijama dve države, Republike Srbije i Sjedinjenih Američkih Država. Podaci vezani za saobraćajne nesreće registrovane na teritoriji Republike Srbije odnose se na nesreće podeljene prema policijskim upravama i objavljeni su na Portalu otvorenih podataka Republike Srbije [5]. Dati podaci obuhvataju period između 2015. i 2021. godine. Skup podataka koji se odnosi na saobraćajne nezgode zabeležene na teritoriji Sjedinjenih Američkih Država objavljen je na sajtu Kaggle i obuhvata period između 2016. i 2020. godine [2,3].

U oba skupa podataka prisutan je atribut koji ukazuje na stepen ozbiljnosti saobraćajne nesreće. Kod podataka vezanih za teritoriju Republike Srbije dato obeležje nosi naziv *vrsta nesreće*, dok se u skupu podataka za teritoriju Sjedinjenih Američkih Država vodi pod imenom *ozbiljnost nesreće* (engl. *Severity*). Na osnovu datih atributa u oba skupa podataka formirano je novo, ciljno obeležje kategorijske prirode, nazvano *stepen ozbiljnosti nesreće* (engl. *Severity level*). Ciljno obeležje zauzima vrednosti *Sa materijalnom štetom* i *Sa fizičkom štetom* kod podataka vezanih za teritoriju Republike Srbije, odnosno *Low to Medium* i *Medium to High* kod podataka vezanih za teritoriju Sjedinjenih Američkih Država. Između ciljnih obeležja oba skupa podataka može da se povuče analogija, zato što oba ukazuju na nivo ozbiljnosti posledica konkretne saobraćajne nezgode na njene učesnike. Količina podataka vezanih za teritoriju Republike Srbije znatno je manja u odnosu na količinu podataka vezanih za teritoriju Sjedinjenih Američkih Država, gde je inicijalni odnos uzoraka 2906610 prema 195285 u korist Sjedinjenih Američkih Država. Broj sadržanih atributa takođe ide u korist Sjedinjenih Američkih Država, pošto se u skupu podataka koji se odnosi na njihovu teritoriju nalazi 47 različitih obeležja, naspram devet u okviru skupa podataka koji je vezan za nesreće registrovane u okviru Republike Srbije.

4. METODOLOGIJA

Za formiranje rešenja upotrebljena su četiri različita algoritma mašinskog učenja [6]: *logistička regresija* (engl. *Logistic regression*), *stablo odlučivanja* (engl. *Decision tree*), *nasumične šume* (engl. *Random forests*) i *stabla pojačana gradijentom* (engl. *Gradient Boosted Trees – GBT*). Poslednja tri su usko vezana za stabla odlučivanja. Za svaki od navedenih algoritama formirano je više modela za različite vrednosti odgovarajućih hiperparametara i odgovarajuće grupe prediktorskih obeležja. Grupe prediktorskih obeležja za podatke zabeležene na teritoriji Republike Srbije obuhvataju geolokacione, administrativne i sve dostupne prediktorske atribute, dok za Sjedinjene Američke Države postoji ukupno pet različitih grupa prediktorskih obeležja:

- geolokaciona obeležja,
- administrativna obeležja,
- meteorološka obeležja,
- unija geolokacionih, administrativnih i datumsko-vremenskih obeležja i
- svi dostupni prediktorski atributi.

Za *logističku regresiju* odabran je hiperparametar koji ukazuje na maksimalni broj iteracija tokom procesa obučavanja modela. Formirani su modeli za 50, 150, 300, 450 i 600 iteracija. Tokom formiranja modela zasnovanih na *stablu odlučivanja* podešavana je vrednost hiperparametra koji se odnosi na maksimalnu dubinu stabla koja može biti dostignuta tokom procesa obučavanja. Dubina stabla uzimana je u vrednostima od 5, 10, 15, 20, 25 i 30 nivoa. Algoritmom *nasumične šume* formirani su modeli tokom čijeg obučavanja se menjao broj stabala potrebnih za formiranje *nasumičnih šuma*. Modeli su formirani na osnovu *nasumičnih šuma* sastavljenih od 5, 100, 200, 300 i 400 stabala odlučivanja. Slično kao kod *logističke regresije*, za formiranje modela uz pomoć algoritma *stabla pojačana gradijentom* korišćene su različite vrednosti hiperparametra koji ukazuje na maksimalni broj iteracija tokom procesa obučavanja. Za različite modele obučavanje je trajalo u dužini od 5, 50, 100, 150 ili 200 iteracija. Postupak pretprocesiranja podataka izvršen je za podatke iz oba skupa i uključivao je odstranjivanje uzoraka kod kojih barem jedno od prediktorskih obeležja poseduje nedostajuću vrednost, kao i formiranje dodatnih datumsko-vremenskih prediktorskih atributa koji se odnose na period dana, naziv dana u sedmici i godišnje doba u okviru koga je registrovana saobraćajna nesreća.

Nad oba korišćena skupa podataka izvršeno je balansiranje podataka na osnovu ciljnog obeležja putem algoritma *ROSE* (engl. *Random Over-Sampling Examples*) [7], usled pojave visokog stepena disbalansa podataka. Obučavajući i testni skup formirani su za oba balansirana skupa podataka u razmeri 75% prema 25% polaznog skupa podataka. Procesi obučavanja i testiranja obučenih modela obavljeni su uz pomoć sistema za distribuiranu obradu velikih količina podataka *Apache Spark*, kojoj se pristupilo putem programske biblioteke *Sparklyr*, namenjene radu sa programskim jezikom R.

U evaluaciji performansi obučenih modela korišćene su sledeće metrike [8]:

- tačnost (engl. *Accuracy*),
- odziv (engl. *Recall*),
- preciznost (engl. *Precision*) i
- mera F1 (engl. *F1 measure*).

Navedene metrike računate su na osnovu matrice konfuzije formirane za svaki obučeni model.

5. REZULTATI

Formiranjem modela za sve moguće kombinacije grupa prediktorskih obeležja i vrednosti odgovarajućih hiperparametara postignut je broj od 168 formiranih modela, od toga 63 modela za podatke prikupljene na teritoriji Republike Srbije i 105 modela za podatke prikupljene na teritoriji Sjedinjenih Američkih Država. Grupe prediktorskih obeležja označavane su skraćenicama navođenim u zagradama, prema sledećoj podeli: geolokacioni atributi – (L), administrativni atributi – (A), meteorološki atributi – (V), svi dostupni prediktorski atributi – (ALL). Dodatno je uvedena grupa

prediktorskih obeležja označena sa (D) i ona predstavlja datumsko-vremenske atribute.

Pokazalo se da maksimalni broj iteracija koji može da bude dostignut tokom procesa obučavanja ne pravi uticaj na modele formirane postupkom *logističke regresije*, gde je razlika između vrednosti evaluacionih metrika za modele kojima je bilo dozvoljeno samo 50 iteracija i one kojima je bilo dozvoljeno 600 iteracija neznatna ili, u određenom broju slučajeva, nepostojeća. Geolokacioni podaci su uzeti u obzir u gotovo identičnoj meri kao podaci administrativne prirode, poput detaljnog opisa saobraćajne nesreće ili prisustva saobraćajnih elemenata u blizini događanja udesa. Na osnovu vremenskih uslova modeli formirani *logističkom regresijom* nisu uspeli da precizno odrede ozbiljnost saobraćajne nesreće i uzeti su u obzir u vidno manjoj meri u odnosu na geolokacione i administrativne atribute. Kombinacijom te dve grupe atributa i datumsko-vremenskih atributa postignuti su bolji rezultati, da bi dostigli vrhunac performansi upotrebom svih dostupnih prediktorskih obeležja.

Modeli obučavani algoritmom *stablo odlučivanja* formirani su za različite maksimalne dubine koje stablo može da dostigne u toku procesa obučavanja. Zapaženo je da različite dubine stabla imaju različit uticaj na ponašanje modela, pri čemu se vrednost mere F1 povećavala kako se dubina stabla kretala ka maksimalnoj zadatoj vrednosti. Geolokacioni i administrativni atributi su uzimani u meri sličnoj kao što je to slučaj kod modela formiranih *logističkom regresijom*. Razlika je napravljena pri upotrebi atributa vezanih za vremenske uslove, gde *stablo odlučivanja* postiže znatno bolje rezultate u odnosu na *logističku regresiju*. Upotreba svih dostupnih prediktorskih obeležja i u ovom slučaju dovodi do najboljih i najpreciznijih modela.

Formiranje modela još jednim algoritmom zasnovanim na stablima odlučivanja, *nasumičnim šumama*, rezultiralo je ishodom veoma sličnim onom koji je zabeležen kod modela formiranih postupkom *logističke regresije*. Performanse formiranih modela se nisu međusobno razlikovale u bitnoj meri, iako se inicijalni broj upotrebljenih stabala bitno razlikovao od krajnje vrednosti upotrebljene za dati hiperparametar. Sami rezultati su bolji u odnosu na *logističku regresiju* ukoliko se posmatra američki skup podataka, a lošiji ukoliko se posmatra srpski skup podataka. Kao što je to slučaj sa *stablom odlučivanja*, *nasumične šume* su pokazale bolje performanse za meteorološke atribute u odnosu na *logističku regresiju*, ali se najbolji rezultati i dalje postižu upotrebom svih dostupnih prediktorskih obeležja.

Modeli formirani uz pomoć algoritma *stabla pojačana gradijentom* pokazali su se kao najbolji i zabeležili su najbolje vrednosti evaluacionih metrika. U pitanju su modeli čije se performanse međusobno vidljivo razlikuju u odnosu na podatak o tome koliko je maksimalno iteracija bilo dozvoljeno u okviru procesa obučavanja. Za većinu grupa prediktorskih obeležja ovi modeli ne postižu rezultate koji odstupaju od rezultata koje postižu modeli obučeni prethodno navedenim algoritmima, ali prave razliku kada se radi o upotrebi svih dostupnih prediktorskih obeležja, pogotovo kada je reč o podacima prikupljenim na teritoriji Sjedinjenih Američkih Država. U tabelama 1. i 2. prikazani su, redom, rezultati evaluacije modela i vrednosti evaluacionih metrika za podatke prikupljene na teritoriji Sjedinjenih Američkih Država i Republike Srbije, pri čemu su u tabelama korišćene sledeće oznake:

- Grupa – naziv grupe prediktorskih atributa,
- LogR – logistička regresija,
- DT – stablo odlučivanja,
- RF – nasumične šume i
- GBT – stabla pojačana gradijentom.

Metrike evaluacije performansi modela označene su sledećim abrevijacijama: tačnost – A, odziv – R, preciznost – P, mera F1 – F1. U tabelama su prikazani najbolji rezultati za konkretne kombinacije grupa prediktorskih obeležja i algoritama mašinskog učenja.

Tabela 1. *Rezultati za podatke iz Sjedinjenih Američkih Država*

Grupa	LogR	DT	RF	GBT
(L)	A: 0,66017 R: 0,68314 P: 0,65452 F1: 0,66852	A: 0,65457 R: 0,63169 P: 0,66356 F1: 0,64723	A: 0,65588 R: 0,62008 P: 0,66952 F1: 0,64385	A: 0,66254 R: 0,68464 P: 0,65706 F1: 0,67057
(A)	A: 0,67361 R: 0,71321 P: 0,66218 F1: 0,68675	A: 0,67769 R: 0,66861 P: 0,68244 F1: 0,67545	A: 0,66761 R: 0,63146 P: 0,68227 F1: 0,65588	A: 0,67826 R: 0,67219 P: 0,6819 F1: 0,67701
(V)	A: 0,5566 R: 0,5769 P: 0,55594 F1: 0,56623	A: 0,60895 R: 0,67344 P: 0,59786 F1: 0,6334	A: 0,61459 R: 0,66893 P: 0,60473 F1: 0,63521	A: 0,62768 R: 0,66492 P: 0,62023 F1: 0,6418
(L, A, D)	A: 0,70354 R: 0,71912 P: 0,69871 F1: 0,70877	A: 0,70558 R: 0,70557 P: 0,70694 F1: 0,70625	A: 0,70413 R: 0,71188 P: 0,70235 F1: 0,70708	A: 0,7248 R: 0,72775 P: 0,72477 F1: 0,72626
(ALL)	A: 0,70468 R: 0,72207 P: 0,69911 F1: 0,7104	A: 0,73657 R: 0,75337 P: 0,7301 F1: 0,74155	A: 0,73368 R: 0,75355 P: 0,72596 F1: 0,7395	A: 0,7676 R: 0,79136 P: 0,75655 F1: 0,77356

Tabela 2. *Rezultati za podatke iz Republike Srbije*

Grupa	LogR	DT	RF	GBT
(L)	A: 0,63559 R: 0,60322 P: 0,6468 F1: 0,62425	A: 0,63485 R: 0,62163 P: 0,64025 F1: 0,6308	A: 0,63191 R: 0,58277 P: 0,64821 F1: 0,61375	A: 0,64674 R: 0,58281 P: 0,67021 F1: 0,62346
(A)	A: 0,68578 R: 0,56743 P: 0,74561 F1: 0,64443	A: 0,6858 R: 0,56768 P: 0,74549 F1: 0,64455	A: 0,658 R: 0,38009 P: 0,86053 F1: 0,52728	A: 0,66303 R: 0,65362 P: 0,66782 F1: 0,66064
(ALL)	A: 0,7318 R: 0,73502 P: 0,73173 F1: 0,73337	A: 0,73239 R: 0,76287 P: 0,72036 F1: 0,74101	A: 0,70745 R: 0,6942 P: 0,71466 F1: 0,70428	A: 0,74584 R: 0,72839 P: 0,75617 F1: 0,74202

Nad modelima koji su zasnovani na stablima odlučivanja primenjen je postupak određivanja važnosti atributa, čime se ispitalo koji atributi i njihove konkretne vrednosti imaju uticaj na dalje ponašanje obučenog modela.

Modeli formirani putem algoritma *stablo odlučivanja* nad podacima za Republiku Srbiju bitnim atributima smatraju geolokacione podatke, poput geografskih koordinata mesta događanja nesreće ili naziva opštine, policijske uprave ili savezne države u okviru čije teritorije se nesreća dogodila. Manji uticaj sa strane imaju faktori koji se tiču konkretnog opisa nezgode ili prisustva određenih saobraćajnih elemenata. Modeli formirani nad podacima za Sjedinjene Američke Države dodatno obraćaju pažnju na različite administrativne i datumsko-vremenske

podatke, poput naziva godišnjeg doba ili prisustva semaforizovanog pešačkog prelaza u blizini događanja nesreće.

Dato ponašanje dele i modeli formirani uz pomoć algoritama *nasumične šume* i *stabla pojačana gradijentom*, sa napomenom da modeli bazirani na *stablima pojačanim gradijentom* uzimaju u obzir meteorološke podatke u većoj meri u odnosu na modele zasnovane na preostalim algoritmima. Pokazalo se da je za određivanje ozbiljnosti nesreće uziman u obzir i podatak o tome da li je postojao sudar i, ako jeste, da li je u pitanju sudar sa parkiranim vozilom ili vozilom u kretnji. Dodatno je utvrđeno da značajan uticaj na ponašanje modela ima podatak o dužini puta na koju je nesreća direktno uticala, što se manifestuje, na primer, izazivanjem zastoja u saobraćaju.

6. ZAKLJUČAK

Ovim radom je predstavljena analiza različitih faktora koji mogu da utiču na ozbiljnost saobraćajne nesreće i samim tim na bezbednost učesnika u saobraćaju. Istraživanje je sprovedeno nad većim količinama podataka prikupljanim u toku dužeg vremenskog perioda. Iz navedenog razloga su upotrebljene tehnologije iz oblasti računarstva visokih performansi, na osnovu kojih su dalje sprovedeni procesi vezani za oblast mašinskog učenja. Prediktivni modeli formirani su na osnovu različitih algoritama baziranih na stablima – *stablo odlučivanja*, *nasumične šume*, *stabla pojačana gradijentom* – i regresionog algoritma *logistička regresija*.

Iz dobijenih rezultata se može zaključiti više stvari vezanih za uticaj različitih fenomenoloških grupa na mogućnost izazivanja saobraćajnih nezgoda sa teškim posledicama po učesnike. Pre svega treba naglasiti da se na osnovu evaluacije formiranih modela može zaključiti da nijedna zasebna grupa parametara, poput geolokacionih podataka ili informacija administrativne prirode, ne može služiti kao apsolutno pouzdani oslonac za određivanje mogućnosti događanja nesreće sa posledicama težeg oblika. Jasno je da postoji određeni uticaj lokacije vozila-učesnika, strukture dela kolovoza na kome se nesreća odvija, kao i različitih datumsko-vremenskih odrednica, ali on sam po sebi nije dovoljan za tačno i sigurno određivanje težine same nesreće, čime postaje jasno da se bezbednost u saobraćaju ne može poboljšati posmatranjem samo jedne grupe parametara. Sa druge strane, ukoliko se prethodno navedene grupe parametara ujedine i formiraju se modeli na osnovu takve unije, rezultati se poboljšavaju i dolazi se do mogućnosti nešto sigurnijeg određivanja stepena ozbiljnosti udesa.

Na osnovu podataka koji su korišćeni i modela koji su formirani zaključeno je da se tokom procesa obučavanja određeni obučavajući parametri smatraju bitnijima od ostalih, pa se tako geolokacioni podaci i jedan deo administrativnih podataka ističu u odnosu na, na primer, različite meteorološke podatke, poput vlažnosti vazduha ili brzine vetra.

7. LITERATURA

[1] World Health Organization, "*Road traffic injuries*", World Health Organization (WHO), Jun 2021. Dostupno na adresi: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries (pristupljeno u avgustu 2021.)

[2] S. Moosavi, M. H. Samavatian, S. Parthasarathy, R. Teodorescu & R. Ramnath, "*Accident risk prediction based on heterogeneous sparse data: New dataset and insights*", Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2019.

[3] S. Moosavi, M. H. Samavatian, S. Parthasarathy & R. Ramnath, "*A Countrywide Traffic Accident Dataset*", arXiv preprint arXiv:1906.05409, 2019.

[4] C. Parra, C. Ponce & S. F. Rodrigo, "*Evaluating the Performance of Explainable Machine Learning Models in Traffic Accidents Prediction in California*", 2020 39th International Conference of the Chilean Computer Science Society (SCCC), pp. 1-8, doi: 10.1109/SCCC51225.2020.9281196, 2020.

[5] "*Подаци о саобраћајним незгодама по ПОЛИЦИЈСКИМ УПРАВАМА и ОПШТИНАМА (СН) - Отворени подаци*", Data.gov.rs, 2021. Dostupno na adresi: https://data.gov.rs/sr/datasets/podatsi-o-saobratshajnim-nezgodama-po-politsijskim-upravama-i-opshtinama/ (pristupljeno u avgustu 2021)

[6] S. Shalev-Shwartz & S. Ben-David, "*Understanding Machine Learning: From Theory to Algorithms*", Cambridge: Cambridge University Press. doi:10.1017/CBO9781107298019, 2014.

[7] N. Lunardon, G. Menardi & N. Torelli, "*ROSE: a Package for Binary Imbalanced Learning*", The R Journal, vol. 6, no. 1, p. 79, 2014. Available: 10.32614/rj-2014-008.

[8] A. Zheng, "*Evaluating Machine Learning Models*", O'Reilly Media, Inc., Sep. 2015.

Kratka biografija:

Jovan Vunić rođen je u Novom Sadu 1997. godine. Završio je gimnaziju „Jovan Jovanović Zmaj" u Novom Sadu, prirodno-matematički smer. Fakultet tehničkih nauka, studijski program Računarstvo i automatika, upisao je 2016. godine. Nakon završenih osnovnih akademskih studija, upisao je master akademske studije na istom studijskom programu.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/16BE20Kosutic

RJEŠAVANJE PROBLEMA EKONOMSKOG DISPEČINGA UZ UVAŽAVANJE OBNOVLJIVIH IZVORA ENERGIJE, BAZIRANO NA GENETSKOM ALGORITMU

GENETIC ALGORITHM BASED SOLUTION FOR ECONOMIC DISPATCH PROBLEM CONSIDERING RENEWABLE ENERGY SOURCES

Stanka Košutić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – PRIMJENJENO SOFTVERSKO INŽENJERSTVO

Kratak sadržaj – *Oblast koja je analizirana u ovom radu jeste ekonomski dispečing uz uvažavanje obnovljivih izvora energije, kao i njegovo rješenje putem genetskog algoritma. Ekonomski dispečing podrazumijeva nalaženje optimalne preraspodjele opterećenja između proizvodnih agregata koji su u pogonu, sa ciljem da se postignu minimalni troškovi rada uz uvažavanje sistemskih ograničenja i uslova. Ekonomski dispečing podrazumijeva određivanje snaga proizvodnje raspoloživih generatorskih jedinica, tako da ukupni pogonski troškovi u sistemu budu minimalni.*

Ključne reči: *Ekonomski dispečing, genetski algoritam, optimizacija rada, ekonomija, generatori, obnovljivi izvori energije.*

Abstract – *The problem analyzed in this paper is economic dispatching as well as its solution through a genetic algorithm. Economic dispatching implies finding the optimal load redistribution between units in operation to achieve minimum operating costs while respecting system constraints, i.e., economic dispatching involves determining the production power of available generator units so that total operating costs in the system are minimal.*

Keywords: *Economic dispatch, genetic algorithm, optimization, economy, generators, renewable sources of energy*

1. UVOD

Proizvodnja električne energije iz konvencionalnih termoelektrana dovela je do značajnih klimatskih promjena uzrokovanih emisijom štetnih gasova. Ovaj problem, kao i ograničena količina fosilnih energetskih resursa i sve veća potreba za održivijom elektroenergetskom mrežom, doveli su do povećanja korišćenja obnovljivih izvora energije u elektroenergetskim sistemima. Javlja se potreba za izučavanjem njihovog uticaja na eksploataciju i planiranje rada elektroenergetskog sistema. Integracijom obnovljivih izvora energije proizilaze mnoge ekonomske i ekološke prednosti.

Sa druge strane, one dovode do povećanja varijabilnosti i nestabilnosti rada sistema, što povećava teškoće prilikom upravljanja. Složenost balansiranja proizvodnje i potrošnje uz uvažavanje sistemskih ograničenja i minimizacijom proizvodnih troškova, pojavom novih „zelenih“ izvora energije, predstavlja još veći izazov na koji je potrebno odgovoriti. Cilj proračuna optimalnog rada elektroenergetskog sistema je pronalaženje optimalne kombinacije proizvodnje električne energije iz pojedinih generatora, i izbora pozicije regulacione sklopke na transformatorima koji mogu da mijenjaju položaj pod opterećenjem, a sve u cilju minimizacije troškova proizvodnje termalnih jedinica, ukupne emisije gasova, kao i ukupnih gubitaka aktivne snage uz zadovoljavanje fizičkih i tehničkih ograničenja mreže. Rješavanju ovoga problema dosta pažnje je posvećeno u sektoru planiranja, upravljanja i kontrole elektroenergetskog sistema. Ova tematika se definiše kao problem angažovanja agregata koji je moguće podijeliti na dvije cjeline: izbor agregata i ekonomski dispečing.

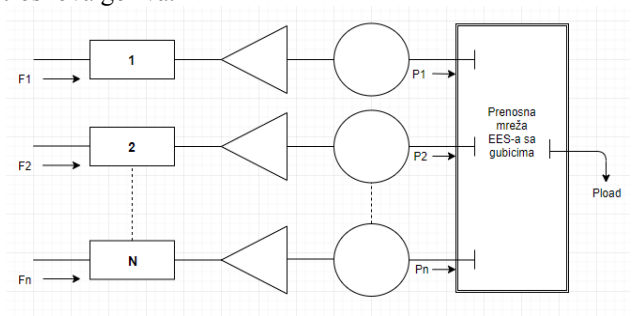
Problematika koja je analizirana u ovom radu jeste ekonomski dispečing kao i njegovo rješenje putem genetskog algoritma. Ekonomski dispečing podrazumijeva nalaženje optimalne preraspodjele opterećenja između agregata koji su u pogonu, sa ciljem da se postignu minimalni troškovi rada uz uvažavanje sistemskih ograničenja, odnosno, ekonomski dispečing podrazumijeva određivanje snaga proizvodnje raspoloživih generatorskih jedinica, tako da ukupni pogonski troškovi u sistemu budu minimalni. Preraspodjela se obično vrši na satnom nivou, sa tendencijom smanjenja vremenskog intervala porastom udjela obnovljivih izvora i razvojem tržišta električne energije.

2. EKONOMSKI DISPEČING

Da bi se održavao visok stepen ekonomičnosti i pouzdanosti elektroenergetskog sistema (EES), ekonomski dispečing je jedna od opcija dostupna preduzećima. Ekonomski dispečing (engl. ***Economic dispatch*** – *ED*) jedan je od temeljnih problema elektroenergetskog sistema čiji je cilj smanjenje ukupnih troškova proizvodnje električne energije. Ekonomski dispečing elektroenergetskog sistema je problem optimalnog raspoređivanja proizvodnje na generatorske jedinice, pri čemu se moraju zadovoljiti uslovi sigurnosti i kvaliteta rada sistema uz minimizaciju troškova proizvodnje. Ovaj algoritam podrazumijeva određivanje snaga prozvodnje tako da ukupni troškovi sistema budu minimalni. Najvažniji aspekti u velikom sistemu međusobno povezanih generatora su planirani i njihov rad treba osigurati da sistem radi ekonomično kako bi podnio

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Selakov.

zahtjevano opterećenje. Elektrane se sastoje od jednog ili više paralelno spojenih generatora. Slika 1 pokazuje sistematski dijagram paralelno povezanih N generatora sa opterećenjem [2]. Ovi generatori su povezani na sabirnicu sa opterećenjem. Glavni cilj je snadbijevanje električnom energijom sa sigurnošću i minimiziranjem troškova proizvodnih jedinica, uz zadovoljavanje svih ograničenja u sistemu. Ako sve ove jedinice imaju iste ili identične ulazno/izlazne karakteristike, opterećenje se može ravnomjerno raspodijeliti. Međutim, u praktičnim slučajevima sve ove proizvodne jedinice imaju različite ulazno/izlazne karakteristike, što implicira da je cijena svake jedinice različita. Ekonomski dispečing je važan za varijaciju snaga generatora sa generatorskim ograničenjima, tako da ove jedinice zadovoljavaju opterećenje unutar optimalnih troškova goriva.



Slika 1. *N generatorskh jedinica opslužuju opterećenje Pload uz uključivanje gubitaka u mreži*

2.1. Formulacija problema

Kriterijumska funkcija za posmatrani vremenski period za koji se traži optimalna raspodjela snaga na izlazu generatorskih jedinica, predstavljena je sumom pogonskih troškova svih agregata za koje se vrši analiza:

$$\min F_t = \sum_{i=1}^{N} F_i(P_{Gi}) \qquad (1)$$

Pogonski troškovi (troškovi rada) generatorskih jedinica se obično predstavljaju kvadratnom funkcijom. Za troškove održavanja, skladištenja, transporta i sopstvene potrošnje, smatra se da ne zavise od izlazne snage agregata, pa su oni predstavljeni slobodnim članom. Drugi dio funkcije pogonskih troškova predstavljaju troškovi proizvodnje električne energije, tj. troškovi sagorijevanja goriva, i oni zavise od izlazne snage generatora. Pogonski troškovi odgovarajuće i-te generatorske jedinice računaju se sledećom relacijom:

$$F_i(P_{Gi}) = \alpha_i * {P_{Gi}}^2 + \beta_i * P_{Gi} + \gamma_i \qquad (2)$$

Oznake:

$\boldsymbol{\alpha_i}$ [\$/Mw^2] , $\boldsymbol{\beta_i}$ [\$/MW], $\boldsymbol{\gamma_i}$ [\$] – koeficijenti pogonskih troškova

$\mathbf{P_{Gi}}$ – izlazna jednočasovna snaga i-tog generatora [MW]

$\mathbf{F_t}$ - suma troškova proizvodnih jedinica

$\mathbf{F_i(P_{Gi})}$ – troškovi i-tog generatora, kada je izlazna snaga datog generatora P_{Gi}

N – broj proizvodnih jedinica (generatora) u EES – u

Ograničenja:

Ograničenje balansne jednačine snage – ukupna proizvodnja svih generatora mora biti jednaka sumi gubitaka (P_D) i potrošnje (P_L) umanjenih za proizvodnju iz obnovljivih izvora energije (P_{RES}).

$$P_D - P_{RES} + P_L = \sum_{i=1}^{N} P_{Gi} \qquad (3)$$

Generatorska ograničenja proizvodnje – izlazna snaga svake generatorske jedinice mora biti između nominalne snage i snage koja odgovara njenom tehničkom minimum

$$P_{Gi}^{min} \leq P_{Gi} \leq P_{Gi}^{max} \qquad (4)$$

3. GENETSKI ALGORITAM

Evolucioni algoritmi su algoritmi inspirisani biološkom evolucijom, primjenjivi na širok skup problema. Pojavili su se već u pedesetim godinama 20. vijeka, te se neprestano razvijaju sve do danas. Prednost te klase algoritama je u tome što ih je moguće primijeniti u slučajevima kad ne postoji neka unaprijed određena, jasno definisana metoda za rješavanje nekog postupka ili postojeće metode imaju preveliku složenost. Jedan primjer za to je traženje optimuma funkcije više varijabli čiji analitički oblik nije poznat. Optimum bi se mogao pronaći uz iscrpnu pretragu svih mogućih kombinacija vrijednosti varijabli, međutim, takav postupak naprosto traje predugo. Princip rada genetskog algoritma se može opisati sledećim koracima [6]:

a) Definisanje svih potrebnih parametara problema
b) Inicijalizacija početne populacije
c) Određivanje prilagođenosti hromozoma (tzv. ***fitness*** funkcija)
d) Odabir selekcije hromozoma koji će opstati za ukrštanje
e) Ukrštanje – iz boljeg dijela populacije se odaberu roditelji koji će na neki način ukrstiti svoj genetski materijal i dati jednog ili više potomaka koji obično zamijene nekog od lošijih hromozoma
f) Mutacije – pri kojima se mijenja genetski sadržaj hromozoma
g) Ispitivanje konvergencije – da bi se utvrdilo da li ima osnova da se tok algoritma prekine, ukoliko nije ispunjen uslov konvergencije vrši se povratak na korak 3.

3.1. Primjena genetskog algoritma na problem ED

Kriterijumska funkcija (***fitness***) je definisana u genetskoj reprezentaciji i mjeri kvalitet predstavljenog rješenja. U implementaciji genetskog algoritma koja je opisana u ovom radu, korišćena je ***fitness*** funkcija koja mjeri kvalitet rješenja na osnovu određenih proračuna nad genima u jedinki (jednom rješenju). Rezultat funkcije je ukupan trošak proizvodnje vezan za korišćenje konkretne raspodjele opterećena na generatorskim jedinicama koje koriste OIE, gdje su uvažena sva zahtjevana ograničenja i uslovi. Cilj jeste pronaći minimalnu sumu troškova proizvodnje, odnosno, najmanju vrijednost ***fitness*** funkcije, zadovoljavajući tražene uslove. Prethodno definisani pogonski troškovi odgovarajuće i-te generatorske jedinice [3]:

$$(P_{Gi}) = \alpha_i * {P_{Gi}}^2 + \beta_i * P_{Gi} + \gamma_i \qquad (5)$$

Kriterijumskom funkcijom tražimo najbolju jedinku (rješenje – najmanji troškovi proizvodnje), tako što, za

svaku jedinku sledećom logikom provjeravamo njen kvalitet (troškovi proizvodnje):

$$FitnessFunctio = \alpha_1 * {P_{G1}}^2 + \beta_1 * P_{G1} + \gamma_1 + \ldots + \alpha_n * {P_{Gn}}^2 + \beta_n * P_{Gn} + \gamma_n \quad (6)$$

4. REZULTATI I DISKUSIJA

Genetski algoritam je korišćen za rješenje problema ekonomskog dispečinga i rezultati su analizirani. Algoritam je implementiran u *C#* programskom jeziku. Glavni cilj je minimiziranje troškova generatorskih jedinica koristeći GA, uz zadovoljavanje ograničenja i isporučivanje zahtjevane snage.

Jedna jedinka je opisana putem vrijednosti ***fitness***-a (rezultat funkcije prilagođavanja) i gena. Sadrži metode za mutaciju i ukrštanje.

Ukrštanje se vrši metodom ***flip coin*** – ako je vrijednost <= 0.5 uzima se gen jednog roditelja, a ako je veća uzima se gen drugog roditelja, nakon toga potomci se dodaju u populaciju.

U svakoj novonastaloj jedinki postoji vjerovatnoća da se dogodi mutacija. Korišćena je uniformna mutacija – vrijednost određenog gena mijenja se sa nasumično odabranom vrijednošću koja se nalazi u granicama određenim za dotični gen. Cilj mutacije jeste izbjegavanje lokalnih optimuma i obnavljanje izgubljenog genetskog materijala.

Jedinka se sastoji od skupa parametara – gena. Rješenje je testirano za elektroenergetski sistem od tri generatorske jedinice, tako da, jedna jedinka ima tri gena (jedan gen – jedan generator).

Svaki od gena predstavlja vrijednost izlazne snage generatora, gdje se za izlazni napon postavlja slučajno (random) generisana vrijednost u opsegu dozvoljene minimalne i maksimalne snage generatora.

Odabrani geni (izlazne snage) se množe sa odgovarajućim troškovnim koeficijentima za pojedini generator, i kao suma tih vrijednosti dobiju se ukupni troškovi proizvodnje za tu kombinaciju izlaznih snaga (jedinka).

Računanje ukupnih troškova, odnosno kvalitet jedne jedinke, računa se u kriterijumskoj funkciji. Kriterijumska funkcija je funkcija „prilagodljivosti", koja određuje sposobnost jedinke da se takmiči sa drugim jedinkama. Vjerovatnoća da će jedinka biti odabrana za reprodukciju bazira se na funkciji prilagodljivosti.

4.1. Test slučaj 1

U testu je analiziran sistem sa tri generatorske jedinice, gde su parametri prikazani u Tabeli 1.

U Tabeli 2 su prikazani rezultati optimizacije putem genetskog algoritma. Za isporučivanje zahtjevane snage potrošačima koja iznosi 300MW, pronađeno je najbolje rješenje raspodjele opterećenja na generatorima tako da troškovi budu najmanji (sa uračunatim gubicima).

Uočen je veliki broj formiranih generacija (oko 250), odnosno veliki broj kreiranih jedinki (oko 1000).

Tabela 1. *Podaci o generatorima*

Generator	α	β	γ	P_{min}	P_{max}
1	0.00525	8.66	328.13	100	300
2	0.00609	10.040	136.91	50	200
3	0.00592	10.040	59.16	10	90

Tabela 2 – *Rezultati algoritma*

	Genetski algoritam
P1	149.8
P2	117.6
P3	42.6
Zahtjevana snaga	300.0
Gubici	10.5
Ukupni troškovi	3627.511

4.2. Test slučaj 2

U testu je analiziran sistem sa tri generatorske jedinice, gdje su parametri prikazani u Tabeli 1. U Tabeli 2 su prikazani rezultati optimizacije putem genetskog algoritma. Za isporučivanje zahtjevane snage potrošačima koja iznosi 300MW, pronađeno je najbolje rješenje raspodjele opterećenja na generatorima tako da troškovi budu najmanji (sa uračunatim gubicima).

Tabela 3. *Podaci o generatorima*

Generator	α	β	γ	P_{min}	P_{max}
1	0.00777	13.67	45.12	50	300
2	0.00111	2.04	434.12	50	200
3	0.00333	30.69	242.16	15	120

Tabela 4. *Rezultati algoritma*

	Genetski algoritam
P1	251.1
P2	228.4
P3	44.0
Zahtjevana snaga	500.0
Gubici	23.5
Ukupni troškovi	6524.453

Potrebno vrijeme za pronalazak rješenja jeste oko 25s. Jedan od razloga za sporiji rad sa ovim podacima jeste i veća mogućnost izbora random vrijednosti gena u početnoj populaciji (veći dozvoljeni opsezi snage generatora). U ovom slučaju je potrebno isporučiti snagu od 500MW, a najmanji troškovi koje je algoritam pronašao za isporučivanje ove snage iznose 6524.453$. Malo veći broj jediniki je kreiran nego u prošlom slučaju (oko 1200).

4.3. Test slučaj 3

U ovom test slučaju iskorišćeni su podaci iz prethodnog testa. Razlika jeste u broju formiranja jedinki. Umjesto 4 jedinke, u svakoj iteraciji nastaje novih 100 jedinki. Ubrzan je postupak dolaska do rješenja, i sada iznosi oko 10s. Ako je jedinki premalo, genetski algoritam ima slabe mogućnosti izvođenja ukrštanja i mala veličina prostora pretrage otežava pronalaženje optimalnog rješenja problema. S druge strane, prevelik broj hromozoma može da uspori rad genetskog algoritma.

4.4. Prednosti i mane genetskog algoritma

Prednosti: proizvoljni optimizacijski problem, puno mogućnosti nadogradnje, ponavljanje postupka rješavanja, daje skup potencijalnih rješenja i traži najbolje, jednostavnost izvođenja.

Mane: često potrebno prilagoditi problem ili algoritam, veliki uticaj parametara, priroda rješenja je nepoznata, sporost proračuna.

5. ZAKLJUČAK

Kako će porasti upotreba obnovljivljivih izvora energije u budućnosti, važno je provesti proučavanje uticaja njihove upotrebe na rad elektroenergetskog sistema, posebno na problem ekonomskog dispečinga. Ovaj rad predstavlja izvještaj o početnoj studiji koja se fokusira na upotrebu genetskog algoritma za detaljno ispitivanje ovoga problema. Dokazano je da uključivanje obnovljivih izvora energije u elektroenergetski sistem ima uticaja na troškove, varijabilnosti opterećenja i proračuna rezervi.

Jedan od predstavnika evolucijskih algoritama – genetski algoritam, je prilagođen rješavanju konkretnog problema – ekonomskog dispečinga u elektroenergetskom sistemu. Glavni cilj jeste olakšan i pouzdan proces pronalaska najboljeg rješenja za optimalnu raspodjelu opterećenja između generatorskih jedinica, tako da ukupni pogonski troškovi budu minimalni. U ovom radu su analizirani različiti faktori koji utiču na novac koji se utroši prilikom proizvodnje električne energije u elektranama koje koriste obnovljive izvore energije. Analizirani su procenti iskorišćavanja različitih obnovljih izvora energije, prednosti i manje njihovog korišćenja, kao i troškovi i ulaganja potrebna za dobijanje električne energije iz istih. Brzom pretragom i kombinacijom različitih parametara, uz zadovoljavanje svih potrebnih ograničenja i uslova, genetski algoritam traži najbolje rješenje koje predstavlja najmanje troškove proizvodnje.

U budućnosti će se preduzimati sveobuhvatniji posao, koji će razmotriti nekoliko aspekata, kao što su: proračun emisije štetnih gasova, gubici u prenosu, realniji model elektroenergetskog sistema, prognozu snage vjetra, kao i troškove. Poređenje sa drugim metodama će takođe biti od velike važnosti za mjerenje efikasnosti ove metode u rješavanju problema.

6. LITERATURA

[1] Fraunhofer ISE, „Levelized Cost of Electricity; Renewable Energy Technologies“, Njemačka, studeni 2013.

[2] Power Generation, Operation and Control

[3] SOLVING ECONOMIC DISPATCH PROBLEM USING HYBRID GA-PS-SQP METHOD, Jamal S. Alsumait, Student Member, Jan K. Sykulski, Senior Member

[4] https://inis.iaea.org/collection/NCLCollectionStore/_Public/30/017/30017703.pdf

[5] Solving Renewables-Integrated Economic Load Dispatch Problem by Variant of Metaheuristic Bat Inspired

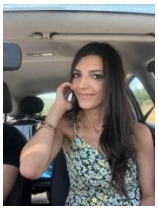
[6] GENETSKI ALGORITAM U PRIMJENI, SVEUČILIŠTE U ZAGREBU, FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

[7] Niknam, T.; Azizipanah-Abarghooee, R.; Aghaei, J. A new modified teaching-learning algorithm for reserve constrained dynamic economic dispatch. IEEE Trans. Power Syst. 2012, 28, 749–763.

[8] Niknam, T.; Azizipanah-Abarghooee, R.; Aghaei, J. A new modified teaching-learning algorithm for reserve constrained dynamic economic dispatch. IEEE Trans. Power Syst. 2012, 28, 749–763.

[9] Komparativna analiza troškova proizvodnje električne energije, University of Zagreb, Faculty of Economics and Business / Sveučilište u Zagrebu, Ekonomski fakultet

Kratka biografija:



Stanka Košutić je rođena 25.09.1997. godine u Gacku. Osnovnu školu završila je 2012. godine u Gacku. Gimnaziju “Pero Slijepčević” u Gacku završila je 2016. godine. Iste godine, upisala je u Novom Sadu Fakultet tehničkih nauka, odsjek Primjenjeno softversko inženjerstvo, koji je završila 2020, zatim upisala master studije na istom odsijeku. Položila je sve ispite predviđene planom i programom.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/16BE21Bibic

ANALIZA I PREDIKCIJA STOPE SAMOUBISTAVA UPOTREBOM TEHNIKA ISTRAŽIVANJA PODATAKA

SUICIDE RATE ANALYSIS AND PREDICTION USING DATA MINING TECHNIQUES

Boris Bibić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Priložen rad opisuje istraživanje o prepoznavanju najznačajnijih faktora rizika samoubistava i kreiranja modela za predikciju stopa samoubistava. Prikupljeni su skupovi podataka nad kojima je izvršen proces analize, obrade i spajanja podataka. Dobijene različite verzije skupova podataka su korišćene za treniranje više različitih verzija modela. Zbog velike dimenzionalnosti podataka vršeno je ispitivanje redukcije podataka na tačnost prediktivnih modela. Pronađeni su faktori rizika na globalnom i državnom nivou, a ispitan je i uticaj geografskih regija, religija i primanja na faktore rizika. Prediktivni modeli su evaluirani, a rezultati najznačajnijih faktora rizika su upoređeni sa rezultatima medicinskih istraživanja.*

Ključne reči: *Predikcija samoubistava, rudarenje podataka, predikcioni algoritmi, algoritmi za redukciju dimenzionalnosti, najznačajniji faktori rizika samoubistava*

Abstract – *This paper describes research on identifying the most significant risk factors for suicide and creating a model for predicting suicide rates. Data sets were collected over which the process of data analysis, processing and merging was performed. The obtained different versions of the data sets were used to train several different versions of the model. Due to the large dimensionality of the data, the reduction of data to the accuracy of predictive models was examined. Risk factors at the global and national levels were found, and the influence of geographical regions, religions and income on risk factors was examined. Predictive models were evaluated, and the results of the most significant risk factors were compared with the results of medical research.*

Keywords: *Suicide prediction, data mining, predictive algorithms, dimensionality reduction algorithms, the most significant suicide risk factors*

1. UVOD

Samoubistvo je ozbiljan javnozdravstveni problem koji disproporcionalno pogađa sve države sveta. Prema statistici Svetske zdravstvene organizacije (SZO) suicid je treći najčešći uzrok smrti kod dece uzrasta 15-19 godina, a drugi najčešći uzrok smrti kod osoba uzrasta 15-29 godina [1]. Prepoznavanje faktora rizika koji doprinose pojavi suicida je od krucijalne važnosti kako za zakonodavce tako i za stručnjake koji se bave prevencijom i prepoznavanjem samoubistva kod pojedinaca. Ovaj rad bi pomogao pri prepoznavanju faktora rizika u pojedinačnim državama i omogućio kreiranje procena kretanja stopa samoubistva.

Ideja istraživanja jeste prepoznavanje najvažnijih faktora rizika i kreiranje modela za predikciju samoubistva u svetu. Faktori rizika koji su razmatrani u radu su odabrani uz pomoć stručne literature i sličnih radova, ali neki su dodati od strane autora kao potencijalno interesantni. Prikupljeno je 49 tabela koje su analizirane, a izvučena statistika i informacije su pomogle pri daljoj obradi tabele i pronalaženju optimalnog opsega godina. Opseg od 1990. do 2017. godine je uzet kao optimalni za dalje razmatranje. Tabele sa premalo podataka u željenom opsegu su izbačene iz dalje obrade, pa je broj tabela spao na 29. Dalja obrada je obuhvatala popunjavanje nedostajućih vrednosti, gde su isprobani regresioni algoritmi *Elastic net* i *XGBoost*, kao i duboke neuronske mreže (engl. *DNN - Deep Neural Networks*). Za potrebe treniranja ovih modela, nedostajući podaci u skupovima podataka su popunjeni sa najmanjom i najvećom vrednosti iz skupa, sa nula vrednosti i sa prosečnom vrednosti iz skupa, čime su dobijene četiri varijacije skupova podataka. Nakon treniranja regresionih modela, originalne tabele su propušteni kroz *Elastic net* ili *XGBoost* model, u zavisnosti od toga koji je imao veću tačnost za posmatranu tabelu. Takođe, tabele su propuštene i kroz *DNN* model, pa su time dobijene dve varijacije početnih tabela bez nedostajućih vrednosti. Ove tabele su filtrirane da sadrže samo godine iz odabranog opsega (1990.-2017.).

Dobijene tabele su spojene po zajedničkim atributima, a to su godina i *ISO 3* kôd države, očuvajući varijacije tabela. Dobijeni su *Classic* i *Neural* verzija konačnog skupa podataka koje su se koristile za treniranje predikcionih modela za predviđanje stopa samoubistava, ali i za pronalaženje najznačajnijih faktora rizika. Broj atributa u obe verzije konačnog skupa je 33, a broj država je 142 za *Classic* i 123 za *Neural* verziju konačnog skupa podataka.

Usled velikog broja atributa u konačnom skupu podataka, rađena je redukcija dimenzionalnosti. Ispitana su 4 algoritma za redukciju dimenzionalnosti - *Low Variance Filter* (*LVF*), *Random Forest* (*RF*), *Principal Component Analysis* (*PCA*) i *Factor Analysis* (*FA*). Obe verzije konačnog skupa podataka su propuštene kroz ova četiri

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

algoritma čime je dobijeno osam novih (redukovanih) verzija konačnog skupa podataka. Ovih osam verzija, zajedno sa originalne dve verzije konačnog skupa podataka, su iskorišćene za treniranje tri prediktivna algoritma. *Elastic net, XGBoost* i *Partial Least Squares* (*PLS*) su odabrani za kreiranje prediktivnih modela.

Validacija prediktivnih modela je izvršena uz pomoć srednje kvadratne greške i koeficijenta determinacije, dok je za *PLS* dodatno rađena unakrsna validacija. *PLS* regresioni model treniran sa *PCA* redukovanim *Classic* skupom podataka se pokazao kao najbolji prediktivni model sa koeficijentom determinacije 93,6%. Unakrsna validacija je pokazala da je *PLS* algoritam davao najbolje rezultate kada je radio sa neredukovanim skupom podataka i algoritamski birao optimalan broj komponenti.

Faktori rizika za samoubistvo su određeni uz pomoć *Random Forest* algoritma. Posmatrana je *Classic* i *Neural* verzija konačnog skupa podataka da bi se dobili najznačajniji faktori rizika na globalnom nivou. Osim ovih faktora na globalnom nivou, posmatrani su i faktori rizika u odnosu na geografsku regiju, religiju, primanja kao i faktori rizika u pojedinim državama. *Random Forest* algoritam je označio, na svetskom nivou, *Fertility rate* kao najuticajniji faktor rizika za samoubistvo u obe verzije skupa podataka. U Istočnoj Aziji i Pacifiku, Južnoj Aziji i Severnoj Americi gustina naseljenosti i rast populacije su predstavljali najznačajnije faktore rizika po algoritmu, dok je za Bliski Istok i Severnu Afriku to bio broj umrlih na 1000 stanovnika. U većini slučajeva pri filtriranju konačnog skupa podataka po primanjima i religiji top deset liste su činile pojedinačne države i regioni, dok su se u slučaju grupisanja po primanjima našle i neke religije. Na kraju su ispitani faktori rizika u pojedinim državama koje su odabrane od strane autora ovog rada. U većini posmatranih država zastupljenost mentalnih poremećaja, kao i zloupotreba alkohola i psihoaktivnih supstanci su predstavljali najbitnije faktore rizika što se slaže sa naučnom literaturom.

2. METODOLOGIJA

Metodologija ovog rada uključuje sledeće oblasti: rad sa skupom podataka, treniranje modela za predikciju i pronalaženje najznačajnijih faktora rizika. U daljem tekstu biće detaljno opisan svaki korak koji je uključen u metodologiju ovog rada po redosledu njihove realizacije počevši od prikupljanja podataka.

2.1. Prikupljanje skupova podataka

Polazni skup podatka sadrži broj samoubistava po državama od 1990. do 2017. godine sa 6468 podataka na osnovu kojih je treniran model. Ostali skupovi podataka obuhvataju podatke iz oblasti državnog uređenja, ekonomije, prava, medija, kao i drugih relevantnih oblasti koje mogu predstavljati potencijalne faktore rizika samoubistva. Većina skupova podataka je odabrana na osnovu istraživanja stručne literature, dok je ostatak izabran kao potencijalno interesantan za istraživanje od strane autora. Tabele su prikupljene sa internet stranica institucija kao što su Ujedinjene Nacije zbog kredibiliteta podataka. Ukupno je prikupljeno je 49 tabela koje su analizirane, a izvučena statistika i informacije su pomogle pri daljoj obradi tabela i pronalaženju optimalnog opsega godina.

2.2. Analiza i obrada podataka

Prvo se pristupilo eksplorativnoj analizi prikupljenih skupova podataka. Cilj ovog postupka je bio da se proceni početno stanje tabela, da se iz statističkih podataka svih tabela izračuna optimalni opseg godina za posmatranje koji bi uključivao što je veći opseg godina, a pri tom sadržao što je više moguće podataka. Izabran je opseg od 1990.-2017. koji je sadržao većinu podataka iz odabranih tabela. Problemi koji su se trebali rešiti pre spajanja su popunjavanje nedostajućih podataka, izbacivanje dupliranih podataka i filtriranje tabela na izabran opseg godina. Nedostajući podaci su inicijalno popunjavani sa konstantama (nula, minimumom, maksimumom i srednjom vrednosti iz datog skupa) da bi se tabele mogle iskoristiti za treniranje regresionih algoritama. Odabrani su *Elastic net, XGBoost* i duboke neuronske mreže kao regresioni algoritmi, čija se tačnost nad sve četiri verzije skupova podataka evaluirao. Najbolja verzija iz grupe *Elastic net* i *XGBoost*, kao i *DNN* verzija je iskorišćena za popunjavanje nedostajućih podataka. Originalne tabele sa nedostajućim podacima su propuštane kroz dva regresiona modela i time su nedostajuće vrednosti iz svih skupova podataka bile popunjene. Pre spajanja skupova podataka, sve tabele su filtrirane tako da sadrže samo podatke iz željenog opsega godina.

2.3. Spajanje skupova podataka

Spajanje je vršeno sa *INNER JOIN* metodom, gde su tabele spajane na osnovu zajedničkih atributa - godina i *ISO 3* kôd države. One države koje se nisu našle u svim tabelama su bile izbačene iz skupa podataka. Kreirane su dve verzije konačnog skupa podataka, gde su tabele obrađene sa *Elastic net* ili *XGBoost* algoritmom činile *Classic* verziju, dok su tabele obrađene sa *DNN* činile *Neural* verziju. Razlog za upotrebu ove metode jeste da se osigura kompletnost skupa podataka, tačnije da sve države imaju sve podatke za svaku od godina u odabranom okviru godina, jer treniranje prediktivnih modela zahteva skup podataka bez nedostajućih vrednosti. Time je ukupan broj država spao sa 231 na 142 za *Classic* verziju skupa podataka i 123 za *Neural* verziju skupa podataka. Tabele koje nisu imale nedostajuće vrednosti na početku analize podataka su se spajale i sa tabelama iz *Classic* verzije i sa tabelama iz *Neural* verzije.

2.4. Obeležja spojenog skupa podataka

Nakon kreiranja dve verzije konačnog skupa, broj obeležja u obe verzije je bio 33, a broj država je bio 142 i 123 za *Classic* i *Neural* verziju. Neka od obeležja iz konačnog skupa podataka su: *ISO3* kôd svake od država u skupu, godina, procenat korupcije u državi, ocena poštovanja ljudskih prava, godišnja inflacija, procenat stanovništva koji ima problem sa zloupotrebom alkohola i psihoaktivnih supstanci, dominantna religija, broj dece koji žena rodi tokom svog života i procenat stanovništva koji umre od samopovređivanja. Za predikciju samoubistava bilo je potrebno kreirati modele koji predviđaju poslednje obeležje na osnovu ostalih.

2.5. Enkodovanje string obeležja

Algoritmi za kreiranje prediktivnih modela koji su korišćeni u radu zahtevaju da ulazni skup podataka ima isključivo numeričke vrednosti. U konačnom skupu podataka postoje četiri obeležja koje imaju string vrednost: *ISO 3* kôd, dominantna religija, pripadnost grupi u odnosu na ukupna primanja stanovništva i region. Ova obeležja je bilo neophodno pretvoriti u numeričke vrednosti kroz postupak enkodovanja podataka. Algoritam za enkodovanje podataka koji je korišćen u radu je bio *One-Hot Encoding*. Nakon propuštanja svih verzija skupova podataka kroz algoritam za enkodovanje, dobijeni su skupovi podataka koji su bili spremni za obučavanje predikcionih modela na globalnom nivou. Ovim skupovima se broj kolona povećao na 188 (*Classic* verzija) i 169 (*Neural* verzija) kao posledica enkodovanja string obeležja.

2.6. Redukcija dimenzionalnosti

Usled povećanja broja obeležja, razmatran je uticaj redukcije dimenzionalnosti na tačnost predikcionih modela. Da bi se ispitao ovaj uticaj, bilo je potrebno kreirati skupove podataka sa redukovanim brojem obeležja, a za to su upotrebljeni redukcioni algoritmi *Low Variance Filter* (*LVF*), *Random Forest*, *Principal Component Analysis* (*PCA*), *Factor Analysis* i *Partial Least Squares* (*PLS*). Propuštanjem obe verzije konačnog skupa podataka kroz četiri odabrana redukciona algoritma dobijeno je osam novih skupova podataka. Novodobijeni (redukovani) skupovi podataka zajedno sa dve (neredukovane) verzije konačnog skupa podataka iskorišćeni su za treniranje prediktivnih modela.

2.7. Treniranje prediktivnih modela

Prediktivni algoritmi u radu su korišćeni za dobijanje modela za predikciju samoubistava na globalnom nivou. Ciljno obeležje koje su regresioni modeli trebali da prediktuju jeste stopa samoubistva. Odabrana su tri regresiona algoritma (*Elastic net*, *XGBoost* i *Partial Least Squares*) koja su trenirana nad 10 skupova podataka (dve neredukovane verzije i osam redukovanih). Rezultat treniranja je trideset modela za predikciju stope samoubistava na globalnom nivou, zajedno sa rezultatima validacije ovih modela.

2.8. Optimizacija hiperparametara modela

Optimizacija hiperparametara, tj. parametara modela je rađena sa ciljem dobijanja što boljih rezultata i posvećeno je dosta pažnje empirijskoj optimizaciji istih. Ona je vršena uz pomoć trening skupa i rezultata unakrsne validacije. Parametri algoritama za redukciju dimenzionalnosti optimizovani su empirijski, ali je uzeta u obzir i domenska osnova. Domenska preporuka granične vrednosti kod *Low Variance Filter* algoritma je 80%, a u obzir su još uzeti 20%, 40% i 60%. Kod *PCA* i *Factor Analysis* algoritma, optimizacija broja željenih obeležja na koji se svodi dimenzionalnost je bila isključivo empirijska i uzeto je trećina, polovina i dve trećine ukupnog broja obeležja. Parametar *alpha* kod *Elastic net* algoritma je podešen na 0,01, ali isprobane su i vrednosti 0,001, kao i 0,1 i 0,8. Za optimizaciju *XGBoost* algoritma su podešavani *learning_rate* (0,1), *colsample_bytree* (0,3), *max_depth* (5) i *n_estimators* (10) parametri.

2.9. Validacija prediktivnih modela

Za validaciju svih modela su korištene dve metode - srednja kvadratna greška i koeficijent determinacije. Dodatna metoda validacije za *PLS* algoritam je bila unakrsna validacija. Validacija prediktivnih modela je zahtevala da se svi skupovi podataka pre početka treniranja podele na trening i test skup. Trening skup se koristio za obučavanje prediktivnih modela i sadrži 80% podataka iz skupa. Preostalih 20% čini test skup za validaciju dobijenih modela.

2.10. Pronalaženje najznačajnijih faktora rizika

Cilj pronalaženja najznačajnijih faktora rizika za samoubistvo je bila provera kvaliteta konačnog skupa podataka uz pomoć *Random Forest* algoritma, ali i utvrđivanje da li se kreiran skup podataka i dobijeni modeli za predikciju slažu sa naučno dokazanim činjenicama o faktorima rizika za samoubistvo. Prvo su posmatrani faktori rizika na globalnom nivou tako što su *Random Forest* algoritmu prosleđene *Classic* i *Neural* verzija konačnog skupa podataka. Zatim su posmatrani faktori rizika na nivou geografskih regija, religija, država sa istim prihodima stanovništva i na nivou pojedinih država. Za ove potrebe, obe verzije konačnog skupa podataka su filtrirane na osnovu željenih obeležja – *Region* za geografsku regije, *Dominant religion* za religije, *Income group* za prihode stanovništva i *Country code* za pojedinačne države, a zatim prosleđene *Random Forest* algoritmu. Ovaj algoritam je zatim vraćao listu svih obeležja sortiranu od najuticajnijeg ka najmanje uticajnom obeležju na rezultat predikcije koja je ograničena na deset najznačajnijih obeležja radi lakšeg prikazivanja na grafikonima.

3. REZULTATI

3.1. Rezultati prediktivnih modela

PLS regresioni model treniran sa *PCA* redukovanim *Classic* skupom podataka se pokazao kao najbolji prediktivni model sa koeficijentom determinacije 93,6%. Originalna *Neural* verzija skupa podataka, kao i ona redukovana sa *Factor Analysis* algoritmom je iskorišćena za treniranje *PLS* regresionog modela koji je dao najbolji rezultat od 92,9%. Najlošije rezultate su pokazali *XGBoost* prediktivni modeli. Pokazalo se da su neredukovane verzije u većini slučajeva imale bolje rezultate od svojih redukovanih verzija. Unakrsna validacija je pokazala da je *PLS* algoritam davao najbolje rezultate kada je radio sa neredukovanim skupom podataka i algoritamski birao optimalan broj komponenti (često je taj broj bio ispod 20). Razlike između *Classic* i *Neural* verzije konačnog skupa podataka su vrlo malo uticale na tačnost prediktivnih modela, gde je prosečna razlika bila ispod 4%.

3.2. Najznačajniji faktori rizika

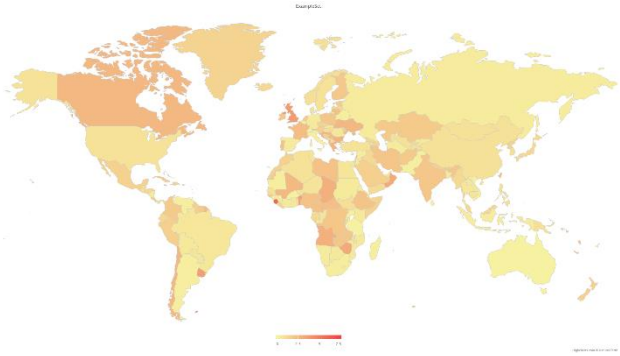
Na globalnom nivou, broj dece koji žena rodi tokom svog života je označen kao najznačajniji faktor rizika. Zdravstveni poremećaji kao što su depresija i bolesti zavisnosti su se našle visoko na globalnoj listi faktora rizika, kao i pojedine države koje imaju visoke stope samoubistava.

Većina faktora rizika koji su se pojavljivali u globalnim listama, nalaze se i u listama za regije. U regijama Istočna Azija i Pacifik, Južna Azija i Severna Amerika gustina naseljenosti i rast populacije su predstavljali najznačajnije faktore rizika.

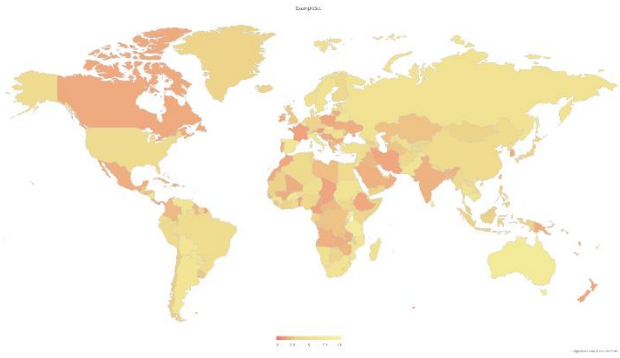
Bliski Istok i Severna Afrika je imala broj umrlih na 1000 stanovnika kao faktor broj jedan za određivanje stope samoubistva. Top deset liste su činile pojedinačne države i regioni kad je vršena filtracija konačnog skupa podataka po primanjima i religiji, a i neke religije su se našle u slučaju grupisanja po primanjima.

Kao na globalnom nivou, većina država su imale zastupljenost mentalnih poremećaja i zloupotrebu alkohola i psihoaktivnih supstanci za najbitnije faktore rizika. Zastupljenost mentalnih poremećaja, depresija, zloupotreba alkohola i psihoaktivnih supstanci su označeni kao najznačajniji faktori rizika u Bugarskoj, Mađarskoj, Litvaniji, Sloveniji, Japanu i Šri Lanki.

Ocena poštovanja ljudskih prava u Hrvatskoj, rast gradskog stanovništva u Crnoj Gori, *Fertility rate* u Severnoj Makedoniji, broj umrlih na 1000 stanovnika u Dominikanskoj Republici, postotak poslodavaca u Indiji i Južnoj Koreji (*Classic* verzije) prepoznati su od strane *Random Forest* algoritma kao najuticajniji na predikciju samoubistva. Korelacija između stope samoubistva (slika 1) i broja dece koji žena rodi tokom svog života (slika 2) može se vizuelno primetiti na mapi sveta.



Slika 1. *Prikaz stope samoubistva na mapi sveta*



Slika 2. *Prikaz obeležja Fertility rate na mapi sveta*

4. ZAKLJUČAK

Korišćenje većeg skupa podataka daje mogućnost dobijanja preciznije predikcije što je možda i ovde bio slučaj. *PCA* je pokazao najbolje rezultate od ispitanih redukcionih algoritama kada je predikcija vršena sa *PLS* algoritmom, dok je *Random Forest* pokazao najveću sposobnost redukcije dimenzionalnosti skupova podataka nezavisno od algoritma predikcije. U autorskom radu su dobijeni modeli čija je tačnost preko 90% kao što su *Factor Analysis* treniran sa *Neural* verzijom i *PCA* treniran sa *Classic* verzijom.

Značaj obeležja koji se dobija uz pomoć *Random Forest* algoritma ne pokazuje da li je korelacija pozitivna ili negativna, već samo ukazuje na jačinu iste. Dobijeni najznačajniji faktori rizika samoubistva na svetskom nivou se poklapaju sa faktorima koji su prepoznati u stručnoj literaturi [2]. Države koje danas imaju visoke stope samoubistava, kao što su Južna Koreja, Šri Lanka i Litvanija, nalaze se na top deset listi najznačajnijih faktora rizika [3].

U Indiji i Južnoj Koreji je broj poslodavaca bio bitan faktor što ukazuje na potrebu za finansijskom stabilnosti i nezavisnosti. Osnovna ljudska prava i uređenost države igraju važnu ulogu u prevenciji suicida što ukazuju rezultati Hrvatske, Albanije, Rumunije i Surinama. Prednost ovog rada u odnosu na većinu spomenutih radova jeste korišćenje skupa podataka na globalnom nivou, tako da je uticaj pojedinačnih država mali. Korišćeni skupovi podataka su javno dostupni i publikovani su od strane organizacija kao što su Ujedinjene Nacije.

5. LITERATURA

[1] World Health Organization. Suicide in the world: global health estimates. No. WHO/MSD/MER/19.3. World Health Organization, 2019.

[2] World Health Organization - Suicide: https://www.who.int/news-room/fact-sheets/detail/suicide (pristupljeno u septembru 2021.)

[3] World Health Organization - Global Health Observatory data repository: https://apps.who.int/gho/data/node.main.MHSUICIDEASDR?lang=en (pristupljeno u septembru 2021.)

Kratka biografija:

Boris Bibić rođen je u Subotici 1996. god. Master rad na Fakultetu tehničkih nauka iz oblasti Računarstva i automatike - Sistemi za istraživanje i analizu podataka odbranio je 2021. god. kontakt: borisbibic1996@gmail.com

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/16BE22Raseta

KATEGORIZACIJA NOVINSKIH ČLANAKA POMOĆU MAŠINSKOG UČENJA

NEWS CATEGORIZATION USING MACHINE LEARNING

Marko Rašeta, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu korišćeno je više modela za klasifikaciju novinskog članka na osnovu njegovog kratkog sažetka, koji se najčešće sastoji iz jedne ili dve rečenice, radi utvrđivanja kojoj kategoriji članak pripada (sport, politika, zabava…). Svakom od tih modela prosleđen je kratki sažetak koji je prethodno obrađen nekom od metoda za vektorsku reprezentaciju teksta. Od modela korišćeni su: logistička regresija, naivni Bajes, Support Vector Machine, neuronska mreža, konvolutivna neuronska mreža i rekurentna neuronska mreža. Za vektorsku reprezentaciju teksta korišćeni su tf-idf, Word2vec i GloVe. Modeli su trenirani na skupu podataka koji sadrži članke iz Huffington Post novina iz perioda 2012-2018. godine, a evaluacija je rađena na tim podacima, kao i na novinskim člancima koji su dobijeni scrape-ovanjem sa njihove veb stranice. Preciznost je računata kao odnos broja tačno pogođenih kategorija i ukupnog broja pogađanja, a prikazana je i F-mera.*

Ključne reči: *klasifikacija teksta, logistička regresija, naivni Bajes, Support Vector Machine, neuronska mreža*

Abstract – *In this paper many different models are used in order to classify news articles using their short description, mostly consisting of one or two sentences, in order to determine article's category (sports, politics, entertainment…). Each of those models is given short description which is first transformed into its vector representation using different methods. Models used are: logistic regression, naïve Bayes, Support Vector Machine, artificial neural network, convolutional neural network and recurrent neural network. For representing text in vector form tf-idf, Word2Vec and GloVe are used. Models were trained on dataset containing articles from Huffington Post from 2012-2018, and evaluation was done using those articles, as well as articles obtained by scraping Huffington Post's webpage. Models' accuracies and F-measures are given.*

Keywords: *text classification, logistic regression, naïve Bayes, Support Vector Machine, neural network*

1. Uvod

U poslednjih nekoliko decenija svedoci smo naglog razvoja tehnologije u mnogim sferama zivota. Neke su od starta atraktivne, dok su neke vremenom postajale sve interesantnije i zastupljenije. U skladu sa tim pojavljuju se neke nove profesije, dok neke u prošlosti ne toliko razvijene doživljavaju tehnološku transformaciju i ekspanziju sto dovodi do popularizacije istih. Neke od njih su novinarstvo, komunikologija, marketing i odnosi sa javnošću. To sve dovodi do pitanja kako unaprediti i olakšati takve poslove, jer ako potpuna automatizacija nije moguća bilo kakav njen vid je poželjan i neophodan. Klasifikacija teksta je prvi odgovor. Njene primene su posebno bitne za organizaciju digitalnih dokumenata i drugih digitalnih podataka kojih je sve više. Klasifikacija ili kategorizacija teksta predstavlja svrstavanje delova teksta, odnosno delova teksta u jednu ili više prethodno definisanih kategorija . Na primer, članci u novinama su uglavnom razvrstani po rubirkama kojima pripadaju, naučni radovi su klasifikovani po oblastima koje obrađuju, medicinski kartoni pacijenata su indeksirani po istorijatu bolesti, brojevima osiguranja, itd.

Svaki od navedenih primera može se predstaviti nekim skupom osobina. Na taj način se dodeljuju klase. Jedna od prvih primena klasifikacije odnosila se na određivanje autora datog teskta, dok danas najveću primenu imamo kada je reč o informativnim portalima, društvenim mrežama itd.

Konkretno kada je reč o novinarstvu, novinarima je uvođenjem e-portala posao već u mnogome olakšan. Međutim poboljšanjem modela novinari nece morati tekstovima dodeljivati i kategorije kako bi čitaoci mogli vršiti lakšu i bržu pretragu, i na efikasniji način pronalaziti njima interesantne informacije, već će ceo taj proces biti automatizovan.

2. PREGLED TRENUTNOG STANJA U OBLASTI

U radu [1] korišćen je samo Naive Bayes model za predviđanje kategorije novinskog članka. Ovaj rad objavljen je 2003. godine i kao takav spada među radove sa začetka istraživanja u ovoj oblasti. Skup podataka preuzet je sa indijskih novina Eenaadu iz perioda 2003-2004. godine. Sastoji se od 9870 članaka i 27.5 miliona reči, ali je u ovom radu odabrano 794 članka iz četiri kategorije (politika, sport, biznis i film) kako bi se predvidela jedna od ove 4 kategorije sa velikom preciznošću. Za pretprocesiranje podataka rađen je stemming, uklanjanje stop reči i identifikacija fraza i izraza. Za validaciju modela korišćeno je nasumično odabranih 20% članaka iz skupa podataka i na njima je dobijena preciznost od čak 94.72%.

U radu [2] korišćena su tri različita modela. Korišćeni modeli su: konvolutivna neuronska mreža, naivni Bajes i

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Aleksandar Kovačević.

SVM. Reči su se svodile na osnovni oblik (stem) i korišćena je metoda gde se od svake reči uzme samo njenih prvih 4-7 slova. Ta dva pristupa korišćena su samostalno, uz stem tag i uz word tag. Najbolje rezultate za KNN dala je metoda gde se uzimalo prvih pet slova reči i to 92.50%. Za naivnog Bajesa takođe najbolji rezultat je za prvih pet slova i on je ujedno i najprecizniji model sa preciznošću od 94.37%. SVM se nalazi u sredini po preciznosti i to kada se uzme prvih pet ili šest slova, jer je preciznost u ta dva slučaja ista – 93.12%.

U radu [3] korišćen je SVM model kako bi se klasifikovali indonežanski novinski članci. Delili su se samo na tri kategorije: biznis vesti, političke vesti i sportske vesti. Svrha ovog rada bila je da uporedi pristup bez selekcije feature i sa selekcijom. Za selekciju feature korišćen je Information Gain pristup i on je doveo do poboljšanja, jer je preciznost SVM modela bez selekcije iznosila 95.11%, dok je sa Information Gain preciznost čak 98.057%.

U radu [4] automatsko dodeljivanje kategorije novinskom članku vršeno je uz pomoć modela Support Vector Machine i Naive Bayes. Skup podataka sastoji se od 130000 novinskih članaka od kojih svaki pripada tačno jednoj od osam disjunktnih kategorija. Isprobane su dve vrste vektorizacije: Count Vectorization i TF-IDF Vectorization. Poređenjem rezultata ustanovljeno je da tf-idf daje bolje rezultate. Nakon vektorizacije uklonjene su stop reči što je dodatno poboljšalo rezultate. Korišćene su dve tehnike za izbor feateure-a koji se prosleđuju modelima, jer modeli rade bolje ukoliko prime manje, ali važnije feature-e. Prva korišćena tehnika je Chi-Squared koja je za SVM model dala čak 88.6% preciznost, a za Naive Bayes preciznost od 79.6%. Druga tehnika je LASSO i ona je za SVM model dala gotovo identičnu preciznost, 88.5%, dok je za Naivnog Bajesa preciznost bila 76.9%.

U radu [5] opisano je rešenje problema klasifikacije novinskih članaka primenom ant colony optimization algoritma. U obzir je uzeto pet mogućih kategorija. Skup podataka sastojao se iz 1000 novinskih članaka iz svake kategorije, odnosno 5000 članaka ukupno. Nad podacima je prvo rađen stemming, odnosno sve reči svođene su na svoj osnovni oblik. Zatim je rađena normalizacija podataka tako što su uklanjani svi znakovi interpunkcije i sva slova su prebačena u mala slova. Nakon toga izvršena je tokenizacija i uklonjene su sve takozvane stop reči, odnosno reči koje se često pojavljuju a ne nose gotovo nikakvo značenje. Model je evaluiran na tri skupa podataka: Berita10, Berita50 i Berita500 koji sadrže redom 10, 50 i 500 novinskih članaka od kojih svaki pripada jednoj od pet kategorija. Purity vrednosti za ova tri skupa bile su redom: 70%, 46% i 28.2%.

3. METODOLOGIJE

U ovom poglavlju biće opisane korišćene metode za pretprocesiranje podataka, kao i skup podataka koji je korišćen za obučavanje i testiranje modela.

3.1. Pretprocesiranje podataka

Svakom kratkom opisu novinskog članka prvo budu uklonjene stop reči i budu sva slova prebačena u mala slova. To je vršeno funkcijom pod nazivom clean_text. Ona je implementirana tako što se prvo sva slova prebacuju u mala slova. Zatim se karakteri /, (,), {, }, [,], |, @, ; i zapeta zamenjuju razmakom. Kada se to izvrši svi karakteri koji nisu malo slovo, cifra, #, + ili _ se uklanjaju. Ovaj postupak se primenjuje kako bi se iz teksta uklonili svi karakteri koji nemaju veliku težinu, odnosno ne nose nikakav ili nose veoma mali značaj. Nakon toga prolazi se reč po reč kroz tako obrađen tekst i izbacuju se sve reči koje pripadaju skupu stop reči za engleski jezik. Stop reči su reči koje ne nose nikakvu semantiku, pa se iz tog razloga mogu ukloniti. Npr. To su reči a, the, is, are. Nakon toga se kratak opis prebacuje u svoju vektorsku reprezentaciju. To je urađeno tako što su reči prosleđivane modelima koji su zatim vršili transformaciju vraćajući vektore. Kako kratki opisi vesti najčešće nisu duži od dve rečenice, do vektorske reprezentacije celog teksta dolazilo se tako što se uzimala srednja vrednost vektora reči tog opisa.

Za tf-idf[4] transformaciju u vektor korišćen je TfidfVectorizer biblioteke sklearn. Word2Vec[5] model koji je korišćen preuzet je preko gensim downloader-a i zove se word2vec-google-news-300 i on prevodi svaku reč u vektor dimenzije 300. Konačno, GloVe[6] model koji je korišćen takođe je preuzet uz pomoć gensim i nosi naziv glove-wiki-gigaword50 i prevodi svaku reč u vektor dimenzije 50.

3.2. Skup podataka

Za rešavanje problema korišćen je skup podataka koji se sastoji od 200000 novinskih članaka koji pripadaju nekoj od čak 41 različitoj kategoriji. Kako je to prevelik broj kategorija da bi se napravili precizni modeli i neke kategorije nisu zastupljene u velikom broju, taj skup podataka sveden je na članke koji pripadaju nekoj od sledećih šest kategorija: sport, kriminal, politika, zdravlje, zabava i putovanje. Time je broj članaka sveden na 85000. Primer jednog JSON objekta koji se nalazi u skupu podataka:

{

"category": "CRIME",

"headline": "There Were 2 Mass Shootings In Texas Last Week, But Only 1 On TV",

"authors": "Melissa Jeltsen",

"link":"https://www.huffingtonpost.com/entry/texas-amanda-painter-mass-shooting_us_5b081ab4e4b0802d69caad89",

"short_description": "She left her husband. He killed their children. Just another day in America.",

"date": "2018-05-26"

}

4. EKSPERIMENTALNI REZULTATI I DISKUSIJA

U ovom poglavlju će biti izneti i prodiskutovani rezultati u zavisnosti od toga koji model i koji način pretprocesiranja podataka su korišćeni.

4.1 Evaluacija

Evaluacija modela rađena je na dva načina. Jedan je tako što je iz celog skupa podataka od oko 85000 članaka

uzeto nasumično 30% podataka, što iznosi oko 25000 vesti, i na tome se vršilo testiranje, dok su se na preostalih 70% podataka modeli obučavali. Drugi način je tako što je sa veb stranice novina čiji su članci i korišćeni za obuku modela prikupljeno 50 članaka koji pripadaju jednoj od šest kategorija na kojima su modeli i obučavani. Ovi rezultati su manje merodavni iz razloga što ih nema mnogo, ali su ipak zanimljivi kako bi se videlo kako se modeli pokazuju na vestima koje su objavljene nekoliko godina nakon vesti na kojima su modeli obučavani, jer su modeli obučavani na podacima koji su prikupljani u periodu 2012-2018. godine. Za prikupljanje je korišćen selenium. Na veb stranici Huffington Post novina nalazi se stranica na kojoj se nalaze vesti sortirane od najnovijih ka starijim. Nakon što je prosleđena putanja do te stranice, od svake vesti skriptom su prikupljene kategorije i kratak opis na osnovu kog modeli predviđaju kategoriju tog članka.

4.2. Diskusija

Modeli neuronskih mreža pokazali su bolje rezultate od modela naivnog Bajesa, SVM modela i logističke regresije. Od ta tri modela logistička regresija je za nijansu uspešnija od SVM-a, dok je naivni Bajes pokazao najslabije rezultate. Od neuronskih mreža konvolutivna se pokazala za malo boljom od rekurentne, dok je obična neuronska mreža dala malo lošije rezultate od prethodne dve. Što se tiče pretprocesiranja podataka, zajedničko svim modelima je da se Word2Vec vektorska reprezentacija pokazala najboljom, GloVe reprezentacija za neke modele nije bila mnogo lošija, a tf-idf reprezentacija je dala najlošije rezultate u svakom modelu. Modeli su, sa obzirom na veći broj kategorija, pokazali solidne rezultate, ali ipak nedovoljno dobre za neku praktičnu upotrebu, npr. Potpuna automatizacija procesa dodeljivanja kategorija. Takvi rezultati bi mogli da se postignu ako bi se skup podataka proširio tako da sve kategorije budu podjednako zastupljene, jer su F-mere ipak bile najmanje za one kategorije kojih ima manje u skupu podataka. Drugi način bi bio da se modeli vektorske reprezentacije pojedinačnih reči, zamene modelima pretprocesiranja podataka koji bi ceo tekst prebacivali u vektor (npr. Doc2Vec[15]).

5. ZAKLJUČAK

U ovom radu korišćeno je šest različitih modela za kategorizaciju vesti. Svaka vest pripada jednoj od šest kategorija. Korišćeni su naivni Bajes, support vector mačine, logistička regresija, veštačka neuronska mreža, konvolutivna neuronska mreža i rekurentna neuronska mreža. Za vektorizaciju ulaznog teksta korišćeni su tf-idf, GloVe i Word2Vec. Kao modeli najbolje su se pokazali KNM i RNM, dok se od metoda za vektorizaciju Word2Vec pokazao kao najbolji, ali nije mnogo lošiji bio ni GloVe. Iz svega navedenog jasno je da je problem rešiv i da je moguće napraviti modele koji će sa velikom preciznošću kategorisati vesti. Veća preciznost mogla bi biti ostvarena ukoliko bi se smanjio broj kategorija. Takođe moguće je isprobati i modele koji u ovom radu nisu korišćeni u cilju poređenja sa njima.

Ostaje prostora i za dalja istraživanja, jer je velik izazov napraviti model koji bi kategorisao ne samo vesti, već i neke druge tekstove i radove i to tako što bi dodeljivao jednu od 10, pa kasnije čak i više kategorija. To bi bilo moguće postići proširivanjem skupa podataka ili korišćenjem modela za vektorsku reprezentaciju dokumenata.

Konkretna primena takvog modela mogla bi biti u novinama, kako ne bi urednik morao da dodeljuje ručno kategoriju svakoj vesti, ali za tako nešto potrebno je imati model sa preciznošću od bar 90% koji dodeljuje jednu od bar 10 kategorija. Pored toga, ovakvi modeli bi mogli doprineti i boljoj pretrazi na Google-u, jer ako postoji neki tekst ili rad bez određenih oznaka ili direktno definisane oblasti/teme, on bi opet mogao da bude rezultat pretrage vezane za neku temu ukoliko model dodeli tu temu tom tekstu.

6. LITERATURA

[1] Kavi Narayana Murthy (2003), “Automatic Categorization of Telugu News Articles”

[2] Burak Kerim Akkus, Ruket Cakici (2013), “Categorization of Turkish News Documents with Morphological Analysis”

[3] Adhy Rizaldy, Heru Agus Santoso (2017), “Performance improvement of Support Vector Machine (SVM) With information gain on categorization of Indonesian news documents”

[4] Juan Ramos (2003), “Using TF-IDF to Determine Word Relevance in Document Queries”

[5] KW Church (2017), “Word2Vec”

[6] Jeffrey Pennington, Richard Socher, Christopher D. Manning (2014), “GloVe: Global Vectors for Word Representation”

Kratka biografija

Marko Rašeta rođen je 15. aprila 1997. godine u Novom Sadu, Srbija. Fakultet tehničkih nauka upisao je 2016. na studijskom programu Računarstvo i Automatika. Diplomski rad iz oblasti XML i veb servisi odbranio je 2020. godine. Master rad na usmerenju Elektronsko poslovanje odbranio je 2021. godine.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/16BE23Radojcin

КОМПАРАТИВНА АНАЛИЗА ТЕХНОЛОГИЈА ЗА МАШИНСКО УЧЕЊЕ НА ИВИЦИ ПРИМЕНОМ NVIDIA JETSON TX2 УРЕЂАЈА

COMPARATIVE ANALYSIS OF MACHINE LEARNING AT THE EDGE TECHNOLOGIES USING THE NVIDIA JETSON TX2

Милош Радојчин, *Факултет техничких наука, Нови Сад*

Област – ПРИМЕЊЕНЕ РАЧУНАРСКЕ НАУКЕ И ИНФОРМАТИКА

Кратак садржај – *У овом раду дате су теоријске основе машинског учења на ивици и обраде тока података, значење термина Machine Learning Operations и опис Nvidia Jetson TX2 уређаја. Затим су анализиране технологије за машинско учење на ивици и дате су њихове компаративне анализе. Неке од ових технологија су примењене на решење демографске аналитике коришћењем Nvidia Jetson TX2 уређаја.*

Кључне речи: *Машинско учење, технологије, компаративна анализа, Nvidia Jetson TX2*

Abstract – *This paper presents the theoretical foundations of machine learning at the edge and data flow processing, the meaning of the term Machine Learning Operations and a description of the Nvidia Jetson TX2 device. Then, machine learning technologies at the edge are analyzed and their comparative analyzes are given. Some of these technologies were applied to the demographic analytics solution using Nvidia Jetson TX2 device.*

Keywords: *Machine learning, technologies, comparative analysis, Nvidia Jetson TX2*

1. УВОД

Крајњи уређаји, попут телефона и IoT сензора, генеришу податке који морају бити обрађени у реалном времену. Нови тренд уводи примену машинског учења за одбраду генерисаних података. Међутим, примена машинског учења и обрада података захтевају доста процесне моћи и брз одзив како би се извршавање обављало у реалном времену.

Како би се овај захтев испунио, за примену машинског учења на ивици (енгл. *machine learning at the edge*) неопходно је изабрати добар скуп технологија. Ово је важно јер у мору информација и података које се креирају великом брзином, у такозваним системима великих скупова података (енгл. *Big Data Systems*), потребно је имати добар алат са којим ће се креирати квалитетна решења и нови производи у софтверској индустрији.

У овом раду биће укратко дате теоријске основе машинског учења на ивици, затим биће укратко описан Nvidia Jetson TX2 уређај, увод у то шта је обрада тока података (енгл. *data stream*, *dataflow*) и значење термина Machine Learning Operations. Након тога биће наведене испробане технологије за машинско учење на ивици применом Nvidia Jetson TX2 уређаја за демографску аналитику и њихова компаративна анализа. Затим ће бити наведене изабране технологије и резултати постигнути њиховим коришћењем. На крају биће дат закључак.

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

2. ТЕОРИЈСКЕ ОСНОВЕ

У овом поглављу дате су теоријске основе области из ког се имплементација система за машинско учење на ивици применом Nvidia Jetson TX2 уређаја за демографску аналитику састоји.

2.1. Машинско учење и његова примена на ивици

Машинско учење је еволуирајућа грана рачунарских алгоритама који су дизајнирани да опонашају људску интелигенцију учећи из окружења. Другим речима, машинско учење је грана вештачке интелигенције (енгл. *Artificial Intelligence*, AI) и рачунарске науке која је фокусирана на податке и алгоритме тако да имитира људе, постепено унапређујући сами себе. Машинско учење на ивици је област са одређеним бројем изазова али и могућности. Предности овог приступа су смањена количина података који се шаљу на централно место за њихово чување и тиме смањено време одзива, децентрализован приступ постаје робуснији на отказ у мрежи и приватност података је на већем нивоу. Постоји неколико метода машинског учења, и оне се деле на надгледано (енгл. *supervised*), ненадгледано (енгл. *unsupervised*) и учење условљавањем (енгл. *reinforcement learning*), у зависности од типа проблема који алгоритам решава.

2.2. Обрада тока података

Многи извори производе податке континуално. Примери укључују мреже, акције корисника, научне податке, мултимедије, трансакције и многе друге. Ови извори се зову извори података. Обрада токова података се у највишем употребљава за детекцију круцијалних ствари, као што су детекција превара, крађа, отказ надгледаног система, али и ради добијања статистичких података у жељене сврхе.

2.3. Machine Learning Operations

Термин Machine Learning Operations (MLOps) је дефинисан као проширење Development Operations (DevOps) методологије укључивањем средстава

машинског учења и науке о подацима као *first-class citizens* у оквиру DevOps екологије.

Принципи MLOps-а су оптимизација и аутоматизација процеса стављања нових ствари у продукцију и добијања повратне информације, континуални развој, испоручивање, тренирање модела и мониторинг података и метрика перформанси модела.

2.4. Nvidia Jetson TX2 уређај

Nvidia Jetson је серија напредних система коришћених од стране инжењера за креирање иновативних AI производа. Са Jetson платформом, која је данас једна од водећих у свету рачунарства на ивици, отворен је широк спектар за добијање искуства и креирање креативних пројектних решења. Платформа је састављена од малих јединица које се називају модули. Хардверска компонента која је срце ове платформе је Nvidia Pascal графичка картица, која је намењена за извршавање алгоритама машинског учења и неуронских мрежа, док софтверска компонента која представља подршку за изградњу и извршавање AI апликација је Nvidia JetPack SDK, што представља скуп пакета и библиотека, а међу најважније спадају OpenCV, TensorRT, cuDNN, CUDA и Nvidia Container Runtime.

3. ТЕХНОЛОГИЈЕ И ЊИХОВА КОМПАРАТИВНА АНАЛИЗА

У овом поглављу су наведене испробане технологије у имплементацији система за демографску аналитику и њихова компаративна анализа.

3.1. OpenCV

OpenCV (Open Source Computer Vision Library) је библиотека отвореног кода (енгл. *open source library*) за анализу и управљање сликама и видео садржајем представљена од стране Intel корпорације [1]. Саграђена је у намери да обезбеди механизам за обраду слике у апликацијама за машинско учење и убрза коришћене перцепције машина у комерцијалним производима [2]. Елементи обраде, као што су слика и видео, су представљени матрично што представља природну репрезентацију ових елемената и тиме чини употребу ове библиотеке једноставном.

3.2. YoloV5

YOLO (You Only Look Once) алгоритам, који је представљен 2015. године, донео је нови приступ преобликовањем детекције објеката као регресиони модел извршавајући се у једној неуронској мрежи. Основна идеја аутора YOLO алгоритма била је да ова архитектура израчуна све особине слике и направи предикције свих објеката истовремено, у једном пролазу. Због начина на који ради и резултата које постиже оцењен је као најбољи алгоритам за детекцију објеката [3].

3.3. Multi-Task Cascaded Convolutional Networks и Haar Cascade

MTCNN алгоритам је спорији због свог поступка који је у свакој фази захтеван. Међутим, даје веома добре резултате у погледу квалитета без обзира на квалитет улазне слике, у погледу угла лица, осветљености и заклоњености.

Haar Cascade алгоритам је брз алгоритам, али једна од мана овог алгоритма је што је у стању да детектује само лица спреда под добрим осветљењем, с обзиром на то да је на таквом скупу података трениран. Такође, често даје лажно позитивне резултате.

3.4. EfficientNetB7, VGG16 и InceptionV3

С обзиром на то да је EfficientNetB7 неуронска мрежа креирана коришћењем претраге неуронске архитектуре за проналажење добре основне мреже а затим и претрагом мреже величина за добијање оптималних вредности величина за скалирање по свим димензијама, она даје најбоље резултате. Разлог услед ког VGG16 неуронска мрежа даје боље резултате од InceptionV3 неуронске мреже јесте тај што је комплекснија у погледу броја рачунања што јој, без обзира што је потребно више времена за рачунање, помаже у остваривању бољих резултата. Табела 1 приказује постигнуте резултате сваке од мрежа понаособ на UTKFace скупу података.

Табела 1. *Постигнути резултати мрежа*

	accuracy	**loss**	**male f1-score**	**female f1-score**
EfficientNetB7	0.855	0.312	0.856	0.855
VGG16	0.833	0.384	0.852	0.809
InceptionV3	0.757	0.521	0.763	0.752

3.5. Apache Kafka и RabbitMQ

Apache Kafka и RabbitMQ системи могу се поредити у два погледа, у погледу квалитета и погледу квантитета.

Квалитативно поређење обухвата неколико особина: *time decoupling*, *routing logic*, *delivery guarantees*, *ordering guarantees*, *availability*, *transactions*, *dynamic scaling*.

Квантитативно поређење обухвата неколико особина: *latency*, *throughput*.

Поређењем система по овим особинама показало се да не постоји генерално бољи систем, већ је један систем у неким особинама бољи од другог па је за такве потребе и намењен.

Предности Kafka система: дуготрајно чување, репликација порука, Kafka Connect и компакција лог порука.

Предности RabbitMQ система: стандардизован протокол, подршка за коришћење више протокола истовремено, дистрибуирани модови топологије, свеобухватни алати за управљање и праћење, изолација окружења, праћење потрошача, мање коришћење диска, контрола произвођача, могућност ограничења величине редова и могућност ограничења животног века порука [4].

3.6. Apache Flink, Apache Kafka Streams и Apache Spark

Један од изазова у развоју архитектуре за обраду токова јесте избор праве технологије за различите

случајеве коришћења. Како свака од испробаних технологија функционише донекле по сличним принципима и архитектурама, корисно је издвојити особине сваке од технологија.

3.6.1. Програмски модел

Stream алат је базиран на микро-пакетном (енгл. *micro-batch*) принципу, што је проширење главног Spark интерфејса. Главна апстракција у Sparku је RDD (Resilient Distributed Dataset).

Flink је *native* алат за обраду података са подршком за обраду података у пакетима. Основни градивни блок Flink програма су токови и оператори трансформације пресликане у токове података.

Kafka Streams је део целокупног Kafka екосистема који се састоји од непромењивих записа, токова података звани *topic*-и, потрошача, произвођача, логова, партиција и кластера. Најважнија апстракција у Kafka Streams је ток који представља неограничен скуп непромењивих записа. Топологија представља граф процесора који су повезани токовима и који деле складишта стања, као и апстракцију кориштену да дефинише логику рачунања у токовима за апликације.

3.6.2. Партиционисање података

Spark аутоматски партиционише RDD компоненте и дистрибуира партиције преко различитих чворова. Hash и Range су честе стратегије за партиционисање подржане од Sparka.

Током извршавања, ток у Flinku садржи једну или више партиција и сваки оператор има један или више подзадатака који су му додељени. Ток може да трансформише податке између два оператора у "један-на-један" шаблону (енгл. *one-to-one pattern*), што очувава редослед елемената.

Kafka Streams партиционише податке ради процесирања док слој порука у Kafka екосистему партиционише податке за чување и транспорт порука. У оба случаја партиционисање омогућава локалитет, еластичност, скалабилност, високе перформансе и отпорност на грешке.

3.6.3. Управљање стањем

На високом нивоу апстракције, Sparkova Structured Streaming библиотека прати стање на сличан начин и у микро-пакетном и у серијском режиму. Стање апликације се прати коришћењем два спољашња система за складиштење, "дневником унапред" (енгл. *write-ahead log*) и складиштем стања.

Flinkov екосистем сачињен од модула и сервиса саграђен на његовом језгру нуди различите начине приступа и изолазије стања. Свака операција у току може дефинисати своје сопствено стање и ажурирати га континуално у намери да одржава резиме до сада виђених података.

Kafka Streams пружа апликацијама моћну, еластичну и високо скалабилну обраду са могућностима отпорности на грешке. Kafka Streams пружа складишта стања која могу бити употребљена од стране апликација за обраду података за складиште и добављају податке, што је важна способност за спровођење операција.

3.6.4. Гаранција процесирања

Sparkova Structured Streaming библиотека обезбеђује брзу, скалабилну обраду отпорну на грешке, *end-to-end exactly-once* обраду података помоћу микро-пакетног *engine*-а. Ове гаранције могу бити постигнуте избором релевантних режима базираних на захтевима апликација без промене Dataset или DataFrame операција.

Механизам за обраду порука у Flinku гарантује да ће и у случају пада, стање програма одржавати сваки запис из тока, тачно једном. Flink може да гарантује да се "тачно једном" стање ажурира у стање које дефинише корисник само када извор учествује у механизму контролних тачака (енгл. *checkpoints*).

Kafka Streams такође подржава *end-to-end exactly-once* семантику која гарантује да за било који запис прочитан из извора Kafka *topic*-а, резултат његове обраде ће бити рефлектован тачно једном у излазни *topic*.

3.6.5. Отпорност на грешке

Structured Streaming библиотека обезбеђује *end-to-end exactly-once* гаранцију отпорности на грешке кроз контролне тачке и *write-ahead logs* приступе. Structured Streaming систем контролних тачака користи већа складишта стања да чува тренутна стања оператора за дуготрајне операције.

Flink имплементира отпорност на грешке коришћењем комбинације репродукције тока и контролних тачака. Стриминг тока података у Flinku може да се надовеже са контролне тачке док се одржава доследност или семантика обраде "тачно једном".

Kafka Streams се надовезује на способност отпорности на грешке интегрисану у језгро Kafke. Kafkine партиције су веома доступне и репликабилне, као када је податак перзистентан и Kafki, доступан је чак и ако апликација падне и потребно га је поново обрадити [5].

3.7. MLflow

MLflow библиотека је дизајнирана да олакша развој машинског учења. То је платформа отвореног кода креирана да ради са било којом библиотеком машинског учења и било којим програмским језиком. MLflow дефинише компоненте које су дизајниране за адресирање фундаменталних изазова у свакој фази циклуса машинског учења, од развоја модела до стављања у продукцију, и то MLflow Tracking за праћење експеримената, MLflow Models за паковање модела, MLflow Projects за паковање кода и MLflow Registry за чување модела.

3.8. Apache Superset

Apache Superset је модерна веб-апликација пословне интелигенције. То је брза, лака, интуитивна апликација која садржи мноштво опција и која корисницима свих нивоа олакшава анализу и визуализацију података, од једноставних до веома детаљних графикона.

Superset пружа интуитиван интерфејс, широк спектар лепих визуализација, креирање визуализације без

кода, моћан SQL алат за припрему података и многе друге ствари.

3.9. MinIO

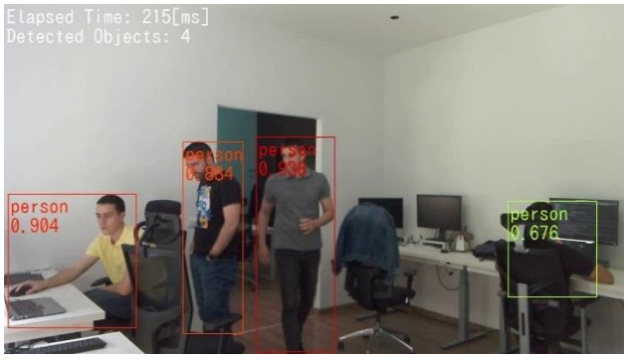
MinIO је систем високих перформанси за складиштење дистрибуираних објеката. MinIO се разликује по томе што је од свог почетка дизајниран да буде стандард у приватном и хибридном складишту објеката у облаку. Будући да је MinIO наменски направљен да служи само за складиштење објеката, једнослојна архитектура постиже сву потребну функционалност без компромиса. Резултат тога је *cloud-native* сервер објеката који је истовремено ефикасан, скалабилан и лаган. Иако се MinIO истиче у случајевима коришћења традиционалног складиштења објеката као што су секундарно складиштење, опоравак од катастрофе и архивирање, јединствен је у превазилажењу изазова повезаних за машинско учење, аналитику и радна оптерећења апликација у облаку.

4. РЕЗУЛТАТИ

За имплементацију система за демографску аналитику употребом Nvidia Jetson TX2 уређаја употребљен је следећи скуп технологија: OpenCV, YoloV5, Multi-Task Cascaded Convolutional Networks, EfficientNetB7, Apache Kafka, Apache Flink, MLflow, Apache Superset и MinIO.

Делови који обухватају учитавање слика са камере, закључивање (енгл. *inference*) неуронских мрежа и креирање материјала за каснију употребу обављају се на самом Jetson уређају, док остали део архитектуре се обавља на екстерној машини.

Слика 1 приказује пример резултата детекције особа на слици.



Слика 1. *Пример резултата детекције особа на слици*

5. ЗАКЉУЧАК

У овом раду су укратко дате теоријске основе машинског учења на ивици, затим је укратко описан Nvidia Jetson TX2 уређај, увод у то шта је обрада тока података и значење термина Machine Learning Operations. Након тога су наведене испробане технологије за машинско учење на ивици применом Nvidia Jetson TX2 уређаја за демографску аналитику и њихова компаративна анализа. Затим су наведене изабране технологије и резултати постигнути њиховим коришћењем.

Оно што ову архитектуру издваја од других често креираних архитектура јесте ниво машинског учења који је постигнут извршавањем машинског учења на Jetson уређају.

Примена ове имплементације на неки од случајева коришћења са циљем у реалној употреби представља тему за даљи развој. Унапређење модела машинског учења и имплементација интеракције више Jetson уређаја представљају тему за даља унапређења.

6. ЛИТЕРАТУРА

[1] Culjak, I., Abram, D., Pribanic, T., Dzapo, H., & Cifrek, M. (2012, May). A brief introduction to OpenCV. In *2012 proceedings of the 35th international convention MIPRO* (pp. 1725-1730). IEEE.

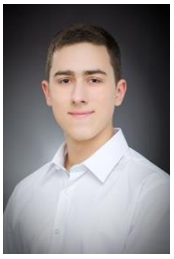
[2] Getting started with OpenCV and Python, https://medium.com/the-andela-way/simple-operations-on-images-using-opencv-d37b26e6e3ab

[3] Thuan, D. (2021). Evolution of yolo algorithm and yolov5: the state-of-the-art object detection algorithm.

[4] Dobbelaere, P., & Esmaili, K. S. (2017, June). Kafka versus RabbitMQ: A comparative study of two industry reference publish/subscribe implementations: Industry Paper. In *Proceedings of the 11th ACM international conference on distributed and event-based systems* (pp. 227-238).

[5] Isah, H., Abughofa, T., Mahfuz, S., Ajerla, D., Zulkernine, F., & Khan, S. (2019). A survey of distributed data stream processing frameworks. *IEEE Access*, *7*, 154300-154316.

Кратка биографија:



Милош Радојчин рођен је у Новом Саду 1997. год. Мастер рад на Факултету техничких наука из области Електротехнике и рачунарства – Примењене рачунарске науке и информатика одбранио је 2021. год.

контакт: milos.radojcin@uns.ac.rs

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.42:004.738.12
DOI: https://doi.org/10.24867/16BE24Jeftic

UPOTREBA DINAMIČKOG DNS-A ZA LAK PRISTUP KUĆNOJ MREŽI SA BILO KOG UDALJENOG MESTA

EASILY ACCESS HOME NETWORK FROM ANYWHERE WITH DYNAMIC DNS

Vanja Jeftić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – PRIMENJENE RAČUNARSKE NAUKE

Kratak sadržaj – *U ovom radu opisana je konfiguracija i upotreba dinamičkog DNS-a za lak pristup kućnoj mreži sa bilo kog udaljenog mesta. Govoriće se o DNS-u, o njegovim bitnim elementima, dinamičkom DNS-u i o konfiguraciji istog. Dat je i primer kreiranja testnog servera i podešavanja DDNS-a u okviru tog servera.*

Ključne reči: *DNS, IP, DDNS*

Abstract – *This paper describes the configuration and use of dynamic DNS for easy access to the home network from any remote location. There will be talk about DNS, its essential elements, dynamic DNS, and its configuration. An example of creating a test server and setting up DDNS within that server is also given.*

Keywords: *DNS, IP, DDNS*

1. UVOD

Veliki broj korisnika ima sadržaje u okviru svoje kućne mreže, kojima bi voleo da pristupi sa nekog udaljenog mesta. Rešenje za pomenuti problem nudi dinamički DNS. Za objašnjenje rešenja ovog problema potrebno je prvo razumeti kako funkcioniše DNS (eng. Domain Name System).

Internet predstavlja ogromnu mrežu uređaja, svaki od tih uređaja ima svoju oznaku, kako bi drugi uređaji mogli da ga identifikuju. Ta oznaka se zove IP adresa. O IP adresi će se detaljnije govoriti dalje u radu, za sada je dovoljno razumevanje da IP adresa predstavlja niz grupisanih brojeva, razdvojenih tačkom, dužine 32 bita (primer 192.168.1.1). Teško je zamisliti situaciju u kojoj bi svaki korisnik koji bi želeo da ostvari komunikaciju sa drugim uređajem ili situaciju kada bi želeo da poseti neki sajt, morao da pamti IP adresu tog uređaja ili IP adresu web sajta.

Korisnik u internet pretraživač unosi ime poput „example.com“ i dobija rezultat koji odgovara tom imenu. To ime se zove domensko ime (eng. Domain name). Veza između pomenutih IP adresa i domenskog imena je DNS. DNS možemo zamisliti kao veliki telefonski imenik, u kome se nalaze pomenute IP adrese i njima odgovarajuća domenska imena. O ostalim pojmovima DNS-a će se detaljnije govoriti u okviru druge oblasti. Treća oblast je posvećena dinamičkom DNS-u ili skraćeno DDNS-u.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio Docent Željko Vuković.

Kratko objašnjenje pojma bi bilo, da je to usluga koja omogućava korisniku da pristupi svom kućnom računaru sa bilo kog mesta u svetu. U okviru treće oblasti se može naći i primer direktne primene konfiguracije DDNS-a na testnom serveru.

2. DNS – (DOMAIN NAME SYSTEM)

2.1. Uopšteno o DNS-u

Gotovo svi korisnici koji koriste internet, koriste i DNS servis, a da toga nisu ni svesni. DNS je jedan od osnovnih servisa na internetu i predstavlja hijerarhijski uređen sistem, u kome se nalaze razne informacije, ali u najvećoj meri su zastupljeni podaci o IP adresama i domenskim imenima, kao i slovnim nazivima računara (eng. Hostname). Hostname predstavlja jedinstveno ime unutar neke mreže, koje koriste razni protokoli za elektronsku identifikaciju nekog računara. Klijentima DNS informacije pružaju DNS serveri, upotrebom DNS protokola za komunikaciju sa klijentima. Svrha i razlog kreiranja DNS-a je da se pojednostavi komunikacija između više računara i komunikacija između korisnika i računara odnosno interneta, jer više nije potrebno pamtiti nizove brojeva koji se nazivaju IP adrese, nego je dovoljno zapamtiti slovne nazive koji odgovaraju tim IP adresama.

2.2. IP adrese

Internet protokol ili skraćeno IP, je protokol koji sadrži podatke o adresiranju, čime se postiže da svaki mrežni uređaj koji je povezan na internet ima jedinstvenu adresu i može se lako identifikovati u celoj internet mreži, a takođe sadrži i kontrolne informacije koje omogućuju paketima da budu prosleđeni na osnovu poznatih IP adresa. Postoje dve verzije internet protokola koje su u upotrebi, verzija 4 (IPv4) i verzija 6 (Ipv6). Tokom dalje obrade teme, govoriće se o verziji 4, jer je ona i dalje više rasprostanjena [1].

IP adresa se sastoji iz dva dela, prvi deo predstavlja adresu IP mreže, koja je ista za sve računare na jednoj IP mreži i deo koji predstavlja adresu računara koji je jedinstven za svaki računar na toj istoj IP mreži. Prvi deo adrese služi za identifikovanje mreže, a drugi deo za identifikovanje računara koji je povezan na računarsku mrežu. IP adrese mogu biti javne i privatne. Javne se mogu koristiti na internetu za razliku od privatnih koje su namenjene mrežama i nisu direktno povezane na internet, stoga se ne mogu koristiti na internetu.

2.3. Domensko ime

Domensko ime je simboličko ime računara, koje ga jedinstveno identifikuje na internetu. DNS preslikava domensko ime u jednu ili više IP adresa i obrnuto. Dat je primer kako izgleda FQDN (енг. Fully Qualified Domain Name), koji predstavlja više labela - niz alfanumeričkih znakova sa maksimalno 63 znaka unutak svake labele, koje zajedno formiraju domesko ime.

FQDN: www.ftn.uns.ac.rs.

лабеле: www, ftn, uns, ac, rs

име рачунара: www

доменско име: ftn.uns

top-level домен: ac.rs

2.4. Domen

Domenska imena su uglavnom grupisana. Kriterijum grupisanja zavisi od labele kojom se ta domenska imena završavaju. Takve završne labele se nazivaju TLD (eng. Top-Level Domain) imena i postove dva tipa. Prvi tip su geografski bazirani domeni ili takozvani ccTLD (eng. Country code TLD). Drugi tip su generički domeni ili gTLD (eng. Generic TLD) domeni koji se uglavnom sastoje od tri ili više znakova. U jednom domenskom prostoru na mogu da postoje dve iste labele. Primer TLD-a:

gTLD: .com, .net, .org, .biz, .info, .museum, .travel, .xxx

ccTLD: .hr, .us, .eu, .fr, .es, .de, .it, .jp, ...

2.5. Registar imena domena – domenski registri

Domenski registri su baze podataka koje sadrže podatke o domenima i njima odgovarajućim IP adresama. Svaki TLD-Top-Level Domen ima svoj domenski registar.

2.6. DNS rezolucija

Rezolucija predstavlja proces primanja zahteva i obrade istog, te vraćanja odgovora na osnovu primljenog zahteva. Osnovna rezolucija je proces pretvaranja domenskog imena u IP adresu. Postoje dva osnovna tipa rezolucije, odnosno dva osnovna tipa prolaska kroz DNS hijerarhiju kako bi se otkrio tačan DNS zapis. Razlikuju se po tome ko obavlja većinski deo posla oko prikupljanja podataka i njihove obrade.

Iterativni tip rezolucije: situacija kada klijent šalje upite, a server odgovora sa jednim od dva moguća odgovora, ili odgovorom na zahtev ili imenom drugog DNS servera koji ima više podataka o traženom upitu.

Rekurzivni tip rezolucije: situacija kada klijent pošalje upit koji je rekurzivnog tipa, server preuzima posao pronalaženja informacija o traženom upitu. U ovom tipu, klijent šalje samo jedan zahtev i dobija ili tražene podatke ili poruku o grešci.

Rekurzivni tip pretraživanja je mnogo povoljniji za klijente od iterativnog, ali on može da optereti DNS servere u velikoj meri, stoga se takve forme upita dopuštaju samo računarima iz lokalne mreže.

2.7. DNS keš

Iterativni i rekurzivni prolazak kroz DNS stablo u cilju pretraživanja su bili veoma neefikasni [2], jer se za svaki upit moralo prolaziti kroz novo stablo. Sa ovakvim načinom rada, svaki upit bi predugo trajao. Rešenje za dati problem predstavlja DNS keš memorija. Način funkcionisanja se zasniva na činjenici da ako se nekom resursu nedavno pristupalo veće su šanse da će mu se ponovo pristupati. Stoga svi DNS serveri imaju internet keš memorije o nedavnim DNS upitima koji im omogućavaju da dobiju odgovor ili deo odgovora upravo iz te keš memorije. Time se smanjuje saobraćaj zahteva kao i vreme potrebno za pribavljanje nekog resursa.

2.8. DNS obrnuta rezolucija

Standardna rezolucija pretvara DNS imena u IP adrese. U standardnoj komunikaciji preko interneta, potrebno je obezbediti proces rezolucije u oba smera. Primer obrnute rezolucije bi bila provera da li određena adresa spada u neki domen. Problem kod ove rezolucije je da se serveri razlikuju prvenstveno po labelama odnosno po domenima za koje su nadležni, a potrebno je vratiti IP adresu kao povratnu informaciju. Da bi se omogućio ovaj proces, formirana je dodatna hijerarhije u vidu INADDR.ARPA domena. Radi se o domenskom prostoru koji se sastoji od četiri nivoa poddomena, a svaki nivo odgovara jednom delu IP adrese. Obrnutom DNS rezolucijom, odnosno prolaskom kroz četiri nivoa poddomena, dolazi se do čvora za traženu IP adresu koji pokazuje na odgovarajuće domensko ime.

2.9. DNS zona

DNS zona može da sadrži zapise za jedan domen ili za više domena. Zapis je tekstualni fajl koji opisuje DNS zonu. DNS dozvoljava da DNS adresni prostor bude podeljen u zone. Za svako domensko ime koje je uključeno u zonu, ta zona postaje autoritativni izvor za informacije koje se odnose na taj domen.

2.10. Način rada DNS-a

Kada korisnik unese određenu IP adresu u internet pretraživač, pretraživač u tom trenutku šalje zahtev putem interneta sa ciljem da dobije odgovor koji odgovara unetoj IP adresi. Taj zahtev dolazi do prve tačke obrade, koju obavlja DNS Resolver. On predstavlja posrednika između klijenta i DNS servera. DNS resolver zna u kom dalje smeru da uputi zahtev, odnosno na koji dalje DNS server da ga usmeri.

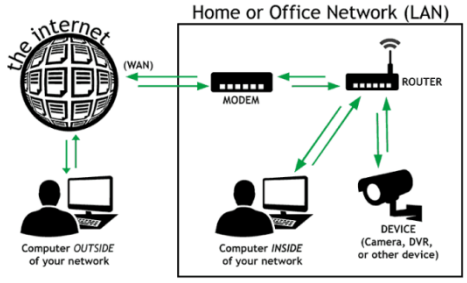
3. DINAMIČKI DNS

3.1. Uvod

Tema ovog rada je rešenje problema pristupa kućnom uređaju, računaru na primer, sa bilo koje udaljene lokacije. Da bi ovako nešto bilo moguće potrebno je da se korisnik opredeli za provajder koji omogućava uspostavljanje dinamičkog DNS-a, kao i da zna kako da konfiguriše svoj ruter.

DDNS (eng. Dynamic Domain Name System) je veoma sličan DNS-u, ali postoji razlika u tome što on pronalazi dinamičku IP adresu i upućuje na domen i obrnuto. Korisnici dobijaju IP adrese od njihovih internet provajdera i one se menjaju na određeni vremenski period. Da bi se rešio problem promenljive IP adrese,

uvodi se DDNS, preko koga se dodeljuje stalno ime IP adresi korisnikove kućne mreže sa automatskim ažuriranjem kako se adresa menja tokom vremena. Kada je DDNS podešen, korisniku je omogućeno da pristupi kućnoj mreži jednostavnim unosom dodeljenog statičkog imena. Na Slika 1. je prikazano kako izgleda pristup kućnoj mreži sa bilo koje lokacije.



Slika 1. *Pristup kućnoj mreži sa bilo koje lokacije [3]*

3.2. DNS Host

Konfigurisanje DNS-a je jednostavno i besplatno za korisnike, i jednom kada se DNS podesi, nije potrebno održavanje tokom vremena. Da bi se konfigurisao DNS, potreban je DNS Host. Na tržištu postoji dosta dobrih provajdera koji nude besplatan DNS hosting, kao što su No-IP, Dynu Systems и Zonomi DNS.

3.3. Ruter sa DDNS podrškom

Veoma je bitno obezbediti ruter koji podržava DDNS usluge, kako bi nakon uključivanja podataka o DDNS provajderu, ruter mogao automatski da ažurira adresu. Dokle god je ruter uključen, DDNS adresa će uvek biti ažurirana. Ukoliko korisnikov ruter ne podržava DDNS usluge, biće potreban lokalni klijent za pokretanje na često korišćenom računaru negde u korisnikovoj kućnoj mreži. Ova aplikacija će proveriti korisnikovu IP adresu, a zatim će se obratiti DDNS provajderu radi ažuriranja korisnikovog DDNS zapisa.

3.4. Konfigurisanje automatskog ažuriranja preko računara

Ažuriranje zasnovano na ruteru je daleko moćnije od korišćenja programa za ažuriranje preko računara, ali ukoliko korisnik nema ruter koji je prilagođen aktivnostima DDNS-a, jedini način za automatizaciju procesa ažuriranja pruža progam za ažuriranje zasnovan na računaru.

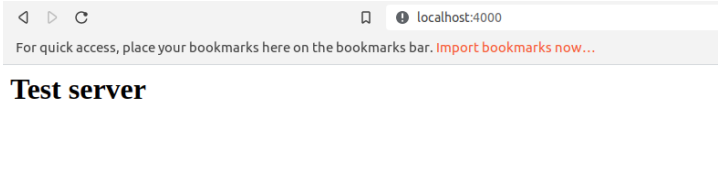
3.5. Usmeravanje imena domena na IP adresu korisnikove kućne mreže

Ovo je poslednji korak u procesu konfigurisanja DDNS-a. Sada je zamenjena teško pamtljiva IP adresa sa lako pamtljivim imenom. Ova konfiguracija neće zameniti već postojeće postavke korisnikove kućne mreže, sve što je radilo pre postavljanja DDNS-a, nastaviće sa radom i sa novom adresom DDNS-a.

Ukoliko se korisnik ranije povezivao na kućni mužički server, dok je na poslu na primer, umesto što je posećivao XXX.XXX.XXX.XXX:5900 (kućna IP adresa, port 5900), sada može da se poveže na isti, unosom noveDDNSadrese.com:5900.

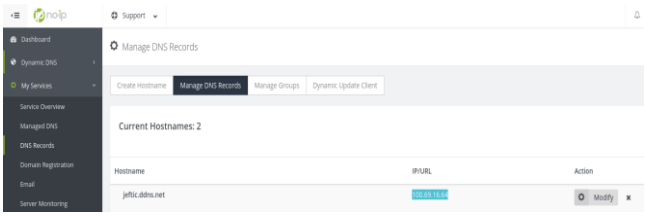
3.6. Pristup testnom serveru preko DDNS-a

Kreiran je primer test aplikacije za potrebe objašnjenja procesa deljenja pristupa korisnicima pomenutoj aplikaciji, upotrebom DDNS-a. Aplikacija je pokrenuta na lokalu, na portu 4000. Rezultat lokalnog pokretanja aplikacije je prikazan na Slika 2.



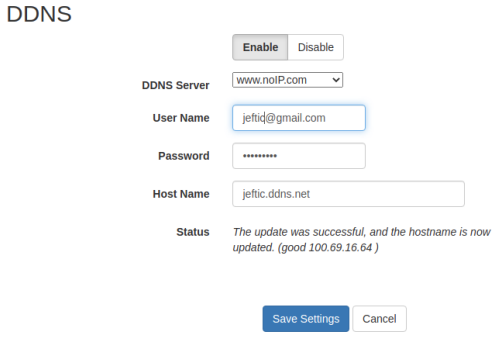
Slika 2. *Rezultat pokretanja aplikacije na lokalu*

Da bi korisnici bili u mogućnosti da pristupe aplikaciji, potrebno je da znaju IP adresu, a pošto je IP adresa u ovom slučaju dinamička, prilikom promene te IP adrese, korisnici neće imati pristup aplikaciji. Da bi se zaobišao ovaj problem potrebno je da se konfiguriše DDNS, koji će dinamičku IP adresu pretvoriti u statičku. U ovom primeru su korišćene usluge No-IP provajdera, za registraciju domena i povezivanje sa ruterom. Na Slika 3. je prikazan primer kreiranog hostname-a (jeftic.ddns.net) na pomenutom provajderu.



Slika 3. *Kreiran hostname na No-IP provajderu*

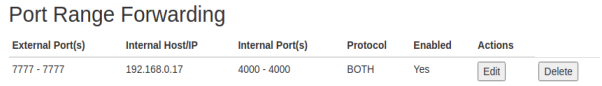
Sledeći korak jeste konfiguracija rutera. U odeljku za konfiguraciju DDNS-a, odabrati www.noip.com kao provajdera, uneti username, password i hostname. Prilikom klika na dugme „Save Settings“ dobiće se odgovor da li je hostname ažuriran. Na Slika 4. prikazan je izgled prethodno pomenutih konfiguracija.



Slika 4. *Konfiguracija DDNS-a na ruteru*

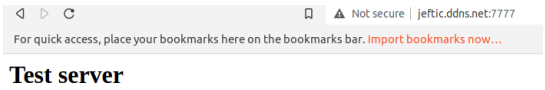
Da bi se pristupilo test aplikaciji potrebno je mapirati spoljne portove na određeni računar i njegov interni port. U ovom slučaju spoljašnji port će biti 7777, koji će se mapirati na lokalni računar na kome je pokrenuta

aplikacija, preko IP adrese tog računara (192.168.0.17) i porta na kome je pokrenuta aplikacija (4000). Mapiranje portova je prikazano na Slika 5.



Slika 5. *Mapiranje portova*

Ovim je omogućeno korisnicima da pristupe test aplikaciji putem pretraživača, unošenjem adrese jeftic.ddns.net:7777. Na Slika 6. je prikazan krajnji rezultat.



Slika 6. *Krajnji rezultat konfiguracije DDNS-a*

4. ZAKLJUČAK

Cilj obrađivanja ove teme je pojašnjenje načina na koji se može lako pristupiti kućnoj mreži sa bilo kog udaljenog mesta. Interpretirano rešenje nudi dinamički DNS. Kako bi bilo moguće implementirati ovu funkcionalnost, potrebno je da se korisnik opredeli za neki od provajdera koji nude DDNS.

Pored odabira provajdera, potrebno je još i odabrati ruter koji podržava DDNS, kako bi se na najlakši način omogućila ova funkcionalnost korisniku. Nakon implementacije svih potrebnih koraka, korisnik ima mogućnost da pristupi sadržaju na lokalnoj mreži, sa bilo kog mesta u svetu ukoliko ima pristup internetu i na taj način je u mogućnosti da na primer, podeli svoj domen sa prijateljima ili da kreira lokalni server za neku igricu koji bi takođe podelio sa nekim korisnicima, ili jednostavno da omogući drugim korisnicima da pristupe njegovoj aplikaciji putem interneta, kao što je to objašnjeno u radu.

5. LITERATURA

[1] „SAVREMENE IP MREŽE: ARHITEKTURE, TEHNOLOGIJE I PROTOKOLI“, Mirjana D. Stojanović, Vladanka S. Aćimović-Raspopović, 2012., Akademska Misao Beograd
[2] https://www.plus.rs/blog/osnove-funkcionisanja-dns-a-3/
[3] https://www.dynu.com/en-US/

Kratka biografija:

Vanja Jeftić rođena je u Novom Sadu 1997. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Primenjene računarske nauke odbranila je 2021.god.
kontakt: vanjajeftic97@gmail.com

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.738.1:766
DOI: https://doi.org/10.24867/16BE25Radic

ФОРЕНЗИКА СКЛАДИШТА У ОБЛАКУ

CLOUD STORAGE FORENSICS

Дијана Радић, *Факултет техничких наука, Нови Сад*

Област – ЕЛЕКТРОТЕХНИЧКО И РАЧУНАРСКО ИНЖЕЊЕРСТВО

Кратак садржај – *У раду су анализиране технике и алати форензике складишта у облаку. Главни фокус рада је на изазовима на које се наилази приликом бављења форензиком облака. Дат је осврт на оквир дигиталне форензике облака, на технике и процесе који се предузимају приликом истраге и на софтверске и хардверске алате који се користе. Такође, споменути су неки од правних аспеката форензике облака са којима треба бити упознат. У оквиру рада илустрована је и студија случаја која има за циљ да читав процес заокружи и симулира на хипотетичком примеру.*

Кључне речи: *форензика складишта у облаку, рачунарство у облаку, дигитална форензика, технике, алати, докази*

Abstract – *In this paper, cloud storage forensics technique and tools are analyzed. The main focus of the paper is on the challenges encountered when dealing with cloud forensics, digital cloud forensics framework, techniques and processes undertaken during the investigation, and the software and hardware tools used. Some of the legal aspects of cloud forensics, that forensics investigators need to be familiar with, are also mentioned. The paper also illustrates a case study that aims to complete and simulate the whole process on a hypothetical example.*

Keywords: *cloud storage forensics, cloud computing, digital forensics, techniques, tools, evidence*

1. УВОД

Складиште у облаку дизајнирано је за пословне или личне сврхе и омогућава корисницима разне услуге. Већина људи већ користи разне услуге рачунарства у облаку, а да тога нису ни свесни. Gmail, Google Drive па чак и Facebook и Instagram су апликације засноване на облаку.

Приступ облаку могућ је и преко телефона и преко рачунара. Један од главних аспеката форензичког истраживања складишта у облаку је открити шта је корисник урадио од тренутка претплате на услугу до краја коришћења услуге.

НАПОМЕНА:
Овај рад је проистекао из мастер рада чији је ментор био др Стеван Гостојић, ванр. проф.

Овај рад обрађује форензику облака. У другом одељку обрађено је рачунарство у облаку. Трећи одељак се бави дигиталном форензиком. Четврти одељак образлаже технике специфичне за форензику складишта у облаку, као и изазове са којима се дате технике суочавају. Кроз пети одељак уводе се хардверски и софтверски алати који се користе током истрага везаних за дигиталну форензику складишта у облаку. Једна од незаобилазних ставки сваког истражитеља који се бави форензиком облака јесте правни аспект ове теме који је обрађен у шестом одељку. Седми одељак доноси студију случаја на хипотетичком примеру како би се илустровао процес форензичке истраге у облаку који је обрађен кроз претходна поглавља. У осмом одељку дата су завршна разматрања.

2. РАЧУНАРСТВО У ОБЛАКУ

Амерички Национални институт за стандарде и технологију (NIST) је 2011. године објавио дефиницију која је често цитирана: „Рачунарски облак је модел који омогућава свуда присутан, погодан мрежни приступ дељивим рачунарским ресурсима (рачунарским мрежама, серверима, складишту података, апликацијама и сервисима), који на захтев корисника и уз минималну интеракцију са испоручиоцем услуга, могу бити брзо стављени на располагање кориснику или отказани".

Карактеристике облака су:

- Пружање услуге на захтев корисника
- Широк мрежни приступ
- Удруживање ресурса
- Брза еластичност
- Измерена услуга

Услуге у облаку су генерално категорисане у три модела услуге: (1) инфраструктура као услуга (IaaS - infrastructure as a service), (2) платформа као услуга (PaaS - platform as a service) и (3) софтвер као услуга (SaaS - software as a service); и четири модела примене: (1) јавни, (2) приватни, (3) хибридни и (4) заједнички. Количина контроле, одговорност и видљивост коју корисник има над оперативним системом, апликацијама и системским софтвером као што су веб сервер и системи за управљање базама података јесте фактор за разумевање разлика између модела.

2.1 Виртуализација

Виртуализација представља један од темеља на којем је изграђено рачунарство у облаку и без којег не би постојало. Виртуализација је технологија која

омогућава креирање ИТ услуга користећи ресурсе који су традиционално везани за хардвер. Омогућава коришћење пуног капацитет физичке машине дистрибуирањем њених могућности међу многим корисницима или окружењима [1].

2.1 Вишеклијентски систем

Вишеклијентски систем сугерише да један ресурс, у случају рачунарства у облаку један сервер, деле потенцијално више корисника. Количина корисника који деле одређене услуге у облаку разликује се по моделима услуге и примене. На пример, у IaaS моделу хардвер ће се делити на више закупаца али оперативни системи и системски софтвер неће. У PaaS-у оперативни систем и системски софтвер могу да се деле са другим корисницима, али апликације се не деле [2]. На крају у SaaS-у се може десити да се чак и иста инстанца физичке базе података дели са другим корисницима.

3. ДИГИТАЛНА ФОРЕНЗИКА

Дигитална форензика је научна дисциплина чији предмет су идентификација, прикупљање, чување, прегледање, анализа и презентација дигиталних доказа коришћењем научно и правно ваљаних метода и алата. Најважнији циљ форензичке истраге је да доведе у корелацију чињенице, особе и друге ентитете извучене са дигиталног уређаја [3].

Два принципа доминантна за дигиталну форензику јесу интегритет доказа и обезбеђивање ланца надлежности.

3.1. Процес дигиталне форензике

Постоји пет узастопних, али и итеративних фаза у процесу дигиталне форензике. Укратко, прва фаза је идентификација уређаја који би потенцијално могли да садрже доказе релевантне за истрагу. У другој фази се са идентификованих уређаја прикупљају докази. Трећа фаза јесте припрема сирових доказа како би се касније лакше обрадили и анализирали. Четврта фаза јесте коначно анализа где се покушавају разумети докази и утврдити след догађаја. Последња фаза је презентовање налаза суду или субјекту од интереса.

3.2. Форензика складишта у облаку

Форензика складишта у облаку представља подграну форензике мреже. Према дефиницији NIST-а 2014: „Форензика рачунарства у облаку је примена научних принципа, технолошке праксе, изведених и проверених методе за обраду прошлих догађаја у облаку идентификацијом, прикупљањем, чувањем, прегледом и извештавањем о дигиталним подацима у сврху олакшавања реконструкције ових догађаја“.

Форензика складишта у облаку има три димензије:

- Техничка димензија
- Правна димензија
- Организациона димензија

Техничка димензија: Ову димензију одликује употреба алата за извођење форензике рачунарства у облаку. Прикупљање података, форензика уживо, праћење комуникације преко мреже и декрипција су неки од задатака који се решавају. Алати се међусобно разликују у погледу њиховог развојног модела и услужног модела.

Правна димензија: Споразум о нивоу услуге (SLA - Service Level Agreement) је медиј за осигурање сигурности података између провајдера услуге у облаку и потрошача. Како би се гарантовала приватност, форензички поступак не би требало да крши ниједан закон и пропис у надлежности центра за обраду података.

Организациона димензија: Провајдер и клијент су главни актери форензичке истраге у облаку међутим постоји и трећа страна тј. три врсте трећих страна које су укључене у истражни процес:

- ИТ стручњаци
- Руковаоци инцидентима
- Правни саветници [4].

Две фазе дигиталне форензике које су највећи изазов за форензику облака јесу идентификација и прикупљање.

4. ТЕХНИКЕ У ФОРЕНЗИЦИ ОБЛАКА

4.1. Анализа лог записа из складишта у облаку

Лог је порука коју рачунарски систем, уређај или софтвер генерише као одговор на неки стимуланс. Стимуланс може да буде акција корисника, грешка у раду апликације или нека друга активност. Типична лог порука се састоји из временске одреднице, извора и података. Логови могу да имају више извора настанка међу њима оперативне системе, рутере, свичеве, бежичне приступне тачке, firewall (заштитни зид), VPN сервере, штампаче итд. Постоји више врста лог фајлова у зависности од тога где су настали и која им је намена.

Због повећане употребе интернета и иновација у технологијама облака, повећао се број рањивости и напада у облаку. Сви ови напади могу да се прате кроз логовање [5].

4.2. Прикупљање доказа кроз веб претраживач

Веб претраживач може бити кључан за дигиталну истрагу, складишти податке о употреби интернета као и неке себи својствене податке у зависности од врсте претраживача. Неки од артефаката који се прикупљају су историја прегледања, кеш, сесије, колачићи, итд.

Према последњим истраживањима највише се користи Chrome са преко 60% удела, након њега иде Safari са око 18%. Када се утврди која врста претраживача коришћена на идентификованим уређајима, артефакти се траже на локацијама на којима их претраживачи складиште.

4.3. Праћење мрежног саобраћаја

Приступ облаку је немогућ без интернета, зато је и праћење мрежног саобраћаја значајно ако за то постоји могућност.

Користећи разне алате могуће је ухватити информације које се преносе и користити њих даље кроз истрагу. Мрежна форензика је често везана за праћење и анализу нестабилног и динамичког саобраћаја на мрежи.

Форензичар тражи податке који могу бити у једном од три облика: статичном, покретном или унутар процеса који се извршава. Мрежна форензика се односи на приступ који укључује употребу наменске инфраструктуре која може прикупљати, очувати и анализирати мрежни саобраћај у сврху истраге [6].

4.4. Изазови техника форензичке истраге облака

Неки од изазова у техникама за форензику облака по фазама форензичке истраге:

Идентификација:

- Није позната физичка локација података,
- Питање јурисдикције,
- Зависност од провајдера услуге у облаку
- Може да се догоди да сви подаци нису на једном месту, због децентрализованости података.
- Сегрегација доказа, како прикупити само релевантним подацима са сервера који опслужује више клијената

Прикупљање:

- Недоступност података
- Вишеклијентски систем
- Обрисани подаци

Анализа:

- Енкрипција података
- Непостојање јединственог шаблона за лог фајлове

5. АЛАТИ У ФОРЕНЗИЦИ ОБЛАКА

Алати који се развијају специјално за дигиталну форензику морају да гарантују прецизност и поузданост.

5.1. Мобилни теренски комплет – Mobile Field Kit

Унутар комплета се налази лиценцирани софтвер уграђен на Windows систему, каблови, прикључци за пуњење, камера за снимање телефона, једна средња и једна већа Фарадејева врећа за телефоне и таблете, све је то спаковано у ојачани кофер за пренос.

Како се докази везани за форензику облака често налазе у центрима података и није могуће испитивања извршити у лабораторији, мобилни теренски комплет је неопходан за прикупљање доказа и њихову заштиту.

5.2. UFED Cloud Analyzer

UFED Cloud Analyzer је производ компаније Cellebrite. Овај алат може да се користи за прикупљање, очување и анализу доказа са облака. Подаци могу да се сортирају, филтрирају, претражују и да се аутоматски генеришу извештаји.

Има следеће карактеристике:

• Прибавља историју прегледа и претраге и сортира у хронолошком реду.

• Прикупља податке са најпопуларнијих друштвених мрежа и складишта у облаку, уз помоћ креденцијала корисника или токена прикупљеног са уређаја.

5.3. FTK Imager

FTK Imager је бесплатан софтверски алат, развијен од стране компаније AccessData. Главна функција овог алата је да направи копије компјутерских података без њихове промене. Има следеће карактеристике:

• Могућност креирања форензичких слика са хард дискова, CD-ова, DVD-ова, целог фолдера или појединачног фајла итд.

• Може да поврати фајлове који су обрисани из корпе за смеће, уколико подаци нису преписани неким новим подацима.

• Може да израчуна хеш вредност креираних форензичких слика користећи једну од две доступне опције Message Digest 5 (MD5) и Secure Hash Algorithm (SHA-1).

5.4. Wireshark

Wireshark је алат отвореног кода који се употребљава за тестирање мреже, решавање проблема и проверу различитог саобраћаја који пролази кроз рачунарску мрежу анализом мрежних пакета.

Сва комуникација и размена података корисника, преко свог налога, са услугама складиштења у облаку се одвија и преноси путем интернета. Потенцијалним анализирањем података прикупљених са мреже могуће је доћи до доказа релевантних за ток истраге.

6. ПРАВНИ АСПЕКТИ У ФОРЕНЗИЦИ ОБЛАКА

6.1. Територијалност података – губитак локације

Већина провајдера услуга складишта у облаку има неколико центара података расутих по целом свету ради редундантности података и оптимизације перформанси.

Приликом сваког отпремања датотеке у облак, она се аутоматски множи и складишти у најмање две (обично три) засебне географске и физичке локације, обично у различитим земљама.

Правна теорија наводи да је законодавство које се треба применити одређено местом на коме се одиграо кривични догађај. Други територијални приступ скреће пажњу на медиј са којим је злочин извршен. Локација крајњег корисника је трећи приступ изазова територијалности података.

Држављанство починиоца или жртве се користи као одлучујући критеријум у четвртом територијалном приступу, без обзира на локацију на којој је злочин извршен [7].

Како би се примена овог принципа регулисала на међународном нивоу, суверене државе међусобно склапају уговоре.

6.2. Власништво над садржајем облака

Провајдер услуга у облаку не поседује ни један податак са правне тачке гледишта.

Да би се неко могао сматрати кривично одговорним за „поседовање“ одређене датотеке, услов који мора да испуни јесте да ју је барем једном „прегледао“ [7].

Може се тврдити да би се облак требао сматрати и правно третирати као виртуелни и удаљени екстерни медиј за складиштење (тј. заправо да је то продужетак сваког дигиталног уређаја који му има приступ). Кључни елемент на основу којег је основана кривична одговорност је намерно и свесно приступање упитним датотекама личним чином [7].

6.3. Очување података

Чланови 16. и 29. Будимпештанске конвенције о сајбер криминалу [8] наводе да су земље потписнице дужне да предузму законодавне мере у вези са потенцијално убрзаним очувањем одређених ускладиштених рачунарских података који су ускладиштени помоћу рачунарских система који се налазе на њиховој територији, посебно тамо где постоје основане сумње да су рачунарски подаци посебно рањиви на губитак или измену. Међу земљама потписницама се налази и Србија.

7. СТУДИЈА СЛУЧАЈА

7.1. Припрема

Претпоставка је да је полиција добила дојаву о инциденту који укључује складиште у облаку. Потребно је утврдити где је провајдер облака регистрован, како би се утврдило да ли поседују јурисдикцију да обаве истрагу.

7.2. Идентификација

Како су данас мобилни телефони и рачунари у широкој употреби, претпоставка је да су ови уређаји заплењени приликом претреса. Пронађени уређаји се фотографишу на месту проналаска и праве се забелешке о њима укључујући забелешке о моделу, марки, серијском броју уређаја итд.

7.3. Прикупљање

Ако је којим случајем рачунар био укључен, потребно је прикупити несталну меморију. Ово може да се обави користећи FTK Imager. Неопходно је да истражитељ пажљиво изврши прикупљање несталне меморије јер ће након гашења уређаја она нестати. Са истим алатом може да се направи форензичка слика хард диска.

За прикупљање података из складишта у облаку постоји више сценарија у зависности од јурисдикције. Провајдер услуге може на захтев истражитеља да прикупи податке, пратећи савете које добије и користећи правно ваљане технике и алате, ако истражитељ то не може лично да уради. Ово је сценарио уколико истражитељи имају надлежност.

7.4. Прегледање

У фази прегледања сви прикупљени уређаји и креиране копије се преносе у лабораторију. Приликом прегледања истражитељ мора да покуша да дешифрује криптоване податке и да поврати податке који су обрисани.

7.5. Анализа

Када заврши припрему доказа прелази се на анализирање и извлачење закључака.

7.6. Презентација

На крају се креира извештај који описује процес и проналаске истраге. Одабрани релевантни пронађени фајлови се могу доставити копирани на CD-у.

8. ЗАКЉУЧАК

Област форензике складишта у облаку је у константном развоју јер постоје проблеми који још нису решени. Такође, са напретком технологије појављиваће се и нови проблеми и изазови који ће захтевати посебну пажњу.

Сваки случај или инцидент јесте јединствен. Оквир истраге може бити исти, али алати и технике се морају прилагодити случају. Мора се водити рачуна да примењени алати и технике морају имати кредибилитет на суду и да су довољно испитани да се може гарантовати интегритет доказа који се помоћу њих обрађују.

9. ЛИТЕРАТУРА

[1] Red Hat, Inc., What is virtualization?, Red Hat, June 2018. [Online]. Доступно: https://www.redhat.com/en/topics/virtualization/what-is-virtualization (приступљено у септембру 2021.)

[2] Greg Gogolin, PhD, CISSP, Digital Forensics Explained, 2nd ed. Boca Raton, Taylor & Francis Group, LLC, 2021

[3] Flora Amato, Aniello Castiglione, Giovanni Cozzolino, Fabio Narducci, A semantic-based methodology for digital forensics analysis, Journal of Parallel and Distributed Computing vol. 138 , pp. 172–177, Jan. 2020

[4] A. M. Poorvi Jain, "Review of Cloud Forensics: Challenges, Solutions and Comparative Analysis,"International Journal of Computer Applications, pp. 28-34, 2019.

[5] Thankaraja Raja Sree and Somasundaram Mary Saira Bhanu, "Data Collection Techniques for Forensic Investigation in Cloud", Intechopen Digital Forensic Science, Sep 2020, [online] Available: http://www.grandviewresearch.com.

[6] N. Raza, "Challenges to network forensics in cloud computing," 2015 Conference on Information Assurance and Cyber Security (CIACS), 2015, pp. 22-29, doi: 10.1109/CIACS.2015.7395562.

[7] Karagiannis, C.; Vergidis, K. Digital Evidence and Cloud Forensics: Contemporary Legal Challenges and the Power of Disposal. Information 2021, 12, 181. https://doi.org/10.3390/info12050181

[8] Council of Europe. Convention on Cybercrime. [Online]. Доступно: https://www.coe.int/en/web/conventions/full-list/-/conventions/rms/0900001680081561 (приступљено у септембру 2021.)

Кратка биографија:

Дијана Радић рођена је у Сомбору 1997. године. Завршила је основну школу „Никола Тесла“ у Бачком Брестовцу, затим гимназију „Вељко Петровић“ у Сомбору. Дипломирала је 2020. на Факултету техничких наука у Новом Саду, смер Рачунарство и аутоматика.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.38:620.9
DOI: https://doi.org/10.24867/16BE26Lucic

SIMULACIJA PROPADA NAPONA U DISTRIBUTIVNOJ TEST MREŽI SA OBNOVLJIVIM IZVOROM ENERGIJE

SIMULATION OF VOLTAGE SAGS IN A DISTRIBUTION TEST GRID WITH A RENEWABLE ENERGY SOURCE

Nikola Lučić, Vladimir A. Katić, Aleksandar M. Stanisavljević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Uradu je posmatrana IEEE 33-bus test mreža u okviru koje su simulirani efekti propada napona, koristeći Typhoon HIL Control Center programski paket. Data je studija slučaja simulacije uticaja različitih vrsta kvarova, ako je dodata jedna solarna elektrana kao obnovljivi izvor energije. Posmatran je harmonijski otisak propada napona na pojedinim sabirnicama u toj test mreži. Studija slučaja pokazala je da se razne vrste kratkih spojeva manifestuju kroz različite karakteristike propada napona, odnosno harmonijskih otisaka. Uočen je i problem pojave višeg nivoa harmonika tokom propada uzrokovan ograničenjima kontrolnog sistema Typhoon-HIL-a.*

Ključne reči: *IEEE 33-bus test mreža, harmonijski otisak, propadi napona.*

Abstract – *The paper observes the IEEE 33-bus test grid in which the effects of voltage sags are simulated, using the Typhoon HIL Control Center software package. A case study of the simulation of the impact of different types of failures, if one solar power plant is added as a renewable energy source, is given. The harmonic footprints of the voltage sags on individual busbars in that test gridwere observed. The case study showed that different types of failures are manifested through different characteristics of voltage sags, i.e., harmonic footprints. The problem of the appearance of higher levels of harmonics during the sags caused by the limitations of the Typhoon-HIL control system was also noticed.*

Keywords: *IEEE 33-bus test grid, harmonic footprint, voltage sags.*

1. UVOD

Nagli razvoj računarstva izazvao je veliki skok unapretku inženjerstva zasnovanog na modelima, uvođenjem različitih inoviranih softverskih paketa i alata, koji se koriste za ovu namenu, a koji su svoje utočište našli u različitim granama industrije. Nije teško zamisliti zbog čega su se računarski programi ove vrste pokazali dragocenim u istraživanjima i na polju elektroenergetike, gde je prelaskom sa fizičkih na testiranje virtuelnih sistema, ostvarena ne samo sloboda u pogledu opsega željenih ispitivanja i jednostavnosti izmena u test-sistemu, nego i neprocenjiva dobit, uštedom vremena, materijalnih sredstava, ali i očuvanjem sigurnosti ljudskih života.

Iako sporijeg razvoja nego računartsvo, elektroenergetika se takođe unapređivala u svojim oblastima proizvodnje i distribucije, s tim da su ove oblasti ostale suštinski nepromenjene sve do kraja 90-ih godina prošlog veka, kada ulaze u period najvećih promena u kom se nalaze i danas. Kroz proces deregulacije došlo je do izmena u vlasničkoj strukturi elektroenergetskog sistema i razbijanja njegove tkzv. vertikalne organizacije. Tome su doprineli i sve veća zabrinutost za životnu sredinu, nagoveštaj skorog iscrpljavanja neobnovljivih fosilnih goriva, pojava novih rešenja za ekonomičnije iskorišćavanje obnovljivih izvora električne energije, želja za smanjenjem emisije gasova staklene bašte i umanjenje efekata ne klimatske promene. Tehničke podloge za ove promene dao je paralelni razvoj energetske elektronike, nove tehnologije snažnih poluprovodnika, kao i efikasna rešenja digitalne (mikroprecesorske) kontrole, metoda brze komunikacije i kvalitetni softverski alati.

Kvalitet električne energije je oblast, koja se pojavila u ovom talasu promena. Po svojoj prirodi, ona je ta koja predstavlja odraz dinamike rada nelinearnih potrošača, poput uređaja električne elektronike i elektromotornih pogona, sa statikom elektroenergetskog sistema. Pojava distribuiranih izvora i mikromreža uvodi nova pitanja na ovakav način razumevanja, menja postojeće zahteve i postavlja nove [1].

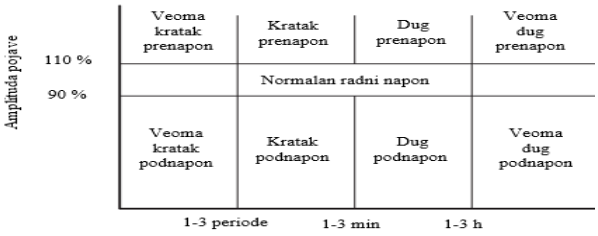
Ovaj rad bavi se jednim od najozbiljnijih poremećaja kvaliteta električne energije – propadima napona. Cilj je da koristeći standardnu IEEE 33-bus test mrežu odgovori na pitanje uticaja kvarova u distributivnoj mreži na karakteristike propada napona na pojedinim sabirnicama koristeći najsavremeniji digitalni simulator za rad u realnom vremenu baziran na *Typhoon HIL Control Center* programskom paketu.

2. PROPADI NAPONA

Propadi napona predstavlja kratkotrajno smanjenje efektivne (RMS) vrednosti napona, koje je uzrokovano kvarovima u EES, njegovim preopterećenjem, a može se javiti i pri startovanju velikih opterećenja poput velikih motora. Prema IEEE standardu 1159-2009, propadi napona su definisani kao redukcija napona u rasponu od 10% do 90% nominalne vrednosti napona, kada je frekvencija sistema nominalna, i kada je trajanje poremećaja u rasponu od pola periode do jednog minuta [2]. Amplituda i trajanje pojave mogu da se koriste za klasifikaciju varijacija u naponu što je prikazanona slici 1.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Vladimir Katić, red.prof.

Istraživanjem harmonijskog spektra tokom poremećaja napona u distributivnim mrežama, primećeno je da amplituda harmonika niskog frekventnog spektra ima gotovo trenutni porast, čija se promena amplitude daleko brže odvija od promene osnovne komponente napona (osnovnog harmonika) i brže se prenosi kroz mrežu od promene ukupne amplitude napona [2]. Nakon ovog saznanja, ispitivane su različite kombinacije harmonika, dok se nije došlo do one koja daje najbolje rezultate, a to je set od drugog, trećeg, petog i sedmog harmonika (HDU2357), koji je u radu [3] originalno nazvan Harmonijski otisak (*Harmonic Footprint*).



Sl. 1. *Grafik amplituda-trajanje za klasifikaciju pojava kvaliteta električne energije [2]*

3. DISTRIBUTIVNE TEST MREŽE

Test mreža je model realne distributivne mreže, koja reprodukuje ponašanje i karakteristike stvarne mreže, uključujući i specifične pojedinosti unutar lokaliteta u kom se mreža nalazi. Zahvaljujući njima, rezultati istraživanja se mogu jednostavno proveravati i upoređivati. Danas se u svetu koristi veliki broj test mreža, koje su propisali IEEE, CIGRE ili druga značajna elektroenergetska udruženja [4].

Za potrebe ovog rada, korišćena je *IEEE 33-bus* mreža. Ova mreža se često koristi za istraživanja u oblasti tokova snaga. Sastoji se od 33 sabirnice (trofazna čvora), raspoređenih na 4 glavna fidera. Čvorovi su povezani sa 32 trofazna voda. Potrošači (skupovi više sabranih potrošača) povezani su na svaki od 33 čvora sistema.

4. MODELOVANJE PROPADA NAPONA

Za istraživanje uticaja propada napona u test mrežama do sada su korišćeni razni pristupi. U radu [5] razmatrani su propadi napona na *IEEE 3-bus* i *IEEE 9-bus* test mrežama uz korišćenje *DIgSILENT Power Factory* softverskog alata. Slična analiza uz primenu istog softverskog alata, ali na *IEEE 13-bus* test mreži prikazana je u [6]. U [7] su propadi napona razmatrani kroz *Matlab/Simulink* model *IEEE 13-bus* test mreže, dok su u [8] za ispitivanje korišćene neuralne mreže.

Za potrebe ovog rada, korišćen je *Typhoon HIL Control Center (THCC)* programski paket [9]. Osnovna mogućnost koju nudi je izrada modela i njihovo simuliranje u realnom vremenu. U okviru ovog rada korišćeni su *Schematic Editor* i *HIL SCADA*, pa je njima posvećena posebna pažnja.

Schematic Editor omogućava izradu modela visoke vernosti radi njihovog simuliranja u realnom vremenu. Kreiranje modela se odvija u grafičkom okruženju, gde korisnik može da kreira električne i kontrolne delove modela, parametrizuje njegove komponente i kompajlira gotove modele pomoću ugrađenog kompajlera.

HIL SCADA predstavlja grafičko okruženje koje omogućava interakciju sa simulacijom u realnom vremenu. HIL SCADA preuzima kompajlirane simulacione modele na HIL platformu i upravlja procesom simulacije, parametrima i izlazima.

Model mreže je napravljen koristeći primer modela *IEEE 33-bus* mreže. Nad ovim modelom izvršeno je niz modifikacija: eliminisani su spojni vodovi koji su u originalnom modelu omogućavali prelaz iz radijalne u upetljanu distributivnu mrežu (za potrebe ovog rada to nije bilo neophodno), uvedene su komponente za simuliranje kratkih spojeva i na nju je povezan DER u vidu fotonaponske (FN) elektrane. Na slici 2 prikazan je konačni model sistema u *Schematic Editor*-u.

Za potrebe ovog rada, razvijen je blok za ekstrakciju viših harmonika iz talasnog oblika napona, kao i blok za računanje harmonijskog otiska.

Blok za ekstrakciju viših harmonika napravljen je pomoću *Harmonic Analyzer* komponente. Unutrašnjost bloka prikazana je na slici 3.

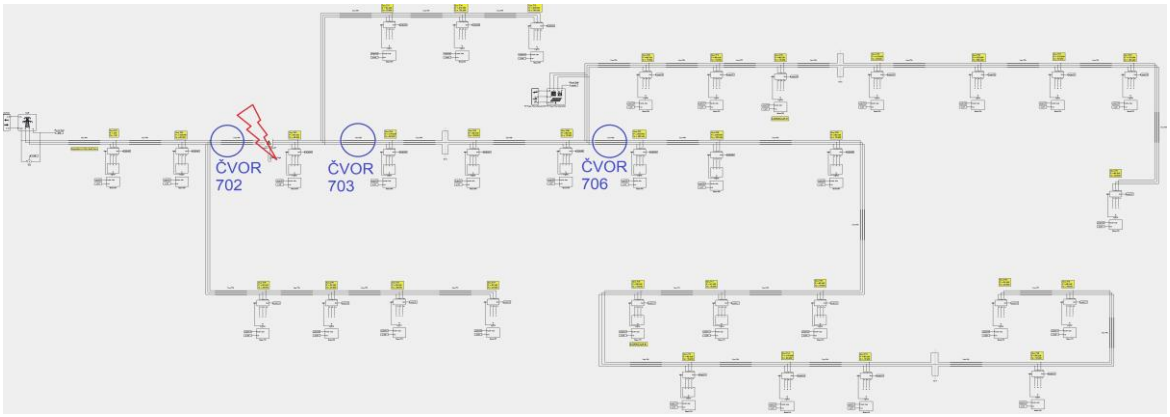
Blok za formiranje harmonijskog otiska napravljen je na osnovu formule harmonijskog otiska koja glasi [3]:

$$HDU2357 = \frac{\sqrt{\sum_{n=2,3,5,7} U_n^2}}{U_1} \cdot 100 \ [\%] \qquad (1)$$

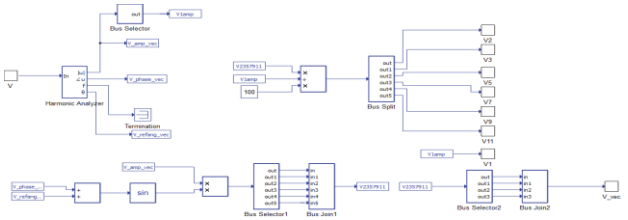
Unutrašnjost bloka za formiranje harmonijskog otiska prikazana je na slici 4.

5. SIMULACIJE PROPADA NAPONA

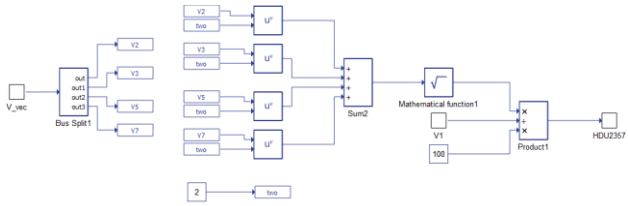
Korišćenjem *HIL SCADA* okruženja, simulirani su kratki spojevi i merene željene vrednosti. Posmatrane su trenutne vrednosti napona, efektivne vrednosti napona radi uvida u njihove propade, talasni oblici viših harmonika napona pogođenih faza (2-gi, 3-ći, 5-ti, 7-mi, 9-ti i 11-ti), kao i harmonijski otisci pogođenih faza.



Sl. 2. *Model IEEE 33-bus test mreže u Typhoon-HIL paketu*



Sl. 3. *Blok za ekstrakciju viših harmonika*



Sl. 4. *Blok za formiranje harmonijskog otiska*

Mesto kratkog spoja nalazi se na vodu, koji povezuje čvor (bus) #702 i čvor #703. Vreme snimanja na osciloskopu je iznosilo 1 s. Posmatrane su vrednosti u čvoru #706 (slika 2). U svim simulacijama, aktivna snaga FN elektrane iznosila je 1000 kW, odnosno 0,5 njene nominalne vrednosti, dok je reaktivna snaga iznosila 400 kVAr, odnosno 0,2 njene nominalne vrednosti. Temperatura u svim simulacijama je nepromenjena i iznosila je 25˚C. Isto važi i za iradijaciju, koja je iznosila 1000 W/m^2.

U okviru prve grupe simulacija, istraživani su tipski kratki spojevi. Simulirani su jednopolni kratak spoj faze A (A-N), zatim faze B, dvopolni kratak spoj faza A i B i kratak spoj faza A i B sa zemljom. Trajanje svakog od kratkih spojeva iznosilo je 0,3 s. Zbog ograničenja prostora, ovde će biti prikazani samo rezultati vezani za propad napona usled jednopolnog kratkog spoja A-N.

U okviru druge grupe simulacija, simulirani su neki složeniji tipovi kratkih spojeva. Iako je mogućnost njihove pojave manja u odnosu na slučajeve iz prve grupe, ona realno postoji. Mesto kratkog spoja, kao i mesto sa kog su posmatrane vrednosti ostalo je isto kao i u prvoj grupi simulacija. Ovde je prikazan samo jedan slučaj, tj. simuliran je složen kvar kroz nastanak jednopolnog kratkog spoja faze B, koji se dalje razvija u dvopolni kratak spoj faza A i B sa zemljom.

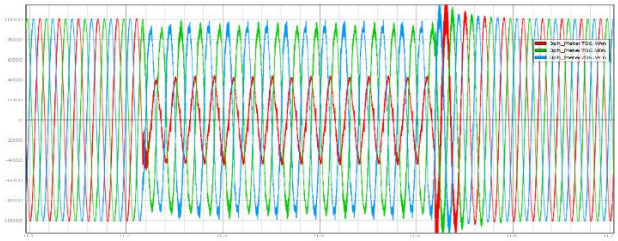
5.1 Jednopolni kratak spoj faze A (A-N)

U toku kvara javlja se značajno izobličenje talasnog oblika napona za sve tri faze, odnosno propad napona, a najviše faze A (slika 5). Prikaz promene efektivne (RMS) vrednosti napona dat je na slici 6. Vidi se da je propad faze A najizraženiji i iznosi 37,06% nominalnog. Propadi druge dve faze su manji i njihove vrednosti su 89,99% nominalnog za fazu B, odnosno 89,61% za fazu C.

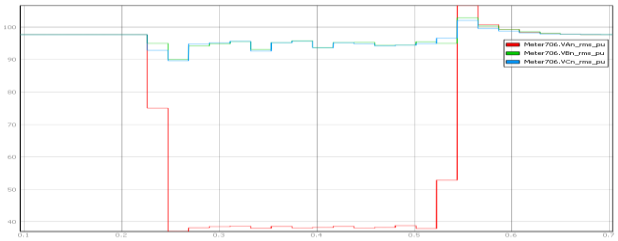
Tokom kvara javljaju se viši harmonici, koji su najizraženiji u tranzijentnim delovima propada, odnosno na početku i na kraju, i prikazani su na slici 7. Na početku propada izmerena je vrednost drugog harmonika od 20,94% osnovnog, odnosno vrednost trećeg od 5,43% osnovnog.

Može se primetiti da prisustvo harmonika postoji i tokom samog kvara, ali ovo se pripisuje FN elektrani, koja sadrži elektroenergetske pretvarače i uzrokovano je određenim ograničenjima kontrolnog Typhoon-HIL sistema. Na kraju propada drugi harmonik iznosi 21,68%, a treći 16,52%.

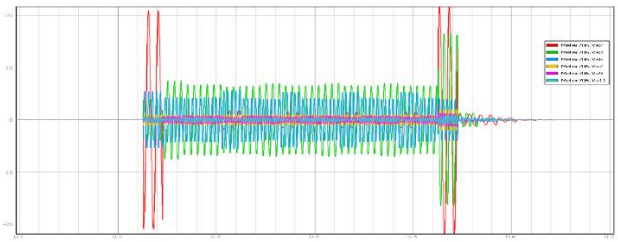
Harmonijski otisak propada napona prema (1) dat je na slici 8. Može se uočiti očekivani karakteristični oblik na početku i kraju propada, s tim da je na početku izmerena vrednost od 29,39% u piku, a na kraju 20,81% u piku.



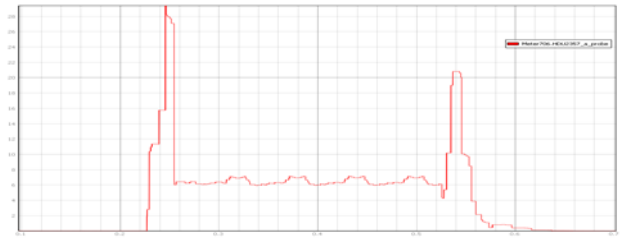
Sl. 5. *Talasni oblici napona u prvoj simulaciji*



Sl. 6. *RMS vrednosti napona u prvoj simulaciji*



Sl. 7. *Harmonici napona pogođene faze prve simulacije*



Sl.8. *Harmonijski otisak pogođene faze prve simulacije*

5.2 Jednopolni (B–N) u dvopolni sa zemljom (A–B–N)

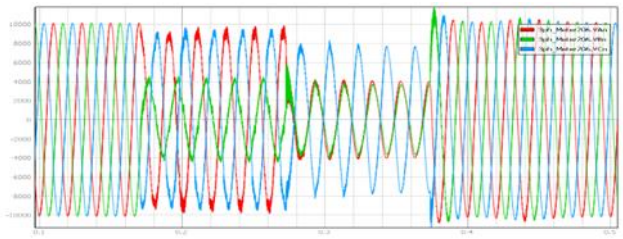
Simulacija je programirana tako da oba kratka spoja imaju trajanje od 0,1 s. Talasni oblici napona u drugoj simulaciji su prikazani na slici 9, gde se jasno mogu videti da jednopolni kvar u fazi B (B-N) prelazi u dvopolni faza A i B sa zemljom (A-B-N).

Posmatranjem RMS vrednosti napona, u prvom delu izmereno je da propad faze B iznosi 37,26% nominalnog napona, dok su ostale dve faze na granici dozvoljenog (propad za fazu A iznosi 90,15% nominalnog, dok za fazu C 89,88%). U drugom delu dolazi do intenziviranja propada, koji sada iznosi 32,93% nominalnog za fazu B, 37,29% za fazu A, te 72,6% za fazu C. Pri povratku u normalno stanje primećen je kratkotrajan porast napona od par procenata dok se napon ne ustali na vrednost pre kvara. RMS vrednosti napona sve tri faze prikazane su na slici 10.

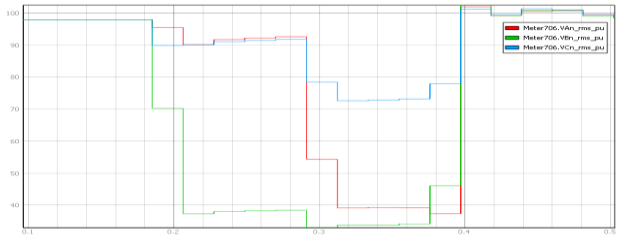
Za posmatranje viših harmonika kao i harmonijskog otiska izabrana je faza B (slika 11). Pri nastanku kvara, izmerena je amplituda drugog harmonika u iznosu od 27,59% osnovnog harmonika. U ovom delu zapaženo je i

prisustvo trećeg harmonika, čija vrednost iznosi 11,93%. Pri prelasku iz jednopolnog u dvopolni kratak spoj sa zemljom, vrednost amplitude drugog harmonika je 23,13%, dok se od ostalih ističe peti harmonik sa 6,46%. Pri povratku u normalno stanje, opet se javlja drugi harmonik sa vrednošću od 35,86%.

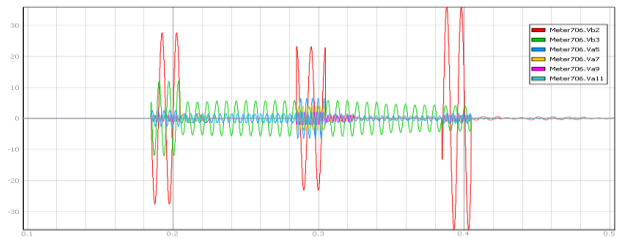
Harmonijski otisak prikazan je na slici 12. Mogu se primetiti tri otiska (skoka), koji odgovaraju trenutku nastanka, prelaska iz jednog kvara u drugi i trenutku prestanka propada. U trenutku nastanka kvara, izmerena je vrednost otiska od 38,12% u piku. Na narednom skoku, koji odgovara prelasku iz jednopolnog u dvopolni kratak spoj sa zemljom, izmerena je vrednost u piku od 18,01%. Na poslednjem skoku, koji odgovara prelasku iz dvopolnog kratkog spoja u stanje pre kvara, u piku je izmereno 25,98%.



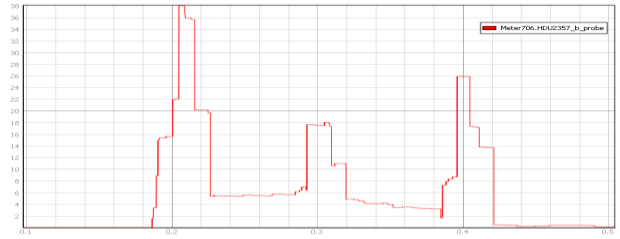
Sl. 9. *Talasni oblici napona u drugoj simulaciji*



Sl. 10. *RMS vrednosti napona u drugoj simulaciji*



Sl. 11. *Viši harmonici napona u drugoj simulaciji*



Sl. 12. *Harmonijski otisak u drugoj simulaciji*

6. ZAKLJUČAK

Posmatrajući set odabranih viših harmonika napona, utvrđeno je da su, kako nastanak kratkog spoja, tako i povratak sistema u stanje nakon kvara, praćeni skokovitim porastom određenih viših harmonika napona pogođene faze. U nekim slučajevima, u tranzijentnim delovima su izmereni pikovi harmonijskog otiska koji dostižu skoro 40%, što može imati izuzetno negativne posledice po rad sistema. Posmatranjem složenijih tipova kvarova, pokazano je da su oni, pošto izazivaju više tranzijenata tokom trajanja kvara, a samim tim i više skokova viših harmonika, još „opasniji“, tj. negativni efekti na elemente u mreži su izraženiji.

7. LITERATURA

[1] “Power Quality in Modern Power Systems”, Editors: P. Sanjeevikumar, C. Sharmeela, J.B. Holm-Nielsen, P. Sivaraman, Academic Press, London, 2020.

[2] A.M. Stanisavljević, “Nova metoda detekcije propada napona u mreži sa distribuiranim generatorima”, Dokt.diser., Fakultet tehničkih nauka, Novi Sad, 2018, https://nardus.mpn.gov.rs/handle/123456789/11090

[3] V.A. Katic, A.M. Stanisavljevic, “Smart Detection of Voltage Dips Using Voltage Harmonics Footprint”, IEEE Trans. on Industry Applications, Vol.54, No.5, Sep./Oct. 2018, pp.5331-5342.

[4] A.M. Stanisavljević, V.A. Katić, B. Dumnić, B. Popadić, “A Brief Overview of the Distribution Test Grids with a Distributed Generation Inclusion Case Study”, Serbian Journal of Electrical Engineering, Vol.15, No.1 , 2018, pp.115-129.

[5] J. Toholj, V.A. Katić, A.M. Stanisavljević, „Modelovanje i analiza uticaja propada napona primjenom test mreža sa distribuiranim generatorima”, Zbornik radova FTN, God.36, br.1, 2021, pp.123-126.

[6] P. Perge, V.A. Katić, A.M. Stanisavljević, „Ispitivanje uticaja obnovljivih izvora primenom test mreže”, Zbornik radova FTN, God.36, br.3, 2021, pp.480-483.

[7] V. Vidačić, V. Katić, “Uticaj obnovljivih izvora energije na propade napona u distributivnim mrežama”, Zbornik radova FTN, God.36, br.7, 2021, pp.1287-1290.

[8] I. Vasić, V. Katić, A. Stanisavljević, „Primena neuralnih mreža za detekciju propada napona na primeru rada test mreža”, Zbornik radova FTN, God. 36, br.3, 2021, pp.436-439

[9] Priručnik za korišćenje Typhoon HIL Control Center, https://www.typhoon-hil.com/documentation/typhoon-hil-software-manual/topics/software_manual_introduction.html

Kratka biografija:

Nikola Lučič rođen je 1996. god. u Subotici. Osnovne studije završio je na Fakultetu tehničkih nauka 2019. god., a master 2021. god. iz oblasti Elektrotehnike i računarstva.

Vladimir A. Katić rođen je 1954. god. u Novom Sadu. Doktorirao je na Univerzitetu u Beogradu 1991. god. Od 2002. god. je redovni profesor Univerziteta u Novom Sadu. Oblasti interesovanja su mu energetska elektronika, kvalitet električne energije, obnovljivi izvori električne energije i električna vozila.

Aleksandar M. Stanisavljević, rođen je u Beogradu 1988. god. Doktorirao ne na Univerzitetu u Novom Sadu 2019. god. gde je trenutno u zvanju docenta. Oblast interesovanja su mu integracija obnovljivih izvora energije na mrežu i kvalitet električne energije.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.31
DOI: https://doi.org/10.24867/16BE27Simic

ROBUSTNI KONTROLER MAKSIMALNE EFIKASNOSTI U POGONU SINHRONE MAŠINE ZA ELEKTRIČNA VOZILA

ROBUST EFFICIENCY CONTROLLER IN SYNCHRONOUS MACHINE ELECTRIC VEHICLE DRIVES

Stefan Simić, Darko Marčetić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu teorijski je izvršena analiza modela niskonaponske sinhrone mašine sa permanentnim magnetima na rotoru kao i analiza gubitaka odgovarajućeg motora. Prikazane su dve klase naprednih upravljačkih algoritama: prva za minimizaciju gubitaka LMC, i druga, strategija za uvećanje momentnog kapaciteta sinhrone mašine. Diskutovane su pojedinačno osnovne osobine analiziranih pristupa. Verifikacija teorijskih razmatranja je izvršena putem simulacija. Na osnovu prikazanih rezultata izvedeni su relevantni zaključci o performansama energetski efikasnog pogona.*

Ključne reči: *MTPA, sinhrona mašina, LMC*

Abstract – *Theoretical analysis of low voltage permanent magnet synchronous motor model as well as it's energy analysis was presented. Two advanced control approaches were discussed: efficiency maximization by LMC, and maximum torque per ampere strategy. The advantages of the presented algorithms were thoroughly discussed. Verification of the theoretical considerations were given through the simulations. Performance evaluation of the energy optimized traction drive was given and corresponding conclusions were derived.*

Keywords: *MTPA, synchronous motor, LMC*

1. UVOD

Upotreba vozila na električni pogon postaje sve veći trend u svetu. Sve veću primenu u tim pogonima beleži trofazna sinhrona mašina sa permanetnim magnetima (*eng. Permanent Magnet Synchronous Machine – PMSM*). Moderne tehnologije izrade baterija zasnovane na litijum–jonskim ćelijama, još uvek ne pružaju specifične gustine snaga, dovoljne za potrebe električnog transporta na velikim udaljenostima.

S tim u vezi danas se u svetu posvećuje velika pažnja projektovanju energetski efikasnih pogona, usled čega se povećava iskorišćenost baterija i sam domet električnog automobila.

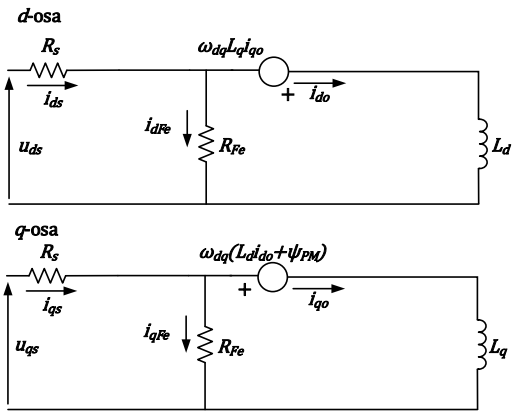
Opšte je poznato u teoriji električnih pogona, da se odgovarajućim izborom vektora struje PMSM, može postići maksimalna efikasnost pogona i stoga i optimalna kontrola PMSM sa stanovišta minimizacije gubitaka pogona [1].

NAPOMENA:
Ovaj rad proistekao je iz master rada, čiji mentor je bio dr Darko Marčetić, red. prof.

U literaturi je predstavljen veliki broj metoda za poboljšanje energetske efikasnosti elektromotornog pogona [1-6]. Ovde su predstavljene dve metode koje poboljšanje energetske efikasnosti postižu na različite načine. Prva metoda obezbeđuje optimalnu kontrolu u ustaljenom stanju, LMC kontrolu (*eng. Loss Model Controller*) [1-3]. Ova metoda bazirana je na matematičkom modelu PMSM. Optimalni vektor struje statora motora se bira na osnovu izračunavanja minimuma gubitaka motora, predstavljanih putem analitičkih jednačina istog matematičkog modela. Druga metoda za poboljšanje energetske efikasnosti i u prelaznim režimima obezbeđuje potpuno iskorišćenje momenta uzimajući u obzir strujne i naponske kapacitete pogona (*eng. Maximum Torque per Ampere/Minimum Ampere per Torque – MTPA/MAPT*) [4-6]. Ove metode obezbeđuju veću efikasnost motora, minimizovanjem statorske struje potrebne da se obezbedi željeni momenat. Utiskivanjem odgovarajućeg vektora struje statora, nastoji se postići minimum gubitaka u bakru, čime se postiže i smanjenje gubitaka tj. povećanje efikasnosti u pogonu.

2. MATEMATIČKI MODEL PMSM SA GUBICIMA U GVOŽĐU

Ovde će biti predstavljen matematički model sinhrone mašine sa stalnim magnetima utisnutim u rotor (*eng. Interior Permanent Magnet Synchronous Machine – IPMSM*). Gubici u gvožđu su gubici u feromagnetnom jezgru mašine, koji se sastoje od gubitaka izazavanih vrtložnim strujama i gubitaka usled histerezisa. Čest prilaz modelovanju efekta gubitaka u gvožđu, jeste dodavanje otpora, paralelnog sa granom magnećenja, R_{Fe}, u ekvivalentnu šemu, kao što je prikazano na Slici 1.



Slika 1. *Ekvivalentna šema IPMSM sa uvaženim gubicima u gvožđu – vektorski model*

Vektorski model se dobija kada se rotirajući dq sistem poravna sa fluksom permanentnog magneta:

$$\theta_{dq} = \theta_{PM}. \tag{1}$$

U skladu sa kolom prikazanim na slici 1 model vektorskog upravljanja u ustaljenom stanju biće zapisan u odgovarajućem obliku, gde su sve veličine i parametri koji figurišu u njemu opisani u [1]:

$$\begin{bmatrix} u_{ds} \\ u_{qs} \end{bmatrix} = R_s \begin{bmatrix} i_{do} \\ i_{qo} \end{bmatrix} + \left(1 + \frac{R_s}{R_{Fe}}\right) \begin{bmatrix} u_{do} \\ u_{qo} \end{bmatrix}$$

$$\begin{bmatrix} u_{do} \\ u_{qo} \end{bmatrix} = \begin{bmatrix} 0 & -\omega_{dq} L_q \\ \omega_{dq} L_d & 0 \end{bmatrix} \begin{bmatrix} i_{do} \\ i_{qo} \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_{dq} \psi_{PM} \end{bmatrix}$$

$$i_{do} = i_{ds} - i_{dFe}, i_{qo} = i_{qs} - i_{qFe} \tag{2}$$

$$i_{dFe} = -\frac{\omega_{dq} L_q i_{qo}}{R_{Fe}}, i_{qFe} = \frac{\omega_{dq}(\psi_{PM} + L_d i_{do})}{R_{Fe}}$$

$$m_{el} = K\left(\psi_{PM} i_{qo} + \left(L_d - L_q\right) i_{do} i_{qo}\right) \rightarrow K = \frac{3}{2} P$$

Ove jednačine će biti iskorišćene u okviru simulacionih analiza za verifikaciju strategija optimalnog upravljanja pogona IPMSM.

2.1. Energetski bilans IPMSM

Ukupni gubici u IPMSM, P_{gub}, se sastoje od gubitaka u bakru statora i rotora P_{cu}, gubitaka u gvožđu P_{Fe} i mehaničkih gubitaka P_m.

$$P_{gub} = P_{cu} + P_{Fe} + P_m \tag{3}$$

- Gubici u bakru mašine predstavljaju gubitke u statorskom namotaju i dati su kao:

$$P_{cu} = R_s\left(i_{ds}^2 + i_{qs}^2\right) \tag{4}$$

- Gubici u gvožđu se sastoje od gubitaka usled histerezisa i vrtložnih struja:

$$P_{Fe} = R_{Fe}\left(i_{dFe}^2 + i_{qFe}^2\right) \tag{5}$$

- Mehanički gubici usled trenja i ventilacije:

$$P_m = k_m \omega \tag{6}$$

Prve dve komponente gubitaka, zavise od električnih veličina IPMSM, te se mogu kontrolisati putem promene vrednosti tih veličina. Sa druge strane, mehanički gubici zavise od brzine, koja je referenca u sistemu regulisanim po brzini i momentu, te se na njih ne može uticati.

Osnovni cilj upravljačkog optimalnog zakona je da minimizuje ove gubitke u cilju poboljšanja stepena iskorišćenja pogona u svim pogonskim režimima.

3. NAPREDNA KONTROLA – UOPŠTENI MODEL

U slučaju IPMSM, može se izvući opšta optimalna relacija koja obezbeđuje poboljšanje performansi unutar pogona. Opšta optimalna relacija podrazumeva orijentaciju vektora struje statora spram pozicije fluksa rotora usled permanentnih magneta na način da se ukupni kontrolabilni gubici (4)–(5) minimizuju. Uvažavanjem celokupnog bilansa posmatra se LMC strategija dok, ukoliko se zanemari komponenta gubitaka u gvožđu (5), tada se podrazumeva MTPA strategija.

Da bi se dobila opšta optimalna relacija za SM pogone, treba poći od skupa jednačina koje opisuju IPMSM vektorski upravljane pogone u ustaljenom stanju sa uključenim gubicima u gvožđu i energetskim bilansom.

Optimizacija karakteristike IPMSM u radnim režimima kada se pogon ne nalazi u sistemskom limitu (strujni, naponski…) se postiže odgovarajućom raspodelom vektora struje statora za ostvarenje zahteva sa mehaničke strane: brzine obrtanja i momenta opterećenja na vratilu. U osnovi, utiskivanjem vektora struje statora se utiče na vrednost statorskog fluksnog obuhvata koji se prilagođava na način da se ostvare pomenuti restrikcioni zahtevi u radnoj tački gde kriterijumska funkcija za optimizaciju ima svoj ekstrem. Formulisanjem optimizacionog kriterijuma u funkciji komponenti vektora struje statora i nalaženjem ekstremnih tačaka mogu se postići optimalne performanse kontrole IPMSM.

Kriterijumska funkcija za optimizaciju predstavlja ukupne kontrolabilne gubitke P_{gub} (4)–(5) izražene u funkciji struje vazdušnog zazora i_{d0} i reference momenta m_{el}. Optimalni uslov se matematički zapisuje na sledeći način:

$$\frac{\partial P_{gub}}{\partial i_{do}} = 0 \rightarrow GH = m_{el}^2 F \tag{7}$$

gde su novodefinisane veličine G, H i F funkcije ugaone učestanosti ω_{dq} i parametara modela:

- $G = K^2\left[R_s R_{Fe}^2 i_{do} + \omega_{dq}^2 L_d (R_s + R_{Fe})(\psi_{PM} + L_d i_{do})\right]$
- $H = \left[\psi_{PM} + (L_d - L_q) i_{do}\right]^3$
- $F = \left[R_s R_{Fe}^2 + \omega_{dq}^2 L_q^2 (R_s + R_{Fe})\right](L_d - L_q)$

Određivanje optimalne radne tačke pogona u kojoj su ukupni kontrolabilni gubici *IPMSM* minimalni svodi se na rešavanje polinoma četvrtog reda po vrednosti reference i_{d0}:

$$f(i_{d0}) = A i_{do}^4 + B i_{do}^3 + C i_{do}^2 + D i_{do} + E = 0 \tag{8}$$

Parametri $A \div E$ definišu koeficijente polinoma (8) i funkcija su parametara ekvivalentne šeme *IPMSM*, ugaone učestanosti ω_{dq} i ulaznog parametra u optimalni blok, reference elektromagnetnog momenta m_{el} (MAPT) ili amplitude vektora struje statora I_s (MTPA).

Vrednosti koeficijenata polinoma (8), za m_{el} kao ulaz u referentni blok, su prikazani u nastavku:

$$A = (L_d - L_q)^3 \gamma$$

$$B = 3\psi_{PM}(L_d - L_q)^2 \gamma + (L_d - L_q)^3 \delta$$

$$C = 3\psi_{PM}^2 (L_d - L_q)\gamma + 3\psi_{PM}(L_d - L_q)^2 \delta$$

$$D = \psi_{PM}^3 \gamma + 3\psi_{PM}^2 (L_d - L_q)\delta$$

$$E = \psi_{PM}^3 \delta - m_{el}^2 \varepsilon$$

uz definicije dodatnih parametara eksplicitno zavisnih od R_{Fe} i ω_{dq}:

- $\gamma = K^2\left[R_s R_{Fe}^2 + \omega_{dq}^2 L_d^2 (R_s + R_{Fe})\right]$
- $\delta = K^2 \omega_{dq}^2 L_d \psi_{PM} (R_s + R_{Fe})$
- $\varepsilon = R_s R_{Fe}^2 (L_d - L_q) + \omega_{dq}^2 L_q^2 (L_d - L_q)(R_s + R_{Fe})$

Rešavanjem polinoma četvrtog reda (8), može se dobiti optimalna referenca fluksa i_{do}. Na osnovu i_{do}, struja koja stvara momenat i_{q0} može se dobiti iz jednačine momenta.

3.1 LMC strategija za IPMSM

Optimalna kontrola zasnovana na LMC pristupu, koristi aproksimativni matematički model pogona i gubitaka u cilju izvođenje zakona optimalnog upravljanja. *Pristupi rešavanju polinoma četvrtog reda (8):*

- Analitički metod korišćenjem Ferarijevog postupka proračuna*: analitika za rešavanje polinoma četvrtog reda* [4].
- *Newton – Raphson metod*: Iterativni numerički postupak za rešavanje, u opštem slučaju, nelinearnih jednačina, predložena u ovom radu.

Ferarijeva metoda je čisto analitička i opšta za polinom četvrtog reda ali je računarski zahtevna, pa nije pogodna za implementaciju u realnom vremenu.

U slučaju rešavanja nelinearnih jednačina, pogodnije je koristi iterativne postupke. *Newton – Raphsonov* iterativni pristup je specijalna vrsta iterativnih metoda koja predstavlja dobar kompromis između preciznosti i složenosti računanja. Omogućava da se izračuna optimalno rešenje u nekoliko koraka. Procedura je sledeća:

- Korak 1: Definisanje početne pretpostavke I_{d0}, $i = 0$
- Korak 2: Izračunavanje $f(I_{d\,0+i})$ i prvog izvoda $f'(I_{d\,0+i})$
- Korak 3: Ponovni proračun ako novo rešenje pogodi $I_{d\,0+i+1} = I_{d\,0} - f(I_{d\,0+i})/f'(I_{d\,0+i})$, $i = i + 1$
- Korak 4: Provera uslova konvergencije $|f(I_{d\,0})| < \varepsilon_f$,

Na osnovu valjanosti uslova u koraku 4:

  - Da – $I_{d\,0+i+1}$ je optimalno rešenje, $i_{do} = I_{d\,0+i+1}$
  - NE – Povratak na korak 2

- Korak 5: Proračun i_{qo}

  $i_{qo} = m_{el}/\{K(\psi_{PM} i_{qo} + (L_d - L_q) i_{do})\}$ za slučaj koji se razmatra a to je da je izlaz brzinskog regulatora momenat

Intenzivni iterativni proračuni mogu biti izostavljeni ako kontroler ne uzme u obzir efekat isturenosti polova unutar IPMSM. Uzimanjem $L_d = L_q$ dobija se:

$$i_{do} = -\frac{\omega^2 L_d \psi_{PM}(R_s + R_{Fe})}{R_s R_{Fe}^2 + \omega^2 L_d^2 (R_s + R_{Fe})} \tag{9}$$

Uzimanjem i_{do} iz prethodne relacije za početno rešenje za Newton – Raphsonov postupak dovodi do bržeg nalaženja optimalnog rešenja.

3.2. MTPA strategija za IPMSM

Zanemarivanjem komponente gubitaka u gvožđu, strategija minimizacije gubitaka u bakru se može koristiti kod IPMSM pogona. Ova metodologija takođe maksimizuje mehanizam za generisanje momenta s obzirom na struju pogona koja je na raspolaganju. Strategije koje se zasnivaju na MTPA/MAPT su manje zavisne od parametara sistema i u nekim slučajevima mogu značajno smanjiti ukupnu računsku složenost. To ih čini glavnim kandidatima za optimalnu kontrolu u baznom opsegu pogona.

Dakle, MAPT strategija je poseban slučaj uopštenog modela (8) kada je $R_{Fe} \to \infty$ i kad se kao ulazni parametar ima momenat m_{el}.

Kao i u slučaju LMC strategije i ovde je moguće izbeći intenzivne proračune ako se smatra da su L_d i L_q bliskih vrednosti i na osnovu te pretpostavke u nastavku će biti prikazan opšti zakon upravljanja za maksimizaciju momenta. Maksimizacija momenta može se ostvariti pomoću MTPA, MAPT i MPTF strategija. Prve dve su već pomenute dok poslednja predstavlja strategiju maksimalnog momenta po fluksu (*eng. Maximum Torque per Flux - MTPF*) i nalazi primenu u oblasti dubokog slabljenja polja gde se efikasno iskorišćavanje raspoloživog napona može dobiti odgovarajućom adaptacijom fluksa.

Opšti optimalni kontrolni blok može se izvesti iz razloga što se slična struktura nalazi tokom procesa istraživanja. Razlika se uočava u koeficijentima skaliranja $\boldsymbol{a}$ i $\boldsymbol{L}$, i karakterističnoj vrednosti struje $\boldsymbol{I}$.

$$x_{opt} = \boldsymbol{L}\frac{\psi_{PM} - \sqrt{\psi_{PM}^2 + 8\boldsymbol{a}(L_d - L_q)^2(\boldsymbol{I})^2}}{4\boldsymbol{a}(L_q - L_d)} \tag{10}$$

Za MTPA metodu x_{opt} je i_{ds}, $\boldsymbol{a}$ i $\boldsymbol{L}$ su 1, dok je $I = I_s$. Za MAPT metodu x_{opt} je i_{ds}, $\boldsymbol{a}$ i $\boldsymbol{L}$ su 3/2 i 1, redom, dok je $I = T/(\psi_{PM} K)$. Za MTPF metodu x_{opt} je λ_{ds}, $\boldsymbol{a}$ i $\boldsymbol{L}$ su 1 i L_q, redom, dok je $I = V/(\omega L_q)$.

4. PARAMETRI POGONA ELEKTRIČNOG VOZILA ZASNOVANOG NA IPMSM

U cilju validacije optimalnih zakona upravljanja zasnovanih na minimizaciji gubitaka pogona kao i na maksimizaciji momenta (LMC i MAPT metode) opisanih u prethodna dva poglavlja, neophodno je upoznati se sa karakteristikama samog pogona korišćenog za potrebe električne vuče.

Električni i mehanički podaci sa natpisne pločice mašine kao i termičke osobine izolacije su prikazane:

$U_{sn} = 48\ V, I_{sn} = 120\ A, f_n = 60\ Hz, P_n = 5.4\ kW$
$n_n = 3500\ o/min, S2 - 60min$

Parametri 5.4kW pogona IPMSM ACX-3434-12 su dobijeni putem implementacije metoda za samopodešavanje pogona IPMSM. Kao ulazni parametri u algoritam su korišćeni podaci sa natpisne pločice i dobijeni su sledeći parametri:

- Otpor statorskog namotaja: $R_s = 3.78\ m\Omega$
- Nazivna induktivnost d ose: $L_d = 86.17\ \mu H$
- Nazivna induktivnost q ose: $L_q = 106.80\ \mu H$
- Nazivni fluks magneta: $\psi_{PM} = 18.5\ mWb$
- Otpornosti gubitaka u gvožđu: $R_{Fe} = 2.35\ \Omega$

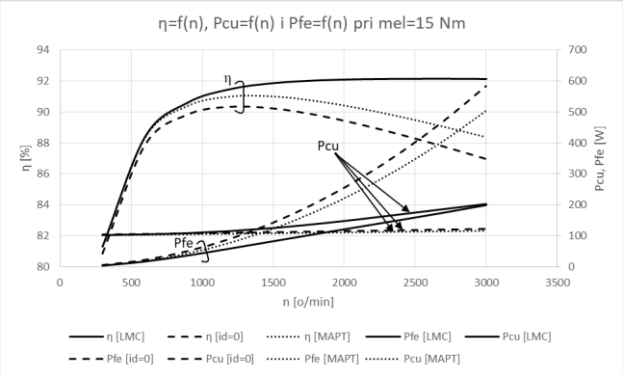
Parametar otpornosti koja modeluje gubitke u gvožđu R_{Fe} ima veliku parametarsku nesigurnost, ako se ne procesuira putem preciznog merenja snage.

Ako se doda i njegova promenljivost sa frekvencijom napajanja, nije moguće očekivati njegovu preciznu estimaciju. Zbog toga je u simulaciji korišćena konstantna vrednost R_{Fe}.

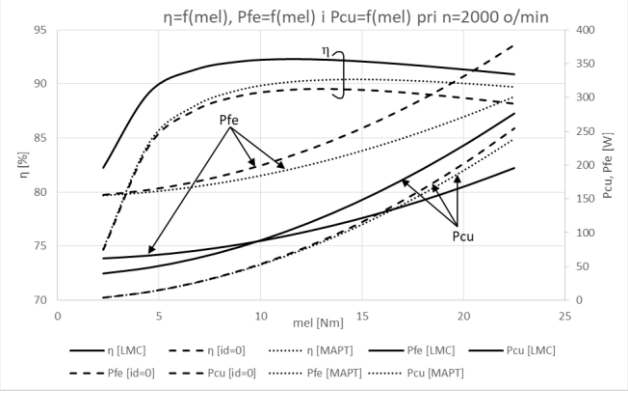
5. SIMULACIONA ANALIZA OPTIMALNOG KONTROLERA NA MODELU IPMSM

Simulacije rada LMC metode zasnovane na uopštenom modelu datom relacijom (8) kao i MAPT strategije (poseban slučaj uopštenog modela (8) kada je $R_{Fe} \to \infty$)

vršene su u Matlab Simulink modelu. Rezultati simulacije dati su na slici 2 i Slici 3.



Slika 2. *Funkcija zavisnosti* $\eta = f(n)$, $P_{cu} = f(n)$ *i* $P_{fe} = f(n)$ *pri momentu opterećenja od 15 Nm*



Slika 3. *Funkcija zavisnosti* $\eta = f(m_{el})$, $P_{cu} = f(m_{el})$ *i* $P_{fe} = f(m_{el})$ *pri brzini obrtanja od 2000 0/min*

Analizom rezultata simulacije, dolazi se do zaključka da se korišćenjem LMC metode postiže najveća efikasnost u svim radnim tačkama. Od svih prikazanih metoda LMC metoda postiže značajno manju P_{Fe}, a ovaj efekat je najvidljiviji i najznačajniji pri velikim brzinama obrtanja kada, za razliku od ostale dve metode, pada efikasnosti i nema. MAPT metod sa minimalnom strujom obezbeđuje potreban momenat i na taj način generišući minimalnu P_{cu} i pri manjim i srednjim brzinama postiže veliku efikasnost. S porastom momenta opterećenja dolazi i do rasta struje pa se minimizacijom P_{cu} postiže velika efikasnost, uporediva sa onom koju postiže LMC metod, što se i vidi na slici 3. MAPT metoda predstavlja odličan izbor u slučaju nepoznavanja parametra R_{Fe} kao i u slučaju kada se zahteva malo procesorsko opterećenje. I LMC metod i MAPT metod problem procesorskog opterećenja mogu rešiti primenom LUT (*eng. Lookup Table - LUT*), s tim što bi za LMC bila potrebna trodimenzionalna (brzina, momenat i struja) LUT, dok je za MAPT dovoljna dvodimenzionalana (momenat i struja).

6. ZAKLJUČAK

LMC metod obezbeđuje maskimalnu efikasnost u čitavom radnom opsegu u slučaju poznavanja parametara IPMSM. Problem sa procesorskim opterećenjem može se rešiti primenom LUT. MAPT strategija je manje zavisna od parametara i u nekim slučajevima može značajno smanjiti ukupnu računsku složenost MAPT metod obezbeđuje odličan dinamički odziv i zbog manjeg procesorkog opterećenja može se koristiti i bez primene LUT. Naročito je koristan za veće momente opterećenja gde ima veliku efikasnost i kada je R_{Fe} nepoznat i primena LMC metode onemogućena.

7. LITERATURA

[1] Morimoto S., Tong Y., Takeda Y., Hirasa T., "*Loss Minimization Control of Permanent Magnet Synchronous Motor Drives* ", IEEE Transactions on Industrial Electronics, vol. 41, no. 5, pp. 511 – 517, 1994.

[2] Mademlis C., Kioskeridis I., Margaris N., "*Optimal Efficiency Control Strategy for Interior Permanent – Magnet Synchronous Motor Drives*", IEEE Transactions on Energy Conversion, vol. 19, no. 4, pp. 715 – 723, 2004.

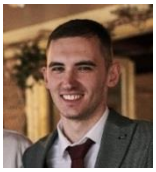
[3] Lee J., Nam K., Choi S., Kwon S., "*Loss-Minimizing Control of PMSM With the Use of Polynomial Approximations* ", IEEE Transactions on Power Electronics, vol. 24, no 4, pp. 1071–1082, 2009.

[4] Jung S. Y., Hong J., Nam K., " *Current Minimizing Torque Control of IPMSM using Ferrari's Method* ", IEEE Transactions on Power Electronics, vol. 28, no. 12, pp. 5603 – 5617, 2013.

[5] Pan C. T., Sue S. M., "*A Linear Maximum Torque Per Ampere Control for IPMSM Drives Over Full – Speed Range* ", IEEE Transactions on Energy Conversion, vol. 20, no. 2, pp. 359 – 366, 2005.

[6] Gallegos – Lopez G., Gunawan F. S., Walters J. E., "*Optimum Torque Control of Permanent – Magnet AC Machines in the Field – Weakened Region* ", IEEE Transactions on Industry Applications, vol. 41, no. 4, pp. 1020-1028, 2005.

Kratka biografija:



Stefan Simić rođen je u Kruševcu 1994. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Energetska elektronika i električne mašine odbranio je 2021.god.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.42
DOI: https://doi.org/10.24867/16BE28Kaljevic

TESTNO OKRUŽENJE ZA GRAFIČKI PRIKAZ GEOPROSTORNIH PODATAKA NA OSNOVU SPARQL UPITA

TEST ENVIRONMENT FOR GRAPHICAL REPRESENTATION OF GEOSPATIAL DATA BASED ON SPARQL QUERIES

Predrag Kaljević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj - *U ovom radu prikazan je dizajn i implementacija proširenja SPARQL Playground aplikacije u cilju testiranja mogućnosti pretrage i grafičkog prikaza geoprostornih podataka. Aplikacija je proširena podrškom za izvršavanje GeoSPARQL upita i mogućnošću grafičkog prikaza geoprostornih podataka izraženih u WKT literal formatu.*

Ključne reči: *RDF, SPARQL, GeoSPARQL, Geospatial, Query*

Abstract - *This paper presents the design and implementation of the SPARQL Playground application extension with the goal of testing the capabilities of querying and graphical representation of geospatial data. The application is extended with support for GeoSPARQL querying and the ability to graphically represent geospatial data presented in WKT literal format.*

Keywords: *RDF, SPARQL, GeoSPARQL, Geospatial, Query*

1. UVOD

SPARQL predstavlja protokol i upitni jezik, namenjen za vršenje upita nad RDF podacima. *SPARQL Playground* predstavlja višenamensku *standalone web* aplikaciju dizajniranu i implementiranu sa ciljem učenja, predavanja i testiranja ovog upitnog jezika. Omogućava korisniku izvršavanje i modifikovanje predefinisanih upita nad testnim podacima, kao i definisanje novih upita. Oslanja se na *OpenRDF Sesame 2.8.6* kao SPARQL *engine*.

Ideja iza ovog rada je migracija aplikacije da se oslanja na novi SPARQL *engine*, RDF4J. RDF4J je nastao iz *Sesame* i predstavlja njegovo proširenje. Razlog migraciji je podrška za *GeoSPARQL*, koji *Sesame* ne podržava. Nakon migracije, aplikacija je proširena interaktivnom mapom (uz pomoć *LeafletJS* biblioteke), kao i sa mogućnostima prikaza geoprostornih podataka i vršenja *GeoSPARQL* upita nad njima. Na kraju su pripremljeni geoprostorni podaci za testiranje performansi novih funkcionalnosti aplikacije pri radu sa velikom količinom grafičkih elemenata.

Aplikacija je razvijena u *Java Spring Boot* i *AngularJS*. Kao razvojno okruženje, korišćen je *Microsoft Visual Studio Code*.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Milan Gavrić, docent.

2. TEORIJSKE OSNOVE

2.1. RDF

Resource Description Framework (RDF) predstavlja standardni model za razmenu podataka na *Web*-u. To je apstraktni model podataka koji organizuje elemente podataka i standardizuje kako su oni povezani međusobno. RDF model podataka služi za predstavljanje i razmenu podataka u vidu trojki, koji se sastoje od subjekta, predikata i objekta [1]. U okviru jedne trojke, subjekat predstavlja resurs definisan preko *IRI*-a ili praznog čvora, predikat je veza između subjekta i objekta, a objekat je neka osobina subjekta koja može biti predstavljena kao literal ili resurs.

2.2. CIM

Elektroenergetske kompanije koriste veliki broj različitih formata za skladištenje svojih podataka. Iako se većina ovih podataka koristi isključivo unutar kompanije, često se javlja potreba za njihovom razmenom, kako između različitih aplikacija unutar kompanije, tako i sa drugim kompanijama.

Common Information Model (CIM) je nastao usled potrebe informacionih sistema (koji učestvuju u upravljanju elektroenergetskim sistemima) za postojanjem opšteprihvaćenog standarda modelovanja i predstavljanja elektroenergetskih mreža. Pre nastanka CIM-a, razmena podataka između aplikacija je često zahtevala njihovu konverziju u model podataka koji je pogodan za obradu nekoj određenoj aplikaciji. Kako bi se eliminisala potreba za različitim konverzijama podataka prilikom razmene, nastao je CIM kao apstraktni model, nezavisan od platforme, za definisanje elemenata elektroenergetskog sistema.

2.3. SPARQL

SPARQL Protocol and RDF Query Language (SPARQL) je protokol i RDF upitni jezik. Dizajniran je za vršenje upita nad RDF podacima, ali nije time ograničen. Dostupni su načini za tretiranje relacionih podataka, XML, JSON i drugih formata kao RDF, tako da se mogu vršiti SPARQL upiti nad podacima u ovim formatima – ili nad kombinacijom izvora, što je jedan od najmoćnijih aspekata SPARQL/RDF kombinacije [2].

Osnova SPARQL-a sastoji se iz upita u formi jednostavnih graf obrazaca. Takođe omogućava i niz funkcija za konstruisanje naprednih obrazaca upita, za navođenje dodatnih uslova za filtriranje i za formatiranje

prikaza konačnog rezultata. Ključne reči koje se najčešće javljaju u SPARQL upitu su: *PREFIX* (definiše namespace), *SELECT* (definiše format prikazanog rezultata) i *WHERE* (predstavlja uslove koji moraju biti zadovoljeni).

2.3.1. GeoSPARQL

Geoprostorni podaci su sve više dostupni na *web*-u u obliku skupova podataka opisanih uz pomoć RDF-a. Iako je SPARQL pogodan za upite nad vezama koje su eksplicitno predstavljene u podacima, nad implicitnim vezama, kao što su geoprostorne, ne mogu se tako lako vršiti upiti. Na primer, mogu postojati skupovi podataka koji opisuju spomenike i parkove, ali nije lako izvodljivo spojiti ta dva skupa na osnovu njihove nedefinisane povezanosti [3].

Sposobnost davanja odgovora na smislen upit, kao što je "Koji parkovi su manje od 3 kilometra udaljeni od spomenika Svetozaru Miletiću u Novom Sadu?", zavisi od toga kako su podaci predstavljeni, da li su svi resursi povezani sa spomenikom Svetozaru Miletiću, kao i da li je ta veza eksplicitno navedena.

GeoSPARQL je OGC (*Open Geospatial Consortium*) standard za predstavljanje i vršenje upita nad geoprostornim podacima u okviru semantičkog web-a. Definiše rečnik za predstavu geoprostornih podataka u RDF-u, kao i proširenje SPARQL upitnog jezika za njihovu obradu [4].

3. TEHNOLOGIJE

3.6. RDF4J

Eclipse RDF4J [5] predstavlja *open source* Java *framework* namenjen radu sa RDF podacima. Obuhvata parsiranje, skladištenje, rasuđivanje i izvršavanje upita nad RDF podacima. Obezbeđuje API koji se može povezati sa svim vodećim RDF skladištima. Dozvoljava povezivanje sa SPARQL *endpoint*-ima i kreiranje aplikacija koje se oslanjaju na *linked data* i semantički *web*.

RDF4J u potpunosti podržava SPARQL 1.1 upitni jezik za pisanje upita i nudi transparentni pristup udaljenim RDF repozitorijumima uz upotrebu istog API-ja kao za lokalni pristup. Takođe podržava najpoznatije formate RDF fajlova, uključujući RDF/XML, Turtle, N-Triples, N-Quads, JSON-LD, TriG i TriX.

Ono što je najvažnije pomenuti jeste da RDF4J nudi proširenu algebru za delimičnu podršku *GeoSPARQL*-a. Ovo podrazumeva dodatne geoprostorne funkcionalnosti kao deo SPARQL *engine*-a, na bilo kom RDF4J repozitorijumu, uz upotrebu *Spatial4J* i JTS biblioteka za geoprostorno zaključivanje. RDF4J podržava *GeoSPARQL* funkcije samo nad geoprostornim podacima predstavljenim u obliku WKT *string*-ova.

3.7. LeafletJS

LeafletJS je open *source JavaScript* biblioteka namenjena za implementaciju interaktivnih mapa u aplikacijama. Inicijalno je razvijena 2011. godine i podržava većinu mobilnih i desktop platformi. Izuzetno je kompaktna, svega 39KB JS koda, i sadrži funkcionalnosti mapiranja koje su od interesa većini developera.

Dizajniran je da podrži tri osnovna zahteva: jednostavnost, performanse i upotrebljivost. Može biti proširen sa velikim brojem *plugin*-a, a takođe je i detaljno pokriven dokumentacijom.

Podržava iscrtavanje tačaka, linija, poligona i drugih geometrijskih oblika, zbog čega je pogodan za potrebe ovog rada, a to su iscrtavanje i prikazivanje geoprostornih podataka učitanih preko *GeoSPARQL* upita.

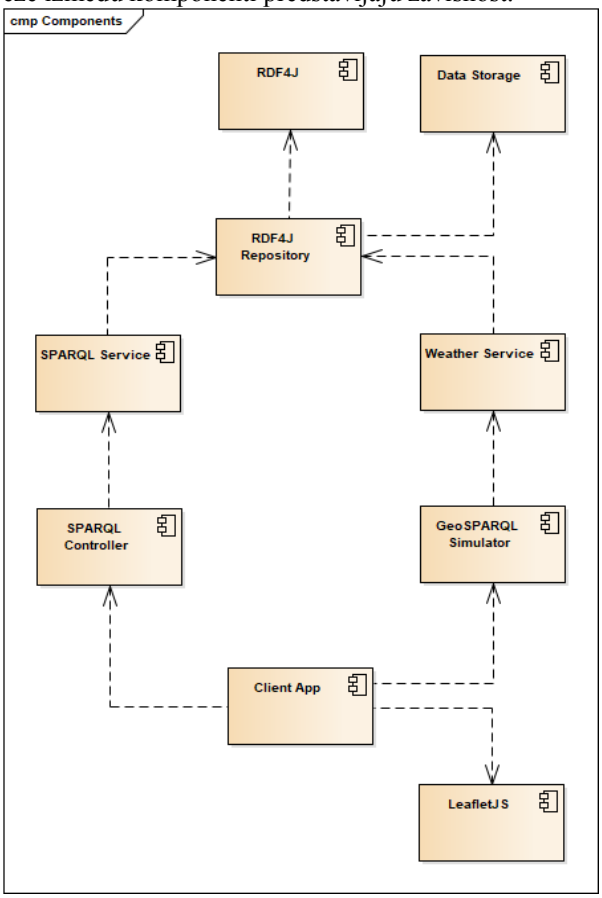
4. DIZAJN REŠENJA

4.1. Opis postojećeg rešenja – SPARQL Playground

SPARQL Playground aplikacija je osnova ovog rada na koju se nadograđuje. Predstavlja višenamensku *standalone web* aplikaciju namenjenu za učenje, predavanje i testiranje SPARQL upitnog jezika. Implementirana je u *Java Spring Boot* i *AngularJS*. Razvijena je na *SIB Swiss Institute of Bioinformatics*. Koristi *OpenRDF Sesame 2.8.6 framework* kao SPARQL *engine*.

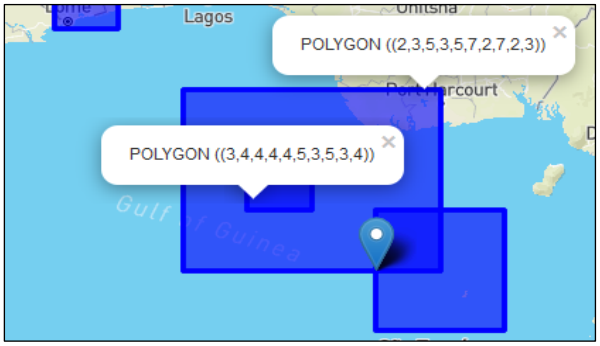
4.2. Dizajn proširenog rešenja

Proširena *SPARQL Playground* aplikacija se oslanja na RDF4J biblioteku kao SPARQL *engine* i koristi *LeafletJS* biblioteku za prikaz geoprostornih podataka. Dodatno, integriše se sa servisom vremenskih prilika [6] na osnovu čega je implementirana simulacija kretanja vremenskih nepogoda na mapi. Dijagram komponenti koji oslikava arhitekturu aplikacije je prikazan na ***Slika 1***. Usmerene veze između komponenti predstavljaju zavisnost.



Slika 1: *Dijagram komponenti proširene aplikacije*

Na Slika 2 je prikazan primer poligona iscrtanih na osnovu WKT literala dobijenih iz rezultata SPARQL upita:



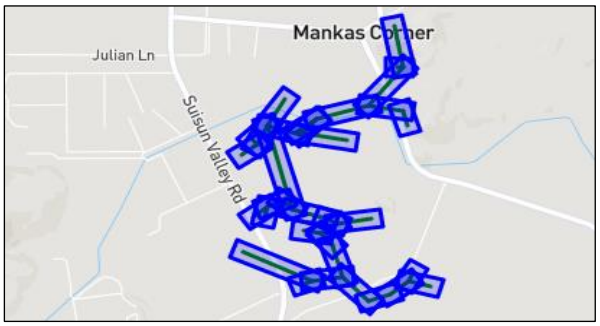
Slika 2: *Primer iscrtanih poligona*

Korisnik pored standardnih SPARQL upita, može da vrši *GeoSPARQL* upite nad geoprostornim podacima u obliku WKT literala. *OpenRDF Sesame 2.8.6 framework*, na koji se inicijalno oslanjao *SPARQL Playground*, ne podržava *GeoSPARQL* upite. Iz tog razloga je bilo neophodno migrirati aplikaciju da se oslanja na RDF4J *framework*, koji ga nasleđuje i proširuje GeoSPARQL podrškom.

4.2.1. Grafički prikaz izvoda

Podržano je iscrtavanje tri vrste geometrijskih oblika: tačaka, linija i poligona. Učitavanje CIMXML datoteka, gde svaka sadrži koordinate svih *ACLineSegment*-a koji pripadaju tom izvodu, omogućava da se taj izvod grafički prikaže na mapi. Svaki *ACLineSegment* sadrži referencu na svoj *Location*, a *Location* sadrži jedan ili više *PositionPoint*-a koji mu pripadaju.

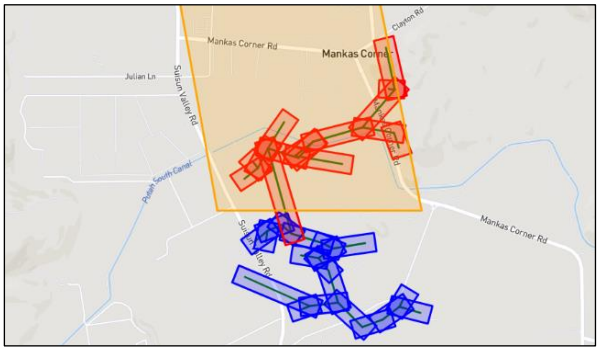
Slika 3 prikazuje primer iscrtanog izvoda zajedno sa poligonima koji predstavljaju njegovu zonu osetljivosti upotrebljenu prilikom rada servisa vremenskih prilika [6]:



Slika 3: *Primer iscrtanog izvoda sa svojom zonom osetljivosti*

4.2.2. Servis vremenskih prilika

Primer rada servisa vremenskih prilika (*Weather service*) [6] prikazan je na Slika 4:



Slika 4: *Simulacija kretanja oluje preko zone osetljivosti*

Predstavlja servis u sklopu proširenja *SPARQL Playground*-a namenjen za simulaciju kretanja vremenskih nepogoda na mapi. Delovi terena koji su pogodile vremenske nepogode su predstavljeni kao poligoni. Ukoliko prilikom njihovog kretanja dođe do preklapanja sa nekom od zona osetljivosti izvoda, ta zona menja boju kao znak upozorenja.

5. IMPLEMENTACIJA

5.1. Migracija aplikacije na RDF4J *framework*

OpenRDF Sesame 2.8.6 framework, na koji se inicijalno oslanjao *SPARQL Playground*, ne podržava *GeoSPARQL*. *Eclipse RDF4J* ga nasleđuje i proširuje sa algebrom za delimičnu podršku *GeoSPARQL* upita. Ovo je bio razlog migracije aplikacije. Proširenje podrazumeva dodatne geoprostorne funkcionalnosti kao deo SPARQL *engine*-a, na bilo kom RDF4J repozitorijumu, uz upotrebu *Spatial4J* i JTS biblioteka za geoprostorno zaključivanje. RDF4J podržava *GeoSPARQL* funkcije samo nad geoprostornim podacima izraženim kao WKT literali.

Prvi korak migracije bio je uklanjanje OpenRDF Sesame zavisnosti (dependency) i njihova zamena novim RDF4J unutar pom.xml datoteke. Ceo *Sesame core framework* je bio uključen kao *sesame-runtime*. *Sesame-sail-nativerdf* zavisnost predstavlja SAIL implementaciju koja čuva podatke direktno kao datoteke na disku. SAIL (*Storage And Inference Layer*) je interfejs za skladištenje RDF podataka, preko kojih može da izvršava upite. Poslednja uklonjena zavisnost je *sesame-runtime-osgi*, koja obuhvata OSGi (*Open Service Gateway Initiative*) *runtime* zavisnosti za *OpenRDF* aplikaciju.

Nakon toga, zamenjeni su odgovarajućim RDF4J zavisnostima. Pre svega je dodat BOM (*Bill Of Materials*) koji služi da otkloni problem nepodudaranja verzija raznih uključenih zavisnosti. Aplikacija je zasnovana na 3.6.2 verziji RDF4J. Zavisnost *rdf4j-storage* obuhvata sve RDF4J biblioteke vezane za skladištenje i upite/-zaključivanje. *Jackson API* je uključen kroz *jackson-core* (2.12.2) zavisnost i neophodan je za parsiranje i rad sa JSON podacima. Poslednja izmena u *pom.xml* datoteci je dodavanje *SLF4J API*-ja za logovanje u aplikaciji.

5.2. Grafički prikaz geoprostornih podataka

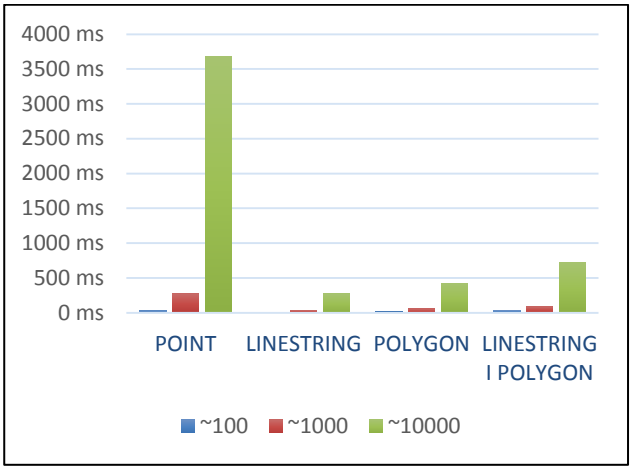
LeafletJS mapa se inicijalizuje prilikom pokretanja aplikacije. Geoprostorni podaci se prikazuju nakon izvršavanja *GeoSPARQL* upita ili nakon prelaska na stranicu *GeoSPARQL* Simulatora, gde se automatski učitavaju i prikazuju svi izvodi.
Podržano je prepoznavanje i obrada 4 tipa WKT literala:

1. ***POINT EMPTY*** – Prazan element, nema grafičku reprezentaciju.
2. ***POINT*** – Predstavlja se kao jedna tačka na mapi.
3. ***LINESTRING*** – Predstavljen kao linija sa početnom i krajnjom tačkom. Izvodi se prikazuju uz pomoć linija, gde svaka od njih predstavlja jedan njegov *ACLineSegment*.
4. ***POLYGON*** – Predstavljen kao niz tačaka gde su prva i poslednja jednake. Zone osetljivosti oko izvoda su prikazane poligonima, kao i vremenske nepogode u sklopu servisa vremenskih prilika.

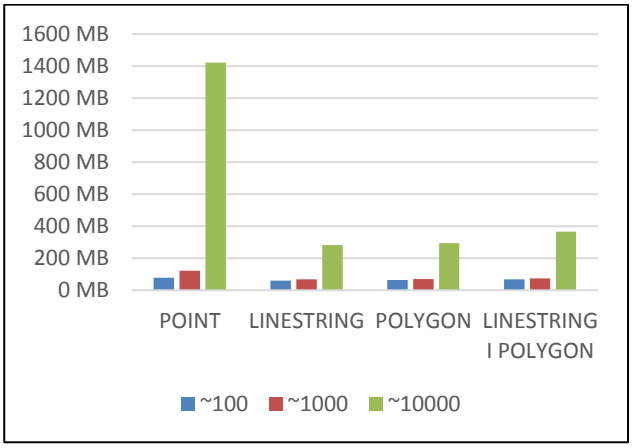
6. IZMERENE PERFORMANSE

Performanse su merene na računaru sa sledećim karakteristikama: procesor *Intel Core i5-4460 3.20GHz*, 16GB RAM memorije, HDD sa kapacitetom od 1TB i *Microsoft Windows 10* operativni sistem. Testni podaci korišćeni prilikom merenja su bazirani na "*IEEE 37 Node Test Feeder*" [7] predstavljenim u CIMXML obliku.

Grafikon 1 prikazuje rezultate merenja performansi aplikacije prilikom iscrtavanja elemenata na mapi, a Grafikon 2 zauzeće memorije. Posmatrano je četiri slučaja: Iscrtavanje tačaka, linija, poligona i linija/poligona zajedno. Poslednji slučaj oslikava rad aplikacije prilikom iscrtavanja izvoda sa njihovim zonama osetljivosti. Svakom od ovih slučajeva je izmereno vreme iscrtavanja i zauzeće memorije 100, 1000 i 10000 elemenata, s tim što je broj elemenata dvostruko veći u slučaju linija i poligona zajedno.



Grafikon 1: *Poređenje performansi iscrtavanja elemenata*



Grafikon 2: *Poređenje zauzeća memorije*

Uočava se da su tačke daleko najslabije i po pitanju vremena iscrtavanja i po zauzeću memorije. Razlog ovome je što *LeafletJS* biblioteka tačke predstavlja *marker* elementima, koji su memorijski daleko zahtevniji od linija i poligona. Linije pokazuju najbolje performanse, a poligoni nešto slabije od njih.

7. ZAKLJUČAK

Migracija *SPARQL Playground*-a na RDF4J *framework* znatno proširuje potencijal ove aplikacije. Otvorena je mogućnost rada sa geoprostornih podacima uz pomoć *GeoSPARQL* upita, preko kojih se mogu izvesti zaključci o relacijama između geometrijskih entiteta. Integracija *LeafletJS* biblioteke dodatno proširuje aplikaciju podrškom prikaza geoprostornih podataka uz pomoć prostih elemenata na mapi.

Aplikacija se generalno pokazala dobro u slučajevima iscrtavanja do 10000 elemenata na mapi istovremeno, osim u slučaju prikaza 10000 tačaka. Potencijalno rešenje ovog problema bilo bi implementacija drugog načina za predstavljanje tačaka. Primer bi bio poligon ili kružnica manjih dimenzija. Drugo potencijalno proširenje može biti omogućavanje iscrtavanja kompleksnijih geometrijskih entiteta u WKT literal obliku. Treće i najveće proširenje može biti podrška selekcije 2 elementa na mapi, nakon čega se prikazuje meni sa skraćenicama za *GeoSPARQL* upite. Na taj način bi se za svaka dva selektovana elementa mogle dobiti informacije o njihovim međusobnim relacijama.

8. LITERATURA

[1] JoshData, „https://github.com," 2014. [Na mreži]. Available: https://github.com/JoshData/rdfabout/blob/gh-pages/intro-to-rdf.md. [Poslednji pristup 9 Jun 2021].

[2] B. DuCharme, Learning SPARQL, drugo ur., O'Reilly Media, 2013.

[3] R. Battle i D. Kolas, GeoSPARQL: Enabling a Geospatial Semantic Web, Knowledge Engineering Group, Raytheon BBN Technologies.

[4] OGC GeoSPARQL - A Geographic Query Language for RDF Data, Open Geospatial Consortium, 2012.

[5] „https://rdf4j.org," [Na mreži]. Available: https://rdf4j.org/about/. [Poslednji pristup 14 Jun 2021].

[6] S. Stojaković, Testno okruženje za rukovanje meteorološkim podacima pomoću SPARQL upita, Novi Sad, 2021.

[7] Distribution System Analysis Subcommittee, *IEEE 37 Node Test Feeder*, IEEE Power Engineering Society.

Kratka biografija:

Predrag Kaljević rođen je 1994. godine u Novom Sadu. Završio je osnovnu školu „Jovan Jovanović Zmaj" u Sremskoj Kamenici, 2009. godine. Završio je srednju ekonomsku školu „Svetozar Miletić" u Novom Sadu, 2013. godine. Iste godine, u septembru, upisao je osnovne akademske studije na Fakultetu tehničkih nauka, smer softversko inženjerstvo i informacione tehnologije. Završio je osnovne studije 2019. godine i upisao master studije iste godine, smer primenjeno softversko inženjerstvo. Ispunio je sve obaveze i položio sve ispite predviđene studijskim programom.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.42
DOI: https://doi.org/10.24867/16BE29Polih

DUGOROČNA PROGNOZA POTROŠNJE ELEKTRIČNE ENERGIJE ZASNOVANA NA LINEARNOM MODELU

LONG-TERM LOAD FORECAST BASED ON A LINEAR MODEL

Kristina Polih, *Fakultet tehničkih nauka, Novi Sad*

Oblast – Elektrotehničko i računarsko inženjerstvo
Kratak sadržaj – *Rad se bavi problemom dugoročne prognoze potrošnje električne energije. Cilj rada jeste da se upotrebi jednostavan model i pokaže korelacija temperature vazduha sa potrošnjom električne energije. Korišteni su SVM i MLR algoritmi, koji rade prognozu potrošnje na osnovu linearnog modela, dok su za određivanje tačnosti upotrebljeni kvantili, MAPE, MSE, PLF i Winkler funkcija.*
Ključne reči: *Dugoročna prognoza potrošnje električne energije, regresioni algoritmi, kvantili, SVM, MLR*
Abstract – *The paper deals with the problem of long - term forecast of electricity consumption. The aim of this paper is to use a simple model and show the correlation of air temperature with electricity consumption. SVM and MLR algorithms were used, which perform consumption forecasting based on a linear model, while quantiles, MAPE, MSE, PLF and Winkler functions were used to determine accuracy.*
Keywords: *Long-term load forecast, Regression algorithms, quantiles, MLR, SVM*

1. UVOD

Prognoza potrošnje električne energije, omogućava da se preciznije mogu odrediti buduće akcije koje su potrebne za normalan rad elektroenergetskog sistema. Efikasnije se može rukovati sistemom, kao i kupovinom i prodajom energije na berzi, ako se zna koliko će biti potrebno energije za napajanje svih korisnika.

Prilikom proračuna prognoze potrošnje, mora se voditi računa o parametrima koji nisu statični i koji uvode neizvesnost u same proračune, kao što su atmosferske prilike tj. temperatura vazduha, kiša, brzina vetra, vlažnost vazduha itd.

Dinamika promenljivih parametara čini dugoročnu prognozu zahtevnijom, pa i težom, a samim tim i izazovnijom za istraživanje.

Ovde se posmatra dugoročna prognoza potrošnje na dnevnom nivou za narednih godinu dana. Pretpostavljeno je da predviđanje prognoze potrošnje može biti poboljšano pretragom po sličnim danima (eng. Similar Day search). U radu su upoređeni MLR i SVM algoritmi, nad istim skupom podataka.

Algoritmi su obučavani i testirani na jednostavnom linearnom modelu sa ciljem da se pokaže korelacija između temperature vazduha i potrošnje električne energije.

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

2. DUGOROČNA PROGNOZA POTROŠNJE ELEKTRIČNE ENERGIJE

Prognoza potrošnje električne energije se može podeliti u četiri kategorije: prognoza bliska realnom vremenu – horizont predikcije od nekoliko sati, kratkoročna prognoza potrošnje električne energije (STLF) – obično predstavlja kratak interval od par dana do dve ili tri nedelje, srednjeročna prognoza potrošnje električne energije (MTLF) – često je definisana u periodu od jednog meseca pa do jedne godine i dugoročna prognoza potrošnje električne energije (LTLF) – predstavlja prognozu koja se izvodi od jedne godine pa čak i do trideset godina.

Za različite vrste prognoza potrošnje, mogu da variraju faktori koji će biti prioritetni. Tačnost prognoza takođe varira u odnosu na to koja prognoza je u pitanju.

LTLF omogućava da se uoči kada će biti vrhunac potrošnje u posmatranom periodu, i da se u skladu sa prognoziranom potrošnjom pripremi sistem. Vremenski period za koji se radi dugoročna potrošnja, može biti od jedne godine pa čak do nekoliko decenija. Dugoročne prognoze od nekoliko decenija se koriste za planiranja i proširivanja od strane preduzeća koja se bave proizvodnjom, prenosom, distribucijom električne enrgije, a najčešće se koriste za uređenje i razvijanje elektroenergetskih sistema. Na osnovu radova koji su napisani na ovu temu, izbrani su MLR i SVM algoritmi, radi njihovog poređenja [1].

Potrošnja zavisi od mnogobrojnih faktora, koji mogu da variraju iz dana u dan, pa samim tim i prilikom proračuna prognoze potrošnje se moraju uzeti u obzir svi važni elementi koji utiču na potrošnju. [6] Jedan od bitnijih faktora su vremenske prilike (temperatura i vlažnost vazduha, vetar, osunčanost i sl.), od čega zavisi da li će korisnici uključiti grejalice, klime, itd. Ako je jako visoka ili niska temperatura, veliki deo stanovništa će uključiti belu tehniku za grejanje ili rashlađivanje prostora ili stanova. Stoga, predviđanje potrošnje električne energije zavisi od vremenske prognoze i naravno temperature.

3. DETERMINISTIČKI ALGORITMI

Deterministički pristup je korišten u cilju izbora optimalnih parametara, npr. koliko je dana potrebno da se iskoristi za obučavanje modela da bi se dobili dovoljno tačni rezultati, itd. Zajednički parametri za sve varijacije modela su: broj dana nad kojim se radi pretraga po sličnom danu, granica temperature i broj dana koji su ulaz u model. Prilikom odabira optimalnih parametara, ulazni modeli se razlikuju, po broju dana nad kojima se radi pretraga po sličnom danu, po graničnim vrednostima temperature, koja je parametar za pretragu po sličnim

danima i po broju dana koji su ulaz za obučavanje modela.

3.1. Višestruka linearna regresija (MLR)

MLR pokušava da uspostavi odnos između dve ili više nezavisnih promenljivih i jedne zavisne promenljive. Iako odnos između zavisne i nezavisnih promenljivih može biti nelinearan u ovom radu je MLR bazirana nad sledećim pretpostavkama [8]: 1. linearna zavisnost između zavisne i nezavisne promenljive, 2. nezavisne promenljive nisu u međusobnoj korelaciji, 3. varijacija ostataka je konstantna, 4. nezavisnost posmatranja, 5. multivarijantna normalnost.

Prema tome, vrednost predviđanja *y* je

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \varepsilon$$

gde su: β_0 – presek tj. vrednost predviđanja kada su svi ulazi nula, $\beta_1, \beta_2, \ldots, \beta_n$ – koeficijenti regresije, $X_1, X_2, \ldots, X_n$ – ulazi kao nezavisne promenljive, ε – greška predviđanja.

3.2. Support Vector Machine (SVM)

SVM su nadzirani obučavajući modeli koji su povezani sa određenim algoritmima za potrebe analiziranja podatka potrebnih za klasifikaciju i regresione analize. Problem linearnog regresionog SVM algoritma jeste da pronađe funkciju koja radi aproksimaciju preslikavanja ulaznih podataka u realne brojeve, na osnovu obučavajućih uzoraka [8].

3.3. Algoritmi za procenu tačnosti

Najvažniji deo, prilikom upotrebe svakog algoritma je određivanje koliko je tačan. U zavisnosti od toga koja se metrika koristi za procenu tačnosti, mogu se dobiti bolji i lošiji rezulatati [10].

U radu će biti korišteni: Mean Absolute Percentage Error (MAPE) i Mean Square Error (MSE). Iako izgleda da je MAPE veoma jednostavan za upotrebu, postoji mnogi nedostaci u praktičnoj primeni, takođe postoje i mnoga istraživanja o nedostacima i o obmanjujućim rezultatima MAPE algoritma [11].

4. PROBABILISTIČKI ALGORITMI

Algoritam A se ponaša kao matematička funkcija preslikavanja tj. mapiranja i to na takav način da ako primenimo A na isti ulaz X, nekoliko puta, uvek ćemo dobiti isti izlaz Y. Matematički zapis takve funkcije bi bio: A : X → Y, gde A predstavlja deterministički algoritam sa ulazima X i izlazima Y. U odnosu na deterministički algoritam, probabilistički algoritam A je takav da njegovo ponašanje, u najvećem broju slučajeva, kontrolišu slučajni događaji. Prilikom proračuna izlaza Y, koristeći ulaz X, rezultat algoritma zavisi od ograničenog broja nasumičnih eksperimenata.

Konkretno, primenom algoritma A na isti ulaz X, više puta, mogu se dobiti potpuno različiti rezultati [16-17]. Algoritmi su u prvoj iteraciji korišteni kao deterministički, radi utvrđivanja optimalnog izbora parametara. Nakon čega su modifikovani po pravilima koja nalažu probabilistički algoritmi, tako da koriste slučajan parametar koji algoritmima obezbeđuje nasumičnost.

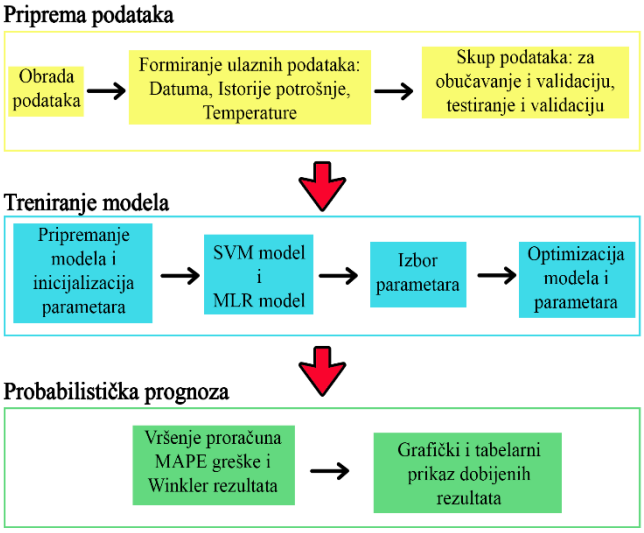
4.1. Algoritmi za procenu tačnosti

Funkcije gubitaka su veoma važne i u teoriji i u praksi predviđanja. Prilikom određivanja tačnosti probabilističkih algoritama često se koriste funkcije gubitaka u kombinaciji sa kvantilima [17]. Kvantili predstavljaju presečne tačke koje odvajaju opsege raspodele verovatnoće na neprekidne intervale sa jednakim verovatnoćama. Za probabilističke rezulatate su korišteni: Pinball Loss Function (PLF) i Winkler Function. Pošto Winkler i PLF računaju vrednosti za jedan i-ti vremenski trenutak, funkcije bodovanja su korištene da bi se napravio prosek vrednosti: Quantile Score (QS) i Winkler Score (WS).

5. IMPLEMENTACIJA

U radu su korišteni linearni regresioni SVM algoritam i MLR algoritam. Iskorišteni su implementirani algoritmi koji se nalaze u Accord.NET biblioteci. SVM algoritam koristi pojednostavljenu verziju Chih-Jen Lin i Jorge Moré's TRON, Newton-ove metode za rešavanje velikih optimizaciono-ograničenih problema [1].

Upotreba algoritama i implementacija se grubo može podeliti na tri etape: priprema podataka, treniranje modela i prikazivanje rezultata probabilističke prognoze. Proces probabilističke prognoze je prikazan na slici 5.1.



Slika 5.1. *Tok implementacije probabilističke prognoze potrošnje*

Korišteni podaci su u periodu od 2007. do 2015. godine, na dnevnom nivou, na nivou Srbije. Potrošnja električne energije je izražena u MW jedinicama, a pribavljena je sa sajta Entsoe za potreban period od 2007 do 2015. Izvor podataka koji je objavio ENTSO-E su OPS-ovi članovi [19]. Potrošnja je bila predstavljena u satima za svaki dan, pa je za potrebe ovog rada urađen prosek na dnevnom nivou. Korištena je prosečna temperatura na nivou Srbije za posmatrani period, za svaki dan. Temperatura je pribavljena sa POWER Data Access sajta [20].

Skup podataka pomoću kojeg je određen optimalan deterministički pristup, koji je osnova za probabilistički pristup i sa kojim su rađena dalja testiranja je organizovan u formatu Datum, Potrošnja i Temperatura.

Varijacije determinističkih algoritama koji su korišteni u radu se razlikuju po broju dana koje algoritam koristi za obučavanje i prognozu potrošnje. Sve varijacije algoritama koriste iste parametre i iste podatke za obučavanje. Parametri koji se koriste su prethodna potrošnja električne energije i srednja temperatura vazduha, kako bi se dokazala korelacija između potrošnje i temperature vazduha. Nakon odabira najtačnije varijacije, algoritmi su pobolj-

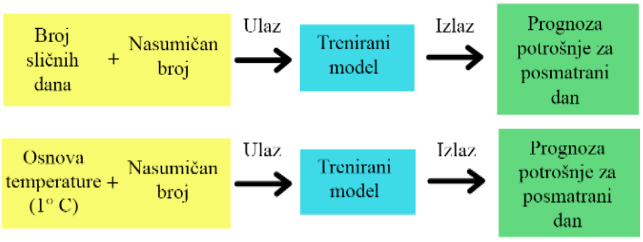
šani tako da koriste Similar Day pretragu (pretraga po sličnim danima), odnosno da pomoćni algoritam pretražuje određeni broj dana, kako bi našao vrednosti koje upadaju u postavljeni okvir.
Pretraga po sličnim danima je testirana na MLR algoritmu. U cilju poboljšanja tačnosti algoritma prilikom pretrage sličnih dana, kada se posmatra parametar temperatura, napravljen je pomoćni algoritam. Pomoćni algoritam beleži koje su vrednosti iskorištene u prethodnim iteracijama, jer bi se u suprotnom uvek dobili isti rezultati, kada se algoritam pokrene više puta.

5.1. Predložena rešenja usled nedostatka temperature vazduha

Jedan od velikih problema na koje se nailazi kod dugoročne prognoze potrošnje je nedostatak predviđene temperature vazduha za duži vremenski period, npr. od jedne godine. Problem nedostatka temperature se najčešće prevazilazi, tako što se uradi srednja vrednost temperature ili se radi predviđanje temperature. Srednja vrednost temperature se računa na osnovu par godina podataka.
Predviđanje temperature je rađeno pomoću MLR algoritma, jer su se na prethodnom istraživanju dobili najtačniji rezultati. Opseg podataka koji je korišten za obučavanje modela je od 2007. godine zaključno sa 2011. godinom. Podaci koji su korišteni za predviđanje su u rasponu od 2012. do 2016. godine. Isti skup podataka je korišten za MLR, takođe i za SVM algoritam, dok za prosečenje temperature je potreban samo skup koji je bio korišten za predviđanje tj. podaci od 2012. do 2016. Upoređivanjem dobijenih rezultata greški, može se utvrditi da je najmanja greška dobijena kada je korišteno prosečenje temperature. Greška MLR algoritma je skoro 2.5 puta veća u odnosu na grešku koja je dobijena kada se koristi prosek temperature, dok je greška SVM algoritma najveća.

5.2. Probabilistički pristup

Pošto su opisani algoritmi primarno deterministički, onda je potrebno da se prilikom njihovog izvršavanja ubaci parametar koji je nasumičan, kako bi pomenuti algoritmi bili probabilistički. Jedan od načina na koji je implementiran probabilistički pristup u radu jeste da se dodaje nasumičan broj dana na osnovni broj dana koji se uzima za pronalaženje sličnih dana, što je dalje u tekstu referencirano kao *Prva varijacija*. Osnovni broj dana je pet, dok nasumični broj dana varira između jedan i petnaest, jer se prethodnim istraživanjem dokazalo da se tačnost algoritama pogoršava, kada se koristi preko dvadeset dana za pronalaženje sličnog dana. Na slici 5.2 prikazan je tok ulaza i izlaza podataka iz treniranog modela.



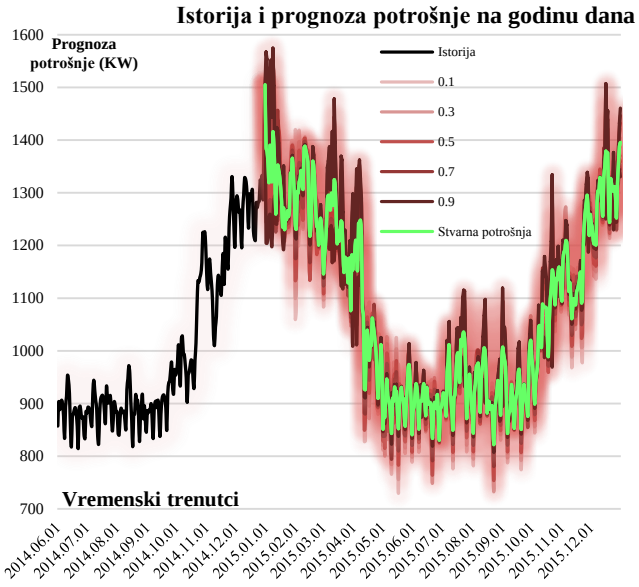
Slika 5.2. *Prikaz ulaza i izlaza iz modela, kada je nasumičan parametar broj sličnih dana nad kojima se radi pretraga i kada je nasumičan parametar broj stepeni, respektivno*

U trenirani model ulazi šest vrednosti od kojih su: tri istorija potrošnje i tri temperatura. Nasumičan broj samo povećava verovatnoću da se pronađe dan koji je najsličniji posmatranom danu po temperaturi.

Drugi probabilistički pristup koji je implementiran jeste da se dodaje nasumičan broj stepeni na osnovu, koji je dalje u tekstu referenciran kao *Druga varijacija*. Osnova na koju se dodaje nasumičan broj stepeni je jedan stepen, kao što je prikazano na slici 5.2. Nasumičan broj stepeni koji se dodaje na osnovu varira između nula i jednog stepena. U trenirani model ulazi šest vrednosti od kojih su: tri istorija potrošnje i tri temperatura.
Dodavanjem nasumičnog broja stepeni na osnovu, povećavamo verovatnoću da se pronađe dan koji ima najviše sličnosti sa posmatranim danom.

6. ANALIZA REZULTATA

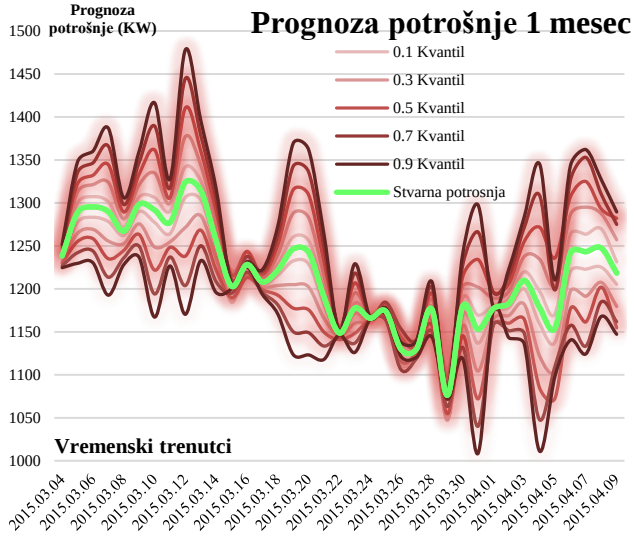
U nastavku će biti prikazani rezultati kada se koristi MLR algoritam za prognozu potrošnje u kombinaciji sa probabilističkim varijacijama algoritama i algoritmima za prognozu temperature. Korišteni algoritmi za prognozu temperature su MLR, SVM i Average (prosečenje). Procena tačnosti algoritama je rađena preko MAPE algoritma. Evaluacijom rezultata se može utvrditi, da prilikom upotrebe SVM algoritma za prognozu temperature u kombinaciji sa prvom varijacijom, dobije se najmanja greška prognoze potrošnje koja varira između 5.13% i 5.37%.



Slika 0.1. *Prikaz istorije i prognoze potrošnje za jednu godinu, primarni MLR algoritam, MLR algoritam za prognozu temperature*

Kada se koristi druga varijacija, najmanja greška se dobije ukoliko se koristi MLR algoritam za prognozu temperature. Greške tada variraju između 5.61% i 5.70%.
Na slici 6.1. prikazana je prognoza potrošnje, koristeći kvantile i Pinball Loss vrednosti. Primarni algoritam koji je upotrebljen je MLR, algoritam koji je korišten za prognozu temperature je takođe MLR.
Pošto je jako puno vrednosti prikazano na grafiku i kvantili ne mogu da se uoče, a samim tim i analiziraju, na sledećem grafiku su prikazane vrednosti prognoze u periodu od jednog meseca. Na grafiku 6.2., mogu se uočiti svi percentili od 0,1 do 0,9. Interval predviđanja je

procena intervala u kojem će završiti budući rezultati, ali sa određenom verovatnoćom. Svaki kvantil nam govori koliko je vrednost veća u odnosu na stvarnu potrošnju. Svi ostali rezultati su jako slični po grafičkom izgledu.



Slika 0.2. *Izdvojeni prikaz prognoze potrošnje za jedan mesec, primarni MLR algoritam, MLR algoritam za prognozu temperatu*

Tačnost prognoze potrošnje, kada je korišten MLR algoritam je raznolika. Sa MAPE algoritmom greška je varirala između 5.05% i 5.71%, pri čemu je prosečna greška iznosila 5.45%. Kada je korišten Winkler Score za procenu tačnosti, greška je u odnosu na SVM algoritam bila veća za skoro $2x10^3$. Prilikom upotrebe Quantile Score za procenu tačnosti, greška je bila značajno manja u odnosu na SVM algoritam.

Tačnosti prognoze potrošnje, prilikom upotrebe SVM algoritma su takođe varirale. Kada je korišten MAPE algoritam za procenu tačnosti, greška je varirala između 5.22% i 7.91%, a u proseku je iznosila 6.12%. Greška koja je dobijena, kada je korišten Winkler Score za procenu tačnosti, u odnosu na MLR algoritam je bila manja za skoro $2x10^3$. Kada je korišten Quantile Score za određivanje tačnosti algoritma, dobijena greška je veća u odnosu na MLR algoritam.

7. ZAKLJUČAK

Električna energija je postala neophodna u svakodnevnom životu. Shodno tome, planiranje potražnje i potrošnje električne energije postaje sve popularnije, a moglo bi se reći skoro pa neophodno.

Upoređivanjem MLR i SVM algoritma, može se zaključiti, da je MLR algoritam bolja opcija za dugoročnu prognozu potrošnje za posmatrani skup podataka. Ako bi se upotrebili hibridni algoritmi i selekcija ili odabir podataka, verovatno bi se dobila bolja tačnost algoritama.

8. LITERATURA

[1] J. H. Pujar, "*Fuzzy Ideology based Long Term Load Forecasting*", Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering, 2010

[2] D. Ali, M. Yohanna, P. M. Ijasini, M. B. Garkida, "*Application of fuzzy – Neuro to model weather parameter variability impacts on electrical load based on long-term forecasting*", Alexandria Engineering Journal, Volume 57, Issue 1, 2018, Pages 223-233, ISSN 1110-0168, 2017

[3] D. Ali, M. Yohanna, M.I. Puwu, B.M. Garkida, "*Long-term load forecast modelling using a fuzzy logic approach*", Natural Science and Engineering, Volume 18, Issue 2, 2016, Pages 123-127

[4] Xu Liwen, Li Chengdong, Xie Xiuying, Zhang Guiqing, "*Long-Short-Term Memory Network Based Hybrid Model for Short-Term Electrical Load Forecasting*", *Information* 9, no. 7: 165, 2018

[5] Tao Hong, Jason Wilson, Jingrui Xie, "*Long Term Probabilistic Load Forecasting and Normalization With Hourly Information*", Ieee Transactions On Smart Grid, Vol. 5, 456-462, 2014

[6] N. Ayub, N. Javaid, S. Mujeeb, M. Zahid, W. Z. Khan, and M. U. Khattak, "*Electricity Load Forecasting in Smart Grids Using Support Vector Machine*", COMSATS University Islamabad, Islamabad 44000, 2019

[7] Yasin, Z. M., Salim, N. A., Aziz, N. F. A., Ali, Y. M., & Mohamad, H, "*Long-term load forecasting using grey wolf optimizer-least-squares support vector machine*", IAES International Journal of Artificial Intelligence, 2020

[8] Berry, W. D. "*Understanding Regression Assumptions, Sage University Paper series on Quantitative Applications in the Social Sciences*", Newbury Park, 1993

[9] M. Pontil, A. Verri, "*Properties of Support Vector Machines*", INFM, Via Dodecaneso, Genova, 1998

[10] B. Dalvi, A. Mishra, W. W. Cohen, "*Hierarchical semi-supervised classification with incomplete class hierarchies*", Carnegie Mellon Uni., Uni. of Massachusetts, 193-202, 2016

[11] J. V. Mynsbrugge, "*Bidding Strategies Using Price Based Unit Commitment in a Deregulated Power Market*", K.U.Leuven, 2010

[12] Tofallis, "A Better Measure of Relative Prediction Accuracy for Model Selection and Model Estimation", 2015

[13] Hyndman, Rob J., A. B. Koehler, "*Another look at measures of forecast accuracy*" IJOF, 2006

[14] Kim, Sungil, H. Kim, "*A new metric of absolute percentage error for intermittent demand forecasts*", 2016

[15] Makridakis, Spyros, "*Accuracy measures: theoretical and practical concerns*", IJOF, 9(4):527-529, 1993

[16] Delfs H., Knebl H., "*Probabilistic Algorithms. In: Introduction to Cryptography. Information Security and Cryptography*", Springer, Berlin, Heidelberg, 2007

[17] Tilmann Gneiting, "Quantiles as optimal point forecasts, International Journal of Forecasting", Volume 27, Issue 2, 2011, Pages 197-207

[18] Chih-Jen Lin, Jorge J. More, "*Newton's Method For Large Bound-Constrained Optimization Problems"*, SIAM J. OPTIM, vol. 9, no. 4, pp. 1100–1127, 1999 Society For Industrial And Applied Mathematics

[19] https://www.entsoe.eu/data/power-stats/ (pristupljeno septembar 2021. godine)
https://power.larc.nasa.gov/data-access-viewer/ (pristupljeno septembar 2021. godine)

Kratka biografija:

Kristina Polih rođena je u Novom Sadu 1996. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehničko i računarsko inžinjerstvo – Primenjeno softversko inžinjerstvo, odbranila je master rad 2021.godine.
kontakt: christina.polih@yahoo.com

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.42:004.738.12
DOI: https://doi.org/10.24867/16BE30Pecic

WEB APLIKACIJA U REALNOM VREMENU SA PODRŠKOM SIGNALR BIBLIOTEKE

REAL TIME WEB APPLICATION WITH SIGNALR LIBRARY SUPPORT

Dušan Pećić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – *Ovaj rad bavi se pregledom web-a u realnom vremenu, kao i svih važnijih tehnika koje pružaju funkcionalnost u realnom vremenu. Takođe, detaljno je objašnjena biblioteka SignalR, koja je jedna od najčešće korišćenja biblioteka za razvoj aplikacija u realnom vremenu. Zatim se prolazi kroz implementaciju aplikacije za chat u realnom vremenu, kao i slanje trenutne lokacije korisnika.*

Ključne reči: *Real-time web, SignalR, ASP.NET Core*

Abstract – *Main goal of this paper is overview of real time web, as if all of the most important tehniques for allowing real time functionality. Also, it is precisely described SignalR library, which is the one of the most used libraries for development of real time web applications. Then we go through implementation of chat application in real time, as well as sending the user curent location.*

Keywords: *Real-time web, SignalR, ASP.NET Core*

1. UVOD

Web je u svojoj ranoj fazi postojanja bio vrlo drugačiji od onoga što danas poznajemo, te je zamišljen za rešavanje specifičnog problema deljenja dokumenata između udaljenih računara. Mogućnost *web*-a koja će se obrađivati u ovom radu su *web* aplikacije koje rade u realnom vremenu. Jedan od nedostataka Web-a je bio taj da je *web* stranicu bilo potrebno ponovo učitati kako bi ona dobila nove podatke od servera. Mogućnost dobijanja podataka bez ponovnog učitavanja stranice otvorila je veliki broj mogućnosti koje su drastično promenile način na koji koristimo *web*.

Danas je ovaj način *web* aplikacija opšte prihvaćen, te se koristi od strane velikog broja *web* stranica, počev od *chat* funkcionalnosti na društvenim mrežama do pretraživanja sadržaja u realnom vremenu na *web* pretraživačima. Razvoj *web* aplikacija u realnom vremenu imao je spor napredak uzrokovan ograničenjima od strane pretraživača, njegovom infrastrukturom i drugim faktorima. Danas postoje standardizovane tehnologije koje podržavaju svi popularniji internet pretraživači, te se koriste u velikom broju javno dostupnih biblioteka i alata za razvoj *web* aplikacija. Internet infrastruktura je takođe napredovala i samim tim više ne predstavlja prepreku u razvoju ovakvih vrsta aplikacija.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Željko Vuković, docent.

Još jedan od razloga jeste to što su korisnici bili izloženi korisničkom iskustvu koje pružaju *web* tehnologije u realnom vremenu i sada zahtevaju tu vrstu iskustva u aplikacijama koje koriste. *Web* tehnologije u realnom vremenu imaju brojne uobičajene upotrebe u svakodnevnom životu, a sve se više otkrivaju nove, inovativne potrebe.

2. RAZUMEVANjE WEB-A U REALNOM VREMENU

Realno vreme (eng. real time), najlakše se može objasniti kao najduže vreme potrebno za izvršenje zadatka u kojem bismo mogli reći da čitav sistem radi prema zadanim specifikacijama. Računanje u realnom vremenu (eng. real time computing), opisuje sisteme koji se temelje na opremi i aplikacijama koji imaju zadato realno vreme. Takve sisteme karakterizuju tri komponente i njihova međuzavisnost, a to su vreme izvršavanja, pouzdanost i okolina sistema. Vreme je najdragoceniji resurs za upravljanje sistemima u realnom vremenu. Zadaci moraju biti dodeljeni i raspoređeni za završetak pre njihovih krajnjih rokova. Pouzdanost se takođe veže na vreme izvršavanja jer svaki deo sistema pretpostavlja da će drugi delovi sistema ispravno izvršiti svoj zadatak u definisanim vremenskim okvirima, jer ukoliko to ne učine to može rezultirati pogrešnim radom sistema. Treća komponenta je okolina sistema i ona predstavlja sve ono što okružuje sistem. tj. spoljašnji delovi kojima se sistem bavi i bez kojih on ne bi imao svrhu [1].

2.1 Šta su *web* aplikacije u realnom vremenu?

Aplikacije u realnom vremenu postepeno dominiraju internetom jer pružaju savršenu ravnotežu informacija, funkcionalnosti, sadržaja i interaktivnosti što na kraju povećava angažovanje korisnika sa aplikacijom. Nepotrebno je naglašavati da aplikacije u realnom vremenu vide veliki obim prometa i stoga zahtevaju efikasno korišćenje hardverskih sredstava sa kojima raspolažu. Ove aplikacije moraju biti lagane, omogućiti iterativan razvoj i pružati sadržaj na organizovan i struktuiran način [2]. Kao što je već rečeno, pojavom *web* aplikacija dolazi do nove dimenzije razvijanja *web-a*, u smeru dinamičkog učitavanja sadržaja na zahtev klijenta, učitavanja sadržaja sa više servera odjednom na istu stranicu, slanja podataka, obrada i slično. Cilj je da korisnik dobije traženu informaciju koja će biti dinamički učitana, što nije problem, dok god klijent zahteva te informacije. Problem nastaje ukoliko se informacija promeni u međuvremenu ili ukoliko pristignu nove informacije koje su vezane za korisnike ili bitne za njegov rad. Da bi se informacija učitala na strani klijenta potrebno je da postoji zahtev koji klijent podnosi serveru

za učitavanje informacija. Sposobnost da klijent dobije informaciju u istom ili gotovo istom trenutku naziva se „*web* u realnom vremenu“.

2.2 Ajax

Asinhroni *JavaScript* i *XML* - *AJAX* (eng. asynchronous JavaScript and XML) je programska tehnika koja koristi JavaScript i objekat *XMLHTTPRequest* za razmenu podataka između *web* pretraživača i *web* servera.
AJAX se koristi za poboljšanje interaktivnosti *web* stranica i pruža programerima načine na koje se pojedinačni delovi stranice mogu ažurirati u realnom vremenu bez potrebe za ponovnim učitavanjem celokupnog sadržaja stranice. Ranije, ako je bilo potrebno ažurirati određeni deo sadržaja na *web* stranici, cela stranica bi se ponovo učitala sa *web* servera, uzrokujući prenos velike količine dupliranih podataka. Korišćenjem *AJAX*-a, sadržaj unutar *web* stranice može se ažurirati na osnovu radnji korisnika. *AJAX* se može opisati kao skup različitih tehnologija koje se mogu koristiti za implementaciju *web* aplikacije koja komunicira sa serverom u pozadini, bez ometanja trenutnog stanja stranice. Konekcije sa serverom se mogu kreirati asinhrono, što znači da korisnik ne mora čekati da server odgovori.

2.3 Polling

Polling je komunikaciona tehnika *web* aplikacija u realnom vremenu u kojoj klijent šalje zahtev serveru tražeći podatke u fiksnim vremenskim intervalima, nakon što dobije odgovor na prethodno poslat zahtev. Klijent ne zna kada server dobija nove podatke, samim tim on kontinuirano šalje nove zahteve serveru, kako bi što pre bio spreman za primanje novih podataka. Klijent šalje normalne *Http* zahteve ali kontinualno u određenim vremenskim intervalima. Server zatim momentalno šalje odgovor koji sadrži nove podatke ili samo prazan odgovor, ako nije bilo podataka za preuzimanje.

2.4 Long Polling

Long Polling je u suštini efikasniji oblik originalne tehnike anketiranja po parametrima trošenja resursa. Upućivanje ponovnih zahteva serveru troši resurse jer svaka nova dolazna veza mora biti uspostavljena, *Http* zaglavlja moraju biti parsirana, upit za nove podatke je potrebno izvršiti, kao i generisati i isporučiti odgovor od servera (obično bez novih podataka). Veza tada mora biti zatvorena i svi resursi očišćeni. Sve navedene nedostatke rešava *Long Polling*. Umesto velikog broja zahteva, klijent upućuje *Http* zahtev serveru na uobičajen način, sa namerom da zatraži podatke koje još nije primio. Ukoliko još nema novih podataka, server drži konekciju otvorenom dok ne pristignu podaci koje je korisnik zatražio. Ako posle određenog vremenskog perioda novi podaci i dalje nisu dostupni, tada server šalje odgovor pauze (eng. timeout response).

2.5 Server Sent Events

Server Sent Events je push tehnologija servera koja omogućava klijentu da prima automatski ažurirane podatke sa servera putem *Http* veze i opisuje kako serveri mogu započeti prenos podataka prema klijentima nakon uspostavljanja inicijalne konekcije. *SSE* je zasnovan na nečemu što se zove *Server Sent DOM Events*. Ideja je jednostavna, klijent (*web* pretraživač) se može pretplatiti na tok događaja koje generiše server, primajući ažurirane podatke kad se dogodi novi događaj. To je dovelo do kreiranja popularnog interfejsa *Event Source* koji prihvata *Http* vezu i održava vezu otvorenom dok iz nje preuzima dostupne podatke [3]. *SSE* je dizajniran da koristi *JavaScript EventSource API* (Programski interfejs aplikacije) u cilju pretplate na tok događaja u bilo kom popularnijem pretraživaču. Ukratko, *Server Sent Events* je tehnologija gde se ažurirani podaci guraju (umesto da se povlače ili traže od strane klijenta) sa servera u pretraživač.

2.6 Web Sockets

Pre nego što objasnim šta je *WebSockets* protokol i kako funkcioniše, osvrnućemo se na neke izazove koje ima za cilj da reši. Današnje aplikacije zahtevaju razmenu poruka sa malim kašnjenjem u realnom vremenu. Bez obzira da li pravite *web* ili mobilnu aplikaciju, korisnici očekuju mogućnost interakcije sa podacima što je moguće bliže realnom vremenu uz minimiziranje uticaja na njihovo celokupno korisničko iskustvo. Danas postoji mnogo aplikacija sa kojima ste verovatno u interakciji koje zahtevaju povezivanje preko interneta, a ipak pružaju korisničko iskustvo u realnom ili skoro realnom vremenu. Ako ste bili u interakciji sa *Twitter*-om ili *Facebook*-om, doživeli ste da se tokovi aktivnosti u stvarnom vremenu stalno ažuriraju po celom *web*-*u*, na mobilnom uređaju ili u pretraživaču, bez potrebe da pritisnete dugme za osvežavanje. *WebSockets* je mrežni protokol koji rešava većinu navedenih problema, omogućavajući brzu, sigurnu, dvosmernu komunikaciju između klijenta i servera bez oslanjanja na više *HTTP* veza. Za razliku od *HTTP*-a koji koristi zahtev-odgovor obrazac, *WebSockets* mogu slati poruke u bilo kom smeru u bilo kom trenutku. Najbolje od svega, *WebSockets* je interoperabilan i na više platformi na nivou pretraživača, izvorno podržava portove 80/443, kao i konekcije između domena

3. Pregled ASP.NET Core SignalR-a

Život se dešava u realnom vremenu i razmena informacija treba da se odvija na isti način. To je lakše reći nego učiniti, a izazovi nadilaze gomilu tehnologije. Međutim, kako komunikacija između servera/klijenata postaje sve pametnija, programeri su uspeli da iskoriste sofisticirane tehnike za izgradnju komunikacije u realnom vremenu između aplikativnih platformi. Sa modernim okvirima otvorenog koda, poput *SignalR*-a, koji apstraktuje složenost mrežnog steka, programeri se mogu usredsrediti na funkcionalnosti aplikacija u realnom vremenu koje omogućavaju moćna rešenja. *SignalR* je jedna od najčešće korišćenih biblioteka za rezvoj *web* aplikacija u realnom vremenu. Stvorena je 2011. godine od strane David Fowlera i Damian Edwardsa, koji sada igraju ključne uloge u razvoju *ASP.NET*-a. Da bi postigli razmenu poruka u realnom vremenu za *web*, programeri su koristili neefikasne tehnike, poput *AJAX*-a i long polling-a, kao i tehnologije poput *Server Sent Events*-a koje pretraživači nisu često primenjivali. *SignalR* je predodređen da reši ovaj problem i obezbedi laku podršku za funkcionalnosti u realnom vremenu na *ASP.NET* steku stvaranjem biblioteka na strani servera i klijenta koje uklanjaju

komplikacije ovih tehnologija [4]. Iza kulisa, *SignalR* pregovara o najboljem protokolu za određenu konekciju na osnovu onoga što podržavaju i server i klijent. Zatim pruža dosledan *API* za slanje i primanje poruka u realnom vremenu.

3.1 Šta je SignalR?

ASP.NET Core SignalR je biblioteka otvorenog koda koja pojednostavljuje dodavanje *web* funkcionalnosti u realnom vremenu aplikacijama. *Web* funkcionalnost u realnom vremenu omogućava serverskoj strani da dostupan sadržaj istog trenutka prosledi povezanim klijentima. *SignalR* se može koristiti za dodavanje bilo koje vrste *web* funkcionalnosti u „realnom vremenu“ u vašu *ASP.NET* aplikaciju. *SignalR* pruža jednostavan *API* za kreiranje udaljenih procedura od servera do klijenta (RPC) koje pozivaju JavaScript funkcije u klijentskim pretraživačima (i drugim klijentskim platformama) iz *.NET Core* koda na strani servera. *SignalR* takođe uključuje *API* za upravljanje vezama (na primer, događaje povezivanja i isključivanja) i grupisanje veza.
Pomoću *SignalR*-a možete slati poruke svim povezanim klijentima istovremeno ili ciljati određenog klijenta ili grupu klijenata [5]. Veza između klijenta i servera je trajna, za razliku od klasične *HTTP* veze, koja se ponovo uspostavlja za svaku komunikaciju. *SignalR* podržava funkciju "server push", u kojoj kod na serverskoj strani može da pozove klijentski kod u pretraživaču pomoću udaljenih poziva procedura (eng. Remote Procedure Calls), umesto zahtev-odgovor modela koji se i danas koristi na *web*-u. *SignalR* aplikacije mogu se proširiti na hiljade klijenata pomoću ugrađenih i nezavisnih provajdera za proširenje.

3.2 Razlike u odnosu na .NET SignalR verziju

Između *ASP.NET SignalR*-a i *ASP.NET Core SignalR*-a bilo je nekoliko značajnih promena, ovo su neke od njih [4] :

- Klijentska *JavaScript* biblioteka

U pretraživaču je najveća promena uklanjanje zavisnosti od *jQuery*-ja. Klijentska *JavaScript*/*TypeScript* biblioteka sada se može koristiti bez pozivanja na *jQuery*, što joj omogućava da se koristi sa okvirima kao što su *Angular*, *React* i *Vue* bez problema.

- Ugrađeni i prilagođeni protokoli

ASP.NET Core SignalR se isporučuje sa novim *JSON* protokolom za poruke koji nije kompatibilan sa ranijim verzijama *SignalR*-a. Osim toga, takođe ima drugi ugrađeni protokol zasnovan na *MessagePack*-u, binarnom protokolu koji ima manje korisnog opterećenja od *JSON*-a baziranom na tekstu.

- Dependency Injection

ASP.NET Core SignalR jednostavno koristi ugrađeni okvir za ubrizgavanje zavisnosti. Čvorovi u *ASP.NET Core SignalR*-u sada podržavaju konstruktor *Dependency Injection* bez dodatne konfiguracije, baš kao što to rade *ASP.NET Core* kontroleri ili razor pages.

- Rekonekcija

ASP.NET Core SignalR ne podržava automatsko ponovno povezivanje ili automatsko skladištenje poruka.

3.3 The Azure SignalR Service

Azure SignalR Service pojednostavljuje proces dodavanja *web* funkcionalnosti u realnom vremenu aplikacijama preko *HTTP*-a. Ova funkcionalnost u realnom vremenu omogućava servisu da prosleđuje ažuriran sadržaj povezanim klijentima, poput single page *web* stranice ili mobilne aplikacije. Kao rezultat toga, klijenti se ažuriraju bez potrebe za pozivanjem servera ili podnošenjem novih *HTTP* zahteva za ažuriran sadržaj [6].
Za šta se koristi *Azure SignalR Service*? Svaki scenario koji zahteva guranje (eng. pushing) podataka sa servera na klijenta u realnom vremenu može koristiti *Azure SignalR Service*. Tradicionalne funkcije u realnom vremenu koje često zahtevaju polling sa servera takođe mogu da koriste *Azure SignalR Service*.

4. IMPLEMENTACIJA *CHAT* APLIKACIJE

4.1 Specifikacija aplikacije

Glavni zadatak ovog projekta jeste *web* aplikacija *Chat* koja predstavlja upravo aplikaciju za ćaskanje preko interneta između različitih korisnika. Da bi korisnik imao mogućnost korišćenja aplikacije potrebno je da napravi svoj nalog, nakon čega može da može da se loguje i koristi aplikaciju. Pored klasične razmene poruka, korisnik ima mogućnost i slanja svoje lokacije.

4.2 Opis korišćenih alata i tehnologija

Aplikacija se sastoji od serverskog i klijentskog dela. Za serverski deo aplikacije je korišćena *ASP.NET.Core* tehnologija i *SQLite*, kao i *Microsoft Visual Studio* kao alat. Za klijentski deo aplikacije je korišćen *Blazor WebAssembly*, *Google maps*, dok je od alata takođe korišćen *Microsoft Visual Studio*.

4.2.1 Alati

Microsoft Visual Studio je integrisano razvojno okruženje, koje je korišćeno za sam razvoj aplikacije. *IDE* (Integrisano razvojno okruženje) je program bogat funkcijama koji se može koristiti za mnoge aspekte razvoja softvera.

4.2.2 Tehnologije

ASP.NET Core je *open-source* i *cloud* optimizovan *web* okvir za razvoj savremenih *web* aplikacija koje se mogu razvijati i pokretati na *Windows*, *Linux* i *Mac* operativnim sistemima.

Blazor je besplatan open-source *web* okvir, razvijen od strane *Microsoft*-a, koji omogućava kreiranje *web* aplikacija korišćenjem *C#* i *HTML*-a. *Blazor WebAssembly* je *single-page* okvir za kreiranje interaktivnih *web* aplikacija na strani klijenta, zasnovanog na *.NET*-u.

SQLite je *ACID*-kompatibilan ugrađen sistem za upravljanje bazama podataka sadržan u relativno maloj *C* programskoj biblioteci. Za razliku od klijent-server sistema za upravljanje bazama podataka, jezgro *SQLite*-a nije samostalan proces sa kojim aplikacija komunicira. Umesto toga, *SQLite* biblioteka je uvezana i postaje sastavni deo aplikacije.

HTML (engl. Hyper Text Markup Language) je opisni jezik specijalno namenjen opisu *web* stranica. Pomoću njega se jednostavno mogu odvojiti elementi kao što su naslovi, paragrafi, citati i slično.

CSS (engl. Cascading Style Sheets) je jezik formatiranja pomoću kog se definiše izgled elemenata *web* -stranice.

4.3 Implementacija *chat* funkcionalnosti pomoću SignalR-a

Blazor WebAssembly aplikacija je kreirana pomoću *Microsoft Visual Studio*-a. *Blazor WebAssembly* aplikacije rade na klijentskoj strani u pretraživaču na *.NET runtime* zasnovanom na *WebAssembly*-ju. *Blazor* aplikacija se izvršava direktno na niti korisničkog interfejsa pretraživača. Ažuriranje korisničkog interfejsa i rukovanje događajima odvija se u okviru istog procesa. Aplikacija je hostovana, što znači da je kreirana sa pozadinskom aplikacijom *ASP.NET Core* za opsluživanje datoteka. Dakle, aplikacija se sastoji iz klijentskog i serverskog dela. Hostovana klijentska aplikacija komunicira sa pozadinskom server aplikacijom preko mreže pomoću *web API*-ja ili *SignalR*-a. Glavni zadatak master rada je kreiranje *Chat* aplikacije koja će omogućiti razmenu poruka između dva autentifikovana korisnika u realnom vremenu. Da bi neautentifikovani korisnik postao autentifikovan, potrebno je da napravi svoj nalog u aplikaciji, nakon čega dobija mogućnost ćaskanja sa drugim korisnicima koji imaju nalog u ovoj aplikaciji. Za postizanje *Chat* funkcionalnosti klijent će biti hostovan na pretraživaču, dok će serverska stranom biti hostovana na računaru. Klijenti su povezani na server, kako bi mogli da primaju i šalju podatke na server.

4.3.1 Slanje lokacije pomoću *Chat* aplikacije

Dodatna funkcionalnost koja je implementirana u *Chat* aplikaciju predstavlja slanje lokacije drugom korisniku. Dobavljanje trenutne geografske lokacije se vrši pomoću *Geolocation API*-ja. Uobičajeni izvori informacija o lokaciji uključuju *GPS* i lokaciju izvedenu iz mrežnih signala, kao što su *IP* adresa, *WiFi*, *Bluetooth* itd. Kako bismo dobili trenutnu lokaciju pokreće se asinhroni zahtev za detekciju lokacije korisnika i traži najnovije informacije od hardvera za pozicioniranje. Kada se odredi položaj korisnika izvršava se definisana funkcija povratnog poziva. Uređajima za *GPS*-om može biti potrebno do nekoliko minuta za dobijanje potpuno tačne lokacije, zato manje precizni podaci (*IP* lokacija ili *WiFi*) mogu biti vraćeni. Nakon što korisnik odobri dozvolu, pomoću *Geolocation API*-ja dobijamo informacije o geografskoj širini i dužini, nakon čega je potrebno te podatke prikazati na mapi. Za prikaz lokacije u chat-u korisnika prvo je potrebno dodati *GoogleMaps API* skriptu.

5. ZAKLJUČAK

Ovim poglavljem završavamo sa pregledom razvoja *web* aplikacija u realnom vremenu. Kroz ovaj rad smo se na kratko osvrnuli na istoriju *web-a* i interneta, gde su pojašnjeni razlozi nastanka i razvoja *web* aplikacija u realnom vremenu. Takođe je objašnjen i sam pojam realnog vremena, njegov način računanja i vrste sistema koji funkcionišu u realnom vremenu. Prošli smo kroz obrazloženje aplikacija u realnom vremenu i njihovih karakteristika, gde smo se dotakli svih važnijih tehnika za postizanje realnog vremena sa prednostima i manama svake od njih. Detaljno je opisana *SignalR* biblioteka koja je korišćena pri implementaciji *Chat* aplikacije, njena istorija nastanka, transporti koje koristi, opis komunikacije čvora sa klijentima i prednosti koje su postignute sa ovom bibliotekom.

Završnim primerom prikazan je jedan mogući slučaj korišćenja ovakve vrste aplikacija, koja je izrađena uz pomoć pomenute *SignalR* biblioteke, jedne od najkvalitetnijih biblioteka za izradu *web* aplikacija u realnom vremenu današnjeg doba. *Web* se razvija iz dana u dan, a *web* aplikacije su već neko vreme deo naše svakodnevnice. Responzivne i kvalitetne aplikacije definitivno su tome doprinele, dok je napredak infrastrukture interneta povećao mogućnosti u izradi ovih aplikacija. Razvojem biblioteka poput one opisane u ovom radu, kao i razvojem *web-a* i interneta, doprinosi se tome da ne postoji razlog zašto različite aplikacije ne bi iskoristile neke od mogućnosti i prednosti koje pružaju *web* aplikacije u realnom vremenu. Na taj način će korisničko iskustvo, uspeh na tržištu, kao i sam razvoj ovakvih aplikacija biti podignut na jedan viši nivo.

6. LITERATURA

[1] Shin.K.G. & Ramanathan, P. (1994). Real-Time Computing: A New Discipline of Computer Science and Engineering

[2] https://www.clariontech.com/blog/key-considerations-while-developing-a-real-time-web-application

[3] https://developer.mozilla.org/en-US/docs/Web/API/EventSource

[4] https://www.codemag.com/article/1807061/Build-Real-time-Applications-with-ASP.NET-Core-SignalR

[5] https://docs.microsoft.com/en-us/aspnet/signalr/overview/getting-started/introduction-to-signalr

[6] https://docs.microsoft.com/en-us/azure/azure-signalr/signalr-overview

Kratka biografija:

Dušan Pećić rođen je 08.12. 1995. godine u Novom Sadu gde i dalje živi. Osnovnu školu „Jovan Jovanović Zmaj“ u Sr. Kamenici završio je 2010. godine kao nosilac Vukove diplome. Elektrotehničku školu „Mihajlo Pupin“ u Novom Sadu završio je 2014. godine. Iste godine upisuje Fakultet tehničkih nauka, smer Elektroenergetski softverski inženjering. U oktobru 2019. godine završava osnovne studije, nakon čega upisuje master studije takođe na Fakultetu tehničkih nauka, smer Računarstvo i Automatika, podsmer – Elektronsko poslovanje.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.38:620.9
DOI: https://doi.org/10.24867/16BE31Ikonic

JEDNO REŠENJE SISTEMA ZA OPTIČKO PREPOZNAVANJE KARAKTERA UPOTREBOM DUBOKIH NEURONSKIH MREŽA

ONE SOLUTION OF OPTICAL CHARACTER RECOGNITION USING DEEP NEURAL NETWORKS

Dejan Ikonić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U okviru ovog rada pokazana je praktična primena dubokih konvolucijskih neuronskih mreža u oblasti detekcije karaktera od značaja na slikama sa identifikacionim oznakama bandera ulične rasvete. Zadatak je realizovan u tri iteracije sa dobijenom tačnošću od 93,04%.*

Ključne reči: *Duboko učenje, Neuronske mreže, Veštačka inteligencija*

Abstract – *This paper presents one approach of automated optical character recognition on photographs with streetlights identification using deep Convolutional Neural Network. The task was realized in three iterations, with an accuracy of 93.04%.*

Keywords: *Deep learning, Artificial intelligence, Neural networks*

1. UVOD

U ovom radu opisano je jedno rešenje za detekciju petocifrenih identifikacionih brojeva na identifikacionim listovima sa fotografija stubova ulične rasvete. Cilj rada je razvoj sistema koji radi na principu najnovijih tehnologija i koji se uz jednostavnu pripremu ulaznog skupa i konfiguraciju parametara sistema može primeniti za rešavanje različitih slučajeva optičkog prepoznavanja karaktera (eng. Optical Character Recognition - OCR).

Alati veštačke inteligencije su pogodni za ovakve probleme jer nude rešenja segmentacije i klasifikacije na šta se OCR problem može i svesti.

Primere primene možemo naći od ranih radova kao što je opisano u [1] iz 1996. godine do novih rešenja kao što nudi članak [2] iz 2019. godine. Implementacije zasnovane na korišćenju veštačkih neuronskih mreža u prepoznavanju teksta i simbola, prvi je iskoristio Dan Ciresan i njegove kolege na takmičenju [3]. Njihove neuronske mreže su bile prvi veštački prepoznavači šablona, koji su mogli da dostignu performanse koje mogu da pariraju čoveku kod prepoznavanja saobraćajnih znakova i rukopisa [4].

Sistem koji je predstavljen u ovom radu realizovan je upotrebom duboke konvolucijske neuronske mreže koja se koristi za segmentaciju karaktera, i daje ulaze za neuronsku mrežu sa povratnim vezama za klasifikaciju identifikovanih karaktera.

Opisan sistem je kreiran, a rezultati obučavanja i testiranja su predstavljeni u ovom radu.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Vladimir Bugarski, docent.

2. BAZA ULAZNIH PARAMETARA MREŽE

Na fotografijama u bazi prikazani su stubovi ulične rasvete sa istaknutim nalepnicama koje nose njihov identifikacioni broj. Cilj sistema je da za datu fotografiju detektuje i prikaže njen identifikacioni broj. Bazu sačinjava skup od 11.000 fotografija. Baza slika je podeljena tako da je za obuku mreže odvojeno 9.000 fotografija, dok je ostatak od 2.000 služio kao baza za testiranje mreže. Primer fotografije je vidljiv na slici 1.

Slika 1. *Primer fotografije za prepoznavanje*

Korišćene su dve metode preprocesiranja: smanjivanje dimenzija i augmentacija.

2.1. Iteracija 1: 64x64 bez augmentacije

Fotografije za trening i testiranje, su sa originalne veličine smanjene na veličinu 64x64 piksela. Osim smanjenja veličine, osetno je smanjen i kvalitet fotografija, što je uticalo na kvalitet same obuke.

2.2 Iteracija 2: 128x128 bez augmentacije

Originalne fotografije u iteraciji 2 smanjene su na veličinu 128x128 piksela.

2.3 Iteracija 3: 128x128 sa augmentacijom

Cilj, u iteraciji 3, je bio da se poveća broj nekarakterističnih fotografija u skupu, kao što je slika 2, a da se pri tome ne preopteretи obuka mreže sa velikim skupom podataka. To je postignuto koristeći operaciju rotacije iz [5] koja rotira fotografiju na ulazu za nasumičnu vrednost između zadatih granica, od -45° do 45°. Na ovaj način je dobijen skup od 18.000 fotografija kao trening skup podataka.

Slika 2. *Primer slabo vidljive fotografije za prepoznavanje*

3. ARHITEKTURA MREŽE

Implementiran je sistem za prepoznavanje koji sadrži konvolucijsku neuronsku mrežu i povratnu neuronsku mrežu sa dugotrajnim pamćenjem (eng. Long Short-Term Memory - LSTM) sa modelom zapažanja.

Konvolucijska neuronska mreža je zadužena za segmentaciju fotografija i učenje regiona od interesa. Ova mreža detektuje delove na fotografiji koji sadrže karaktere za prepoznavanje, od njih pravi jednodimenzionalne sekvencijalne regione koje predaje LSTM delu mreže čiji je zadatak da date regione analizira, prepozna karaktere i prosledi rezultate na izlaz. Korišćena konvolucijska mreža ima sledeće slojeve:

1. Sloj sa ispravljačkim mehanizmom praćen slojem sažimanja maksimalnom vrednošću po obe dimenzije;
2. Sloj sa ispravljačkim mehanizmom praćen slojem sažimanja maksimalnom vrednošću po obe dimenzije;
3. Sloj sa normalizacijom i ispravljačkim mehanizmom;
4. Sloj sa ispravljačkim mehanizmom praćen slojem sažimanja maksimalnom vrednošću po širini;
5. Sloj sa normalizacijom i ispravljačkim mehanizmom;
6. Sloj sa ispravljačkim mehanizmom praćen slojem sažimanja maksimalnom vrednošću po širini;
7. Sloj sa normalizacijom i ispravljačkim mehanizmom praćen slojem sažimanja maksimalnom vrednošću po širini.

Kao enkoder i dekoder u implementiranom sistemu koristi se LSTM mreža i model zapažanja. Broj skrivenih slojeva u mrežama korišćenim za enkoder i dekoder je 128. Enkoder kao ulaz prima vektor karaktera za prepoznavanje koji daje prethodna mreža. Dekoder koristi model zapažanja koji omogućava da naučene karakteristike na ulaznoj sekvenci utiču na kasnije karakteristike karaktera koji se dekodiraju. Zapažanje funkcioniše tako što računa kontekst vektor sa težinama svakog elementa na ulazu po važnosti za određenu operaciju na izlazu. Pored ovoga dekoder koristi pretragu snopom koja pomoću jezičkog modela određuje vrednost svakog karaktera za prepoznavanje. U ovom slučaju jezički model su cifre 0-9.

4. PROCES UČENJA SISTEMA

Za treniranje mreža u ovom radu korišćen je algoritam prostiranja greške unazad. Na ulazni sloj mreže se dovode ulazni podaci da bi se započelo učenje. Sledeći korak je propuštanje signala kroz mrežu i generisanje krajnjeg rezultata na izlaznom sloju. Greška predstavlja razliku između generisanih i željenih rezultata. Tokom učenja izlaz mreže se približava željenom, a greška se smanjuje. Prvo se određuje greška u izlaznom sloju, zatim se za prethodni sloj računa koliko je uticaja imao svaki neuron na grešku sledećeg sloja [6]. Greška izlaza neurona se računa kao suma razlike kvadrata dobijenog izlaza g_k i željenog izlaza o_k i data je formulom (1):

$$J(w) = \frac{1}{2} \bullet \sum_{k=1}^{c} (g_k - o_k)^2 = \frac{1}{2} \|g - o\| \quad (1)$$

U formuli (1) g i o su dobijeni i željeni izlazni vektori dužine c, a w predstavlja težine unutar mreže. Težine su na početku nasumično inicijalizovane i vrednosti težina se menjaju u toku treniranja, u cilju smanjivanja greške, po izrazu (2):

$$\Delta w = -\eta \frac{\partial J}{\partial w} \quad (2)$$

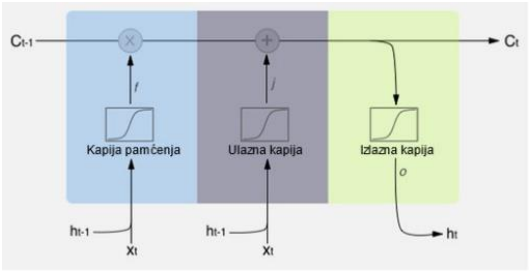
Relativna stopa učenja je predstavljena sa η. Jednačina (1) pokazuje da J nikada ne može dostići negativnu vrednost, što osigurava da će proces učenja u jednom trenutku da se zaustavi, po nekom od zadatih kriterijuma. Izračunata greška trenutnog sloja se zatim propagira na prethodni sloj. Ovo se postiže sumiranjem uticaja neurona iz prethodnog sloja j na neurone iz trenutnog sloja i, kao što je prikazano izrazom (3):

$$\frac{\partial J}{\partial y_i} = \sum_j \frac{\partial J}{\partial net_j} \frac{\partial net_j}{\partial y_i} = \sum_j w_{ij} \frac{\partial J}{\partial net_j} \quad (3)$$

Parametar *net* predstavlja aktivacijsku funkciju. Sledeći korak je računanje parcijalne derivacije po težinama, po jednačini (4):

$$\frac{\partial J}{\partial w_{ij}} = \frac{\partial J}{\partial net_j} \frac{\partial net_j}{\partial w_{ij}} \quad (4)$$

Kod LSTM mreže postoje kapije za propuštanje znanja i stanje ćelije. Stanje ćelije u ovoj mreži služi za propagiranje znanja iz prošlih iteracija, dok kapije nose informaciju da li određenu osobinu u ćeliji mreže treba uzeti u obzir ili ne. Kapije predstavljaju sigmoidne aktivacijske funkcije sa izlaznom vrednošću između 0 i 1. Postoje tri kapije u jednoj ćeliji povratne LSTM mreže (sika 3): ulazna i, kapija pamćenja f i izlazna o.



Slika 3. *LSTM ćelija*

Jednačina za ulaznu kapiju (5) nam govori koje nove informacije će uticati na stanje date ćelije. Kapija pamćenja u svojoj jednačini (6) opisuje koje informacije ne treba uzeti u obzir za datu ćeliju. Izlazna jednačina (7) se koristi kao aktivacijska funkcija konačnog izlaza LSTM ćelije.

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \quad (5)$$

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \quad (6)$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o \quad (7)$$

gde je:

- σ - aktivacijska sigmoidna funkcija
- w_x - težina neurona na datoj kapiji (x)
- h_{t-1} - izlaz prethodnog LSTM bloka
- x_t - tekući ulaz
- b_x – gradijent date kapije (x)

Stanje LSTM ćelije, kandidat stanja i konačni izlaz dobijaju se po formulama (8), (9) i (10).

$$\widetilde{c_t} = \tanh(w_c[h_{t-1}, x_t] + b_c) \quad (8)$$

$$c_t = f_t * c_{t-1} + i_t * \widetilde{c_t} \quad (9)$$

$$h_t = o_t * \tanh(c^t) \quad (10)$$

gde je:

- c_t - stanje čelije u trenutku t
- $\widetilde{c_t}$ - kandidat za stanje ćelije u trenutku t

Jednačina (9) pokazuje da u svakom trenutku LSTM ćelija zna šta treba da zanemari iz prethodnog stanja, a šta da uzme u obzir iz tekućeg ulaza. Nakon ovoga, stanja ćelija se filtriraju i propuštaju kroz aktivacijsku funkciju koja predviđa šta se treba pojaviti kao izlaz u datom trenutku t.

Konvolucijska mreža i LSTM mreža su povezane. Izlaz konvolucijske korišćen je kao ulaz LSTM. Obučavanje je podeljeno na manje celine: epohe i gomile. Epoha predstavlja proces u kome je kompletan skup podataka za trening propušten kroz mrežu jedanput, što predstavlja jednu iteraciju gradijentnog spusta. Gomila predstavlja broj podataka za treniranje nakon koga se model obučavanja osvežava.

4.1 Iteracija 1: 64x64 bez augmentacije

Za obuku sistema je korišćeno 9.000 fotografija veličine 64x64 piksela. Podaci su podeljeni u gomile od 150 fotografija i 200 epoha. Ovakva podela znači da se nakon obrađenih 150 fotografija model učenja ažurira, a svih 9.000 fotografija je prošlo kroz mrežu 200 puta. Kriterijum zaustavljanja je postavljen tako da se obuka završi nakon svih 200 epoha ili kada greška padne ispod 10^{-4}. Treniranje je završeno nakon 200 epoha, sa greškom 10^{-2}.

4.2 Iteracija 2: 128x128 bez augmentacije

Za drugi pokušaj, 9.000 fotografija za treniranje mreže veličine 128x128 piksela, podeljeno je u 120 gomila, obučavanih u 220 epoha. Kriterijum zaustavljanja je postavljen na 220 epoha ili dostignutu vrednost greške od 10^{-4}. Greška je u ovom slučaju smanjena na 10^{-3}, nakon što je završeno svih 220 epoha treniranja.

4.3 Iteracija 3: 128x128 sa augmentacijom

Za treću iteraciju skup za obučavanje je, nakon augmentacije, imao 18.000 fotografija veličine 128x128 piksela. Fotografije su podeljene u gomile od po 80, dok je definisani broj epoha bio 250. Kriterijum zaustavljanja je podešen na završetak svih 250 epoha ili vrednost greške 10^{-4}. Obučavanje je prekinuto kada je greška pala ispod 10^{-4}, da bi se izbegla loša generalizacija podataka.

5. TESTIRANJE MREŽE I REZULTATI

Testiranje je izvršeno nad skupom slika koje nisu učestvovale u trening fazi da bi se proverilo da mreža nije pretrenirana i da nije naučila neke odlike specifične samo za podatke u okviru trening skupa [7].

Skup od 2.000 fotografija za testiranje je propušten kroz prethodno obučenu mrežu, jedna po jedna, zajedno sa očekivanim vrednostima za svaku od njih. Klasifikovane vrednosti karaktera su zatim sačuvane zajedno sa očekivanim. Nad rezultatima testiranja je izvršena analiza koja govori o uspešnosti predloženog sistema.

Testiranih 2.000 fotografija su sadržale po 5 karaktera kao identifikacioni broj bandere, što znači da je sistem trebao da prepozna ukupno 10.000 karaktera. Odnos prepoznatih karaktera i ukupnog broja za prepoznavanje uzet je kao parametar uspešnosti sistema. Pored ovoga, analizirane su još neke karakteristike koje su se istakle kao bitne prilikom treniranja i testiranja: uspešnost prepoznavanja kompletnog identifikacionog broja, prepoznavanje u odnosu na cifru koja treba da se prepozna i u odnosu na poziciju u identifikacionom broju.

Kada se uporede rezultati treniranja i testiranja mreže (Tabela 1) može se uočiti da je svakom od izmena koje su napravljene, poboljšana obučenost sistema da uspešno identifikuje fotografije u fazi testiranja. To se najbolje vidi u broju prepoznatih karaktera, koji se kreće od 66,53% u prvoj iteraciji do 93,04% u trećoj.

Tabela 1. *Uspešnost prepoznavanja*

	Očekivano	**Prepoznato**	**Procenat uspešnosti**
Iteracija 1	10000	6653	66,53%
Iteracija 2		8744	87,44%
Iteracija 3		9304	93,04%

Sistem je u iteraciji 2, povećanjem slika za trening i test, obučen da mnogo bolje uočava cifre na fotografijama koje su napravljene sa veće udaljenosti ili imaju neku smetnju. To se može jasno videti i u smanjenju broja fotografija sa velikim brojem promašaja kroz iteracije (Tabela 2).

Tabela 2. *Uspešnost prepoznavanja kompletnog identifikacionog broja kroz iteracije*

Prepoznato cifara	**Iteracija 1**	**Iteracija 2**	**Iteracija 3**
0	29	0	0
1	276	28	0
2	294	280	12
3	405	321	142
4	406	539	376
5	590	832	1470
Ukupno:	**2000**	**2000**	**2000**

U iteraciji 3, gde je za trening korišćen veći skup podataka, postignuto je to da je sistem mnogo bolje naučio osobine brojeva koji se ređe javljaju, jer nisu na prve dve pozicije, kao 2 i 3. Ovi podaci su jasno vidljivi u uporednom prikazu uspešnosti u odnosu na broj koji se prepoznaje (Tabela 3).

Tabela 3. *Uspešnost prepoznavanja u odnosu na broj koji se prepoznaje*

Očekivana cifra	Broj cifara	Iteracija 1	Iteracija 2	Iteracija 3
0	816	569	678	742
1	743	448	674	714
2	1993	1636	1940	1973
3	1579	686	1418	1538
4	845	221	717	836
5	753	568	656	692
6	673	496	589	632
7	733	630	642	665
8	895	612	639	680
9	970	787	791	832
Ukupno:	**10000**	**6653**	**8744**	**9304**

Analiza uspešnosti, u odnosu na poziciju broja koji se prepoznaje, pokazuje da se i dalje cifre slabije prepoznaju što im je veća pozicija u identifikacionom broju. Uzrok je kombinacija dva pomenuta problema, u većim pozicijama se češće pojavljuju brojevi koji nisu 2 i 3, a fotografije su takve da su često te pozicije nejasno vidljive. Popravkom parametara obučavanja kroz 3 iteracije uspeli smo da i u ovom kontekstu postignemo veliku tačnost (Tabela 4).

Tabela 4. *Uspešnost u odnosu na poziciju broja koji se prepoznaje*

Pozicija	Broj cifara	Iteracija 1	Iteracija 2	Iteracija 3
1	2000	569	1921	1990
2	2000	448	1842	1975
3	2000	1636	1830	1929
4	2000	686	1674	1816
5	2000	221	1477	1594
Ukupno:	**10000**	**6653**	**8744**	**9304**

Krajnji rezultat uspešnosti sistema se smatra zadovoljavajućim za skup podataka koji je korišćen.

6. ZAKLJUČAK

Glavni zadatak ovog rada bio je da da jedno rešenje problema koji je dosta rasprostranjen u modernom svetu: optičko prepoznavanje karaktera (OCR).

Opisani sistem sastoji se od konvolucijske neuronske mreže koja za zadatak ima da vrši segmentaciju fotografija i izdvaja regione od interesa, koje sadrže karaktere za prepoznavanje, i predaje ih LSTM mreži sa modelom zapažanja koja radi klasifikaciju dobijenih regiona i kao rezultat daje karaktere sa slike u digitalnoj formi.

Glavna prednost ovakvog sistema su prilično jednostavno preprocesiranje, jer ne zahteva obeležavanje regiona od interesa u cilju pripreme za učenje. Zbog specifičnosti skupa podataka odrađen je i korak augmentacije ulaznog skupa podataka u cilju isticanja osobina prisutnih na manjem delu skupa.

Sistem opisan u ovom radu je moguće primeniti na bilo koji OCR problem. Ovo je moguće postići jednostavnim treniranjem nad novim skupom uz eventualnu izmenu karaktera skupa za prepoznavanje.

Dobijeni rezultati na test skupu su prilično dobri za ati problem i korišćeni skup podataka. Korišćen je veoma veliki broj fotografija za prepoznavanje i dobijena je tačnost od 93,04%. S obzirom na raznolikost skupa podataka, gde su fotografije pravljene sa različitih udaljenosti od znaka, pod različitim uglovima, a sami objekti bili zakrivljeni, ovakav rezultat je zadovoljavajući. Uspešnost bi se mogla povećati dodatnim filtriranjem skupa podataka, gde bi se izbacile nejasne fotografije ili primenom namenskog sistema koji se bazira na detekciji i zahteva ručnu pripremu trening skupa.

7. LITERATURA

[1] C. Tanprasert, T. Koanantakool: "Thai OCR: a neural network application", Proceedings of Digital Processing Applications, vol. 1, pp. 90-95, Perth, Australia, 1996.

[2] S. Ali, Z. Shaukat, M. Azeem, Z. Sakhawat, T. Mahmood, K. ur Rehman: "An efficient and improved scheme for handwritten digit recognition based on convolutional neural network", SN Applied Sciences, Springer Nature Switzerland, vol. 1, art. 1125, pp. 1-9, 2019.

[3] D.C. Ciresan, A. Giusti, L.M. Gambardella, J. Schmidhuber: "Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images", Advances in neural information processing systems, Lake Tahoe, Nevada, United States, 2012.

[4] S. Gavran: "*Veštačke neuronske mreže u istraživanju podataka: pregled i primena*", Univerzitet u Beogradu, Matematički fakultet, Master rad, 2016.

[5] https://github.com/mdbloice/Augmentor (pristupljeno u septembru 2021.)

[6] A. Krizhevsky, I. Sutskever, G.E. Hinton: "Imagenet classification with deep convolutional neural networks", Communications of the ACM, Association for Computing Machinery, vol. 60, issue 6, pp. 84–90, 2017.

[7] S. Hassanpour, N. Tomita, T. DeLise, B. Crosier, L. A. Marsch: "Identifying substance use risk based on deep neural networks and Instagram social media data", Neuropsychopharmacol, Springer Nature, vol. 44, pp. 487–494, 2019.

Kratka biografija:

Dejan Ikonić rođen je 27.07.1991. god. u Sarajevu. Osnovne studije iz oblasti elektrotehnika i računarstvo završio je 2015. na Fakultetu tehničkih nauka u Novom Sadu.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004:007
DOI: https://doi.org/10.24867/16BE32Carapic

MULESOFT PLATFORMA ZA INTEGRACIJU

MULESOFT INTEGRATION PLATFORM

Milana Čarapić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – *U ovom je radu opisana Mulesoft platforma za integraciju sistema. Objašnjena je potreba za korištenjem integracija. Opisane su mogućnosti koje ova platforma nudi u cilju integrisanja sistema. U radu je prikazan i proces kreiranja Mulesoft aplikacije kao i njeno testiranje, odnosno prikazani su dobijeni rezultati.*

Ključne reči: *Integracije sistema, ESB, Mulesoft*

Abstract – *This paper describes the Mulesoft integration platform. The need for integrations is explained. Possibilities that this platform offers in order to integrate different systems are described. The paper also describes the process of creating a Mulesoft application as well as its testing, the obtained results are presented.*

Keywords: *System integration, ESB, Mulesoft*

1. UVOD

U današnje vreme poslovni sistemi ne mogu da funkcionišu izolovano, već postoji potreba za komunikacijom sa drugim sistemima kako bi razmenili potrebne informacije i na taj način uspešno obavili svoju funkciju. Što je sistem složeniji, to mu je potrebno više podataka iz drugih sistema kako bi nesmetano obavljao svoj zadatak. Sistemi između kojih je potrebno uspostaviti komunikaciju mogu da imaju različito definisane formate podataka i protokole i zato njihovo povezivanje predstavlja pravi izazov. Rešenje je integracija sistema.

Integracija predstavlja proces spajanja više različitih sistema u celinu koja funkcioniše kao jedinstveni sistem, putem kojeg se prate i upravlja svim integrisanim sistemima [1]. Integracija dozvoljava komunikaciju, odnosno deljenje informacija između spojenih sistema u cilju izvršavanja zajedničkog zadatka.

2. O INTEGRACIJAMA

2.1. Šabloni za integraciju

Integracioni šablon predstavlja metod komunikacije koji sistemi koriste za slanje i primanje podataka [1].

Neki od integracionih šablona nabrojani su u nastavku:

1. Migracija (data migration) – predstavlja slanje podataka od jednog sistema do drugog.
2. Emitovanje (broadcast pattern) – predstavlja komunikaciju gde jedan sistem šalje podatke većem broju sistema.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Željko Vuković, docent.

3. Agregacija (aggregation pattern) – je kontinualni proces u kojem više sistema šalju podatke jednom sistemu [2].
4. Bidirekciona sinhronizacija (bi-directional sync) – je kontinualna komunikacija između dva ili više integrisanih sistema gde se slanje podataka odvija u oba smera (za razliku od tri predhodna šablona).
5. Korelacija (correlation) – ovaj šablon je veoma sličan bidirekcionoj sinhronizaciji, jer se podaci šalju kontinualno u oba smera. Razlika je u tome što, kod korelacije, šalju se samo oni podaci koji postoje u oba sistema. Podatak koji postoji samo u jednom integrisanom sistemu neće učestvovati u komunikaciji.

2.2. Načini implementacije

Implementacija integracije sistema može da se obavlja na više načina. Jedan od načina jeste "integracija od tačke do tačke" (point-to-point integration), gde se sistemi povezuju posebno svaki sa svakim. Ovaj način je izvodljiv u slučaju kada imamo svega nekoliko sistema koje treba međusobno integrisati. U realnom svetu, broj integrisanih sistema je daleko veći. Svakim dodavanjem nove komponente u integrisani sistem, broj point-to-point veza eksponencijalno raste [3]. Dakle, svako dodavanje/uklanjanje/zamena point-to-point veze u integrisanom sistemu koji sadrži više od 3 komponente predstavlja veoma komplikovan i dug proces sklon greškama.

Još jedan od načina implementacije koji prevazilazi mane point-to-point integracije jeste ESB (Enterprise Servise Bus). ESB je arhitektura čiji je osnovni koncept da se integracija različitih komponenti ostvari postavljanjem komunikacione magistrale između njih, a zatim da se omogući svakoj komponenti da komunicira sa magistralom. ESB je apstraktni sloj koji predstavlja univerzalni prevodilac koji omogućava komunikaciju između više različitih komponenti [4].

3. MULESOFT KAO PLATFORMA ZA INTEGRACIJU

3.1. Mule ESB

Mule ESB je integraciona platforma bazirana na Java programskom jeziku. Omogućava integraciju različitih sistema pomoću komunikacione magistrale koja predstavlja središnji sloj (middleware) putem koje sistemi komuniciraju, odnosno razmenjuju podatke.

Koncept magistrale razdvaja sisteme jedne od drugih. Podaci koji putuju do magistrale su kanonskog formata. Kanonski format poruke znači da postoji jedan dosledan format unapred ugovoren od strane sistema. Korištenje kanonskog modela podataka je odličan način da se strukturiraju poruke koje stižu na magistralu, a to onda

pojednostavljuje implementaciju same magistrale [6]. Postoji "adapter" između aplikacije i magistrale, koji transformiše format podataka koje dolaze od aplikacije u kanonski format magistrale, i obrnuto [5].
U Mule ESB arhitekturi, magistralu, odnosno središnji sloj koji integriše sisteme čine Mule aplikacije. Mulesoft aplikacije se sastoje od komponenti koje zajedno čine flow, a kroz koji se procesiraju Mule dogadjaji, u koje spadaju Mule poruke i Mule promenljive.

4. ALATI ZA RAZVOJ MULE APLIKACIJE

Alati koje možemo koristiti za razvoj Mule aplikacija su Anypoint Studio i Flow Designer.

4.1. Anypoint Studio

Anypoint Studio je desktop integrisano razvojno okruženje za razvoj i testiranje Mule aplikacija, bazirano na Eclipse-u. Ovo je grafičko razvojno okruženje, čija je karakteristika da sve komponente od kojih ćemo razvijati našu Mule aplikaciju su već predefinisane i vizuelno predstavljene ikonicama, dok sam razvoj aplikacije se izvodi drag-and-drop metodom. Odabirom i prevlačenjem komponenti u vizuelni editor gradimo Mule aplikaciju. Prilikom odabira komponente, potrebna je vrlo mala konfiguracija da bismo izabranu komponentu prilagodili potrebama naše aplikacije i da bi komponenta izvršavala svoju funkciju. Pozadinski kod aplikacije je definisan u XML-u.

4.2. Flow Designer

Drugi način za kreiranje Mule aplikacije jeste preko alata Flow Designer. Flow Designer je deo Design Center komponente Anypoint platforme. Za razliku od Anypoint studio-a koji je desktop razvojno okruženje, Flow Designer je web grafičko okruženje.
U odnosu na Anypoint Studio, Flow Designer je veoma ograničen opcijama i alatima za razvoj Mule aplikacija. Flow Designer može nesmetano koristiti sa razvoj jednostavnih aplikacija, ali za kompleksnije aplikacije neophodno je koristiti Anypoint Studio.

5. ARHITEKTURA MULE APLIKACIJE

5.1. Flow

Mule aplikacije obrađuju poruke i druge delove Mule događaja putem komponenata, konektora i modula koji su grupisani unutar jednog toka (flow). Tok (flow) je niz komponenata povezanih međusobno i koji imaju zajedničku funkciju da prime, obrade poruku i proslede rezultat obrade.
Unutar jednog flow-a, redosled komponenti je bitan. Prva komponenta se zove izvor poruke (message source) koja prima poruku od jednog ili više spoljnih izvora. Nakon što prva komponenta primi poruku, to rezultira pokretanjem izvršavanja toka. Zatim slede komponente koje obrađuju pristiglu poruku i oni se zajedno zovu obrađivači poruke (message processors). Poruka putuje od jednog do drugog obrađivača poruke redosledom kojim su definisani. Svaki flow prestavlja smislenu jedinicu rada u celokupnom procesu obrađivanja poruke.

Jednostavna Mule aplikacija može da sadrži samo jedan flow, ali obično aplikacije se sastoje od mnoštva flow-ova koji su podeljeni po ulogama. Svaki flow ima određenu funkciju unutar aplikacije. U toku razvoja aplikacije, kada uočimo da se redosled nekih obrađivača poruke ponavlja u više flow-ova, možemo ih izdvojiti u posebnu jedinicu koja se naziva subflow. Na taj način izbegavamo ponavljanje istog koda, već ga izdvojimo u posebnu jedinicu i onda on može biti korišten više puta od strane više flow-ova.

5.2. Struktura Mule poruke

Kada poruka stigne i putuje kroz flow, ona se javlja u formi događaja. Mule događaji (Mule events) sadrže osnovne informacije koje se obrađuju u Mule aplikaciji.
Mule događaj se sastoji iz sledeće dve komponente:

- Mule poruke (mule message) – to su podaci koji će se koristiti za izvršavanje flow-a unutar aplikacije.
- Mule promenljive (mule variables) – to su metapodaci Mule događaja koji se koriste unutar flow-a.

Mule poruka (Mule message) je deo Mule događaja i predstavlja podatke iz spoljašnjih sistema koji se obrađuju kroz flow unutar Mule aplikacije.
Mule poruka se sastoji iz dva dela:

- Sadržaj poruke (payload) – sadrži telo poruke. To može biti sadržaj fajla, zapis iz baze podataka, odgovor na REST ili Web service zahtev itd. Sadržaj poruke se menja kroz flow u zavisnosti od obrađivača poruke koji se nalaze u tom flow-u.
- Atributi (attributes) – metapodaci koji povezani sa sadržajem poruke. Specifični atributi poruke zavise od izvora poruke, konektora koji dobavlja poruku u flow. Atributi mogu sadržati zaglavlja (headers) i svojstva koji prima ili vraća konektor, kao i druge metapodatke koji su popunjeni od strane konektora.

Mule promenljive (Mule variables) su korisnički definisani metapodaci o poruci. Promenljive se koriste za privremeno čuvanje informacija o poruci koju aplikacija procesuira. Promenljive neće biti prosleđene na odredište zajedno sa porukom.

6. DATAWEAVE JEZIK ZA TRANSFORMACIJU PODATAKA

Transformacija podataka je proces konvertovanja podataka iz jednog formata u drugi. U toku integracija sistema, ovaj proces je veoma bitan jer obično kada se podaci šalju iz izvornog sistema u odredišni sistem su različitog formata i neophodno je transformisati ih u odgovarajući format pre slanja na odredište. Konkretno, u Mule integraciji, podaci iz izvornog sistema prvo stižu do Mule magistrale koja je sačinjena od Mule aplikacija, pa tek onda na odredište. Kako smo već rekli da magistrala ima definisan kanonski format, može se desiti izvorni sistem šalje podatke u drugačijem formatu od kanonskog formata magistrale. Zato postoje „adapteri" koji transformišu podatke čim oni pristignu na magistralu.

Za pristup i transformaciju podataka, Mulesoft je razvio svoj jezik pod nazivom DataWeave, i koji se koristi unutar Mule aplikacija. U pitanju je jezik izraza (expression language) i koji podržava koncepte koji su zajednički većini programskih jezika.

7. KOMPONENTE MULE APLIKACIJE

Komponente su osnovne gradivne jedinice flow-a unutar Mule aplikacije. Komponente pružaju logiku obrade poruke koja prolazi kroz flow aplikacije. Komponente su podeljene u dva kategorije: Osnovne komponente (Core) i

konektori. Osnovne komponente su one komponente koje razvijaju internu logiku unutar flow-a, dok konektori omogućavaju eksternu komunikaciju sa resursima izvan flow-a.

Konektor je softver koji omogućava konekciju između Mule flow-a i eksternog resursa. Resurs može biti bilo kojeg tipa: baza podataka, protokol, API.
Pomoću konektora vršimo konekcije sa:

- Softverskim aplikacijama - Koristimo konektore da povežemo Mule aplikaciju sa specifičnim softverom i vršimo akcije nad konektovanom aplikacijom.
- Bazama podataka – povezujemo Mule aplikaciju sa jednom ili više baza podataka i vršimo akcije nad konektovanom bazom.
- Protokolima – koristimo konektore da šaljemo i primamo podatke nad protokolima.

7.1. Message Source komponente

Message source komponente su komponente koje se nalaze na početku flow-a, primaju poruku od jednog ili više spoljnih resursa i iniciraju izvršavanje flow-a. Najčešći konektori koji se koriste za pokretanje flow-a su HTTP Listener i Scheduler.

7.1. Transformatori

Transformatori (transformers) su komponente koje omogućavaju transformaciju podataka u bilo koji format ili strukturu koja nam je potrebna. Tranformator prima određeni sadržaj poruke i priprema je za dalju obradu menjajući njen sadržaj. Osim poruke, transformeri omogućavaju i transformacije promenljivih.

Komponente koje manipulišu promenljivim unutar Mule događaja su:

- Set Variable – služi za kreiranje promenljive ili modifikovanje vrednosti već postojeće promenljive. Kao vrednost promenljive možemo staviti string ili poruku, sadržaj poruke, ili atribute objekta. Takođe, vrednost promenljive može biti i DataWeave izraz.
- Remove Variable – briše promenljivu iz Mule događaja. Ukoliko u toku obrade Mule događaja primetimo da neka promenljiva nam više neće koristiti, možemo je lako obrisati iz Mule događaja korištenjem ove komponente.

Transform Message komponenta konvertuje ulazne podatke u novu izlaznu strukturu ili format. Kod ove komponente postoje dve prezentacije iste transformacije: grafički editor i editor za kodiranje. U grafičkom editoru se nalaze dva stabla koji predstavljaju očekivane ulazne i izlazne strukture metapodataka. Metapodatke ova komponenta obradi i prikaže njihovu strukturu u grafičkom pogledu, kako bi nam bilo lakše da mapiramo podatke drag-and-drop tehnikom. Grafički editor olakšava posao ukoliko nam je potrebna jednostavna transformacija kao što je mapiranje podataka. Složenije transformacije ne možemo postići u grafičkom editoru. Svaka izmena koju napravimo u grafičkom editoru će biti prikazana i u editoru za kodiranje kao DataWeave kod. Složenije transformacije se rade u ovom editoru pomoću DataWeave izraza.

7.2. Kontrola toka

Kontrola toka u Mule aplikacijama se postiže komponentama koje se zovu Ruteri (Routers). Kao što im i sam naziv kaže, ove komponente usmeravaju poruke na različite destinacije unutar flow-a. Pomoću ovih komponenti moguće je:

- Podeliti poruku na nekoliko segmenata, a zatim svaki preusmeriti na drugi procesor.
- Kombinovati nekoliko poruka u jednu poruku pre nego što se pošalje sledećem procesoru u flow-u.
- Preurediti listu poruka pre slanja sledećem procesoru.
- Odrediti na koji procesor od nekoliko mogućih procesora će poruka biti preusmerena.
- Poslati jednu poruku na više različitih procesora.

U rutere spadaju komponente Choice, su Scatter-Gather i First Successful.

7.3. Rukovanje greškama i izuzecima

Kada neka aktivnost u Mule aplikaciji se ne izvrši uspešno, odnosno pojavi se greška, Mule izbaci izuzetak. Za upravljanje ovim izuzecima, Mule nam nudi nekoliko komponenti koje se zovu Error Handlers, i pomoću kojih možemo da definišemo strategiju ovih izuzetaka.
Rukovanje greškama može da se obavi na više različitih nivoa. Možemo rukovati greškama na nivou aplikacije tako što definišemo globalne rukovaoce greškama. Moguće je definisati rukovanje greškama na nivou flow-a, ili čak za blok manji od jednog flow-a (custom error handlers).

Postoje dve strategije za rukovanje greškama: On Error Propagate i On Error Continue. Ove dve strategije se odnose na način na koji ćemo grešku koja se javila unutar subflow-a proslediti flow-u koji je pozvao izvršenje tog subflow-a. On Error Continue je strategija koja će registrovati grešku unutar subflow-a, vratiće odgovor o grešci flow-u koji ga je pozvao ali sa statusnim kodom 200 – to će omogućiti da se nastavak flow-a izvrši. On Error Continue je strategija koja će registrovati grešku unutar flow-a, vratiće odgovor o grešci flow-u koji ga je pozvao ali sa statusnim kodom 500 – to će obustaviti dalje izvršavanje flow-a.

7.4. Rad sa bazama podataka

To su konektori koji uspostavljaju vezu između Mule aplikacije i relacione baze podataka. Ovi konektori mogu da uspostave vezu sa skoro svakom JDBC relacionom bazom podataka i da vrše SQL operacije. Konektori nam omogućavaju da koristimo SQL operacije nad bazom podataka, kao što su Select, Insert, Update, Delete, Stored Procedured itd. Pre upotrebe bilo kojeg konektora za rad sa bazama podataka, potrebno je prvo napraviti konfiguraciju baze podaka. Podaci neophodni za uspostavljanje konekcije su host, port, username i password.

7.5. Rad sa datotekama

Mule podržava efikasnu obradu velikih podataka poput datoteka, dokumenata i zapisa. To je omogućeno konektorima za rad sa datotekama.

Kada komponenta koristi datoteku, Mule čuva njen sadržaj u privremenom baferu. Paralelno korištenje jedne datoteke od strane više komponenti je dozvoljeno, što znači da više komponenti mogu da čitaju istu datoteku u isto vreme. Mule automatski osigurava da kada jedna komponenta čita datoteku, to ne uzrokuje nikakve neželjene efekte za drugu komponentu.

Konektore možemo podeliti na dve grupe: prva grupa obrađuje datoteke na lokalnom sistemu (local files), a druga grupa konektora obrađuje datoteke na udaljenim sistemima (remote files).
Sa lokalnim datotekama rade File konektori. Sa datotekama na udaljenim sistema rade FTP, SFTP, FTPS, MFT konektori.
FTP konektori omogućavaju pristup fajlovima i folderima koji se nalaze na FTP (File Transfer Protocol) serveru.
SFTP omogućavaju pristup fajlovima i folderima koji se nalaze na SFTP (Secure File Transfer Protocol) serveru.

7.6. HTTP Konektori

Slanje i primanje poruka preko HTTP protokola je na Mule platformi omogućeno ugrađenim HTTP konektorima. HTTP konektori nam nude opciju deklarisanja servera koji osluškuje zahteve i pokreće izvršavanje Mule aplikacije kao i HTTP klijente koji mogu komunicirati sa bilo kojim HTTP servisom [7]. Za sve tipove ovih konektora potrebno je definisati konekciju pre upotrebe. Konekcija definiše host i port gde će server biti podešen, kao i protokol: HTTP za obične konekcije ili HTTPS za TLS konekcije.

8. TESTIRANJE MULE APLIKACIJE

Testiranje softvera koristi se da bi se osiguralo da se očekivani poslovni sistemi i karakteristike proizvoda ponašaju ispravno i prema očekivanju [8].
Unit testing (jedinično testiranje) je osnovni nivo testiranja softvera. Unit testing je praksa provere ispravnosti izlaznih vrednosti (outputs) na osnovu automatizovanih ulaznih vrednosti (inputs) za određene segmente/jedinice koda.

Za Mule aplikacije, unit testovi se prave u MUnit-u, komponenti Anypoint platforme. MUnit je framework za testiranje Mule aplikacija koji pravi unit automatizovane testove za naše aplikacije. MUnit je integrisan u Anypoint Studio.

Prilikom pravljenja novog projekta u Anypoint studio-u, folder na putanji src/test/munit se automatski kreira.

Prilikom kreiranja MUnit fajlova, njihova lokacija će biti upravo ovaj folder. Pošto smo rekli da se jedan Unit test pravi za jednu jedinicu koda, a u Mule aplikaciji to je flow, znači da za jedan flow pravimo jedan Unit test.

9. IMPLEMENTACIJA STUDENTS-APP APLIKACIJE NA MULE PLATFORMI

Za potrebe ovog rada kreirana je Mulesoft aplikacija u Anypoint Studio-u pod nazivom students-app, koja predstavlja primer korištenja tehnologije. Aplikacija konzumira podatke iz MySQL baze podataka, manipuliše pomenutim podacima i šalje ih na dva odredišta: lokalni fajl i email adresu preko SMTP servera. Realizovane su funkcije za prikaz, ažuriranje i brisanje podataka o studentu iz sistema, dodavanje novog studenta u sistem, kao i prikaz podataka svih studenata u sistemu uz mogućnost filtriranja dobijenih rezultata.

U ovom poglavlju u radu je detaljno prikazan razvoj aplikacije kao i dobijeni rezultati prilikom njenog testiranja.

10. ZAKLJUČAK

U ovom radu je predstavljena Mulesoft platforma za integraciju. Tehnologija je detaljno opisana kroz poglavlja u kojima su navedene mogućnosti Mulesoft platforme u cilju integrisanja sistema.

Mulesoft integraciona platforma poseduje širok spektar modula i ugrađenih konektora koji olakšavaju proces integracije. Veoma mala konfiguracija je dovoljna da se ugrađeni konektor prilagodi potrebama specifičnog sistema i osposobi za korištenje. Mulesoft kompanija prati trenutnu situaciju na tržištu, te kroz nove verzije softvera redovno kreira nove konektore koji su dostupni u Anypint Exchange-u, gde ih je lako pronaći i dodati u svoje razvojno okruženje.

Ovaj rad je obuhvatio osnovne pojmove i uvod u integracionu platformu, kao i načine za kreiranje i testiranje integracionog rešenja (aplikacije) na specifičnoj platformi.

11. LITERATURA

[1] https://www.mulesoft.com/resources/integration (pristupljeno u junu 2021.)
[2] https://www.mulesoft.com/resources/esb/top-five-data-integration-patterns (pristupljeno u junu 2021.)
[3] https://www.mulesoft.com/resources/esb/eliminating-point-point-integration-pain-mule-esb-use-cases (pristupljeno u junu 2021.)
[4] https://blog.dreamfactory.com/esb-and-api-management-which-one-and-why/ (pristupljeno u junu 2021.)
[5] https://www.mulesoft.com/resources/esb/what-esb (pristupljeno u junu 2021.)
[6] David Dossot, John D'Emic, Victor Romero , "*Mule in Action, Second Edition",* Shelter Island, Manning, 2014.
[7] https://docs.mulesoft.com/http-connector/1.5/ (pristupljeno u junu 2021.)
[8] https://cubes.edu.rs/sr/20/obuke-i-kursevi/sta-je-software-qa-ili-testiranje-software-a (pristupljeno u junu 2021.)

Kratka biografija:

Milana Čarapić rođena je 17.11.1995. godine u Užicu, Srbija. Završila je gimnaziju "Ivo Andrić" u Višegradu, Bosna i Hercegovina. Školske 2014/2015 godine upisuje osnovne akademske studije prvog stepena na Fakultetu tehničkih nauka u Novom Sadu, studijski program Računarstvo i automatika, usmerenje Primenjene računarske nauike i informatika. Dana 05.10.2018. godine diplomirala je sa prosečnom ocenom 8,78 (osam i 78/100). Diplomirala je na temu "Poređenje programskih jezika C++ i C#". Školske 2018/2019 godine upisuje master studije na Fakultetu tehničkih nauka u Novom Sadu, studijski program Računarstvo i automatika, usmerenje Elektronsko poslovanje.

kontakt: mcarapic95@gmail.com

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/16BE34Radojcic

VIZUELNO PRETRAŽIVANJE PODATAKA

VISUAL DATA MINING

Tanja Radojčić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Ovaj rad daje uvid u podatke i tehnike za vizuelizaciju podataka u cilju lakšeg korišćenja velikih količina podataka. Primeri pokazuju koliko vizuelni prikaz može olakšati rad čoveka u svakodnevnom životu. Kombinuju se čovek i algoritam za najbolji mogući ishod.*

Ključne reči: *vizuelizacija, pretraživanje podataka, velika količina podataka, tehnike vizuelizacije*

Abstract – *This paper provides insight into data and data visualization techniques to facilitate the use of large amounts of data. Examples show how much visual representation can facilitate a person's work in everyday life. Man and algorithm are combined for the best possible outcome.*

Keywords: *visualization, data search, big data, visualization techniques*

1. UVOD

Napretkom tehnologije u 21. veku došli smo do toga da današnji kompjuteri mogu da skladište ogromne količine podataka. Od početka razvoja kompjuterske tehnologije, nije bilo predviđeno da će godišnji unos novih informacija prelaziti čak 1 milion terabajta. Kako svakim danom tehnologija napreduje, kao i potreba za većim pohranjivanjem informacija, dolazimo do procene da će u naredne tri godine biti generisano više podataka nego što je ukupno od postojanja čovečanstva.

Prvo poglavlje je uvod u vizuelno pretraživanje podataka i sam rad. Drugo poglavlje je bazirano na osnovnim informacijama vezanim za vizuelno pretraživanje podataka. Treće poglavlje predstavlja opis svih vrsta podataka koji se koriste za vizuelizaciju. Četvrto poglavlje je posvećeno tehnikama vizuelizacije i njihovim reprezentacijama.

Peto poglavlje objašnjava tehnike interakcije i izobličenja, koje uz tehnike vizuelizacije služe za njihov detaljniji prikaz. Šesto poglavlje se osvrće na praktičnu primenu i opisani su primeri iz života sa dostupnim bazama podataka. Zaključak je pregled uvida stečen tokom pisanja rada.

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

2. TEORIJSKE OSNOVE

Vizuelno pretraživanje podataka podrazumeva prikazivanje podataka vizuelno kroz grafike, tabele i slično, pomoću kojih čovek dolazi do određenih zaključaka i povezuje date informacije. Postoje razne tehnike koje su se pokazale krajnje efikasno u ovim pretragama, te algoritmi pretraživanja zasnovani na njima mogu da pretraže ogromne količine podataka.

Kako je čovek uključen u sam proces, zaključci i ciljevi se lako menjaju ukoliko je to potrebno. Određene zaključke i veze između podataka može doneti softver, koje čovek može lako da previdi.

2.1. Proces vizuelnog pretraživanja

Prvo, korisnici dobijaju širok pregled svih unesenih podataka. U pregledu podataka, korisnici pronalaze zanimljive obrasce i fokusiraju se na neke od njih. Za njhovu analizu, moraju da zalaze duboko u podatke kako bi dobili detaljniji prikaz. Drugo, zumiranje različitih grupa podataka unutar vizualizovanih podataka će omogućiti da se filtriraju različite vrednosti i kreiraju obrasce pre nego što se dođe do zaključka zbog kog je proces i započet.

Tehnologije vizuelizacije se mogu koristiti za sva tri koraka, tako da je dobro koristiti jednu tehniku za prvi korak i prikazivanje podataka od interesa, dok se neka druga tehnika koristi za fokusiranje na deo tih podataka, detaljnije i opširnije. Vizuelizacijom možemo prikazati i veze između ta tri koraka, ne samo konkretno njih.

2.2. Klasifikacija tehnika

Za tri dimenzije klasifikacije - tip podataka koji se vizualizuje, tehnika vizualizacije i tehnika interakcije i izobličenja - može se pretpostaviti da su ortogonalne. Ortogonalnost znači da se bilo koja od tehnika vizualizacije može koristiti zajedno sa bilo kojom od tehnika interakcije, kao i bilo kojom od tehnika izobličenja za bilo koji tip podataka.

Određeni sistem može biti dizajniran da podržava različite tipove podataka i da može koristiti kombinaciju više tehnika vizualizacije i interakcije.

3. TIPOVI PODATAKA

U vizualizaciji informacija, podaci se obično sastoje od velikog broja zapisa od kojih se svaki sastoji od niza promenljivih ili dimenzija. Svaki zapis odgovara opservaciji, merenju, ili transakciji.

3.1. Jednodimenzionalni podaci

Jednodimenzionalni podaci obično imaju jednu gustu dimenziju. Tipičan primer jednodimenzionalnih podataka je vremenski podatak, vreme ili novac. Imajte na umu da se sa svakom tačkom vremena može povezati jedna ili više vrednosti podataka.

3.2. Dvodimenzionalni podaci

Dvodimenzionalni podaci imaju dve različite dimenzije. Tipičan primer su geografski podaci gde su dve različite dimenzije geografska dužina i širina. X-Y-grafikoni su tipična metoda za prikazivanje dvodimenzionalnih podataka, a mape su posebna vrsta X-Y -grafikona za prikazivanje dvodimenzionalnih geografskih podataka.

3.3. Višedimenzionalni podaci

Mnogi skupovi podataka sastoje se od više od tri atributa i stoga ne dopuštaju jednostavnu vizualizaciju kao dvodimenzionalni ili trodimenzionalni grafikoni. Primeri višedimenzionalnih (ili viševarijantnih) podataka su tabele iz relacionih baza podataka, koje često imaju desetine do stotine kolona (ili atributa). Pošto ne postoji jednostavno mapiranje atributa u dve dimenzije ekrana, potrebne su sofisticiranije tehnike vizualizacije. Primer tehnike koja omogućava vizualizaciju višedimenzionalnih podataka je tehnika paralelnih koordinata. Paralelne koordinate prikazuju svaku višedimenzionalnu stavku podataka kao poligonalna linija koja preseca horizontalne ose dimenzija na položaju koji odgovara vrednosti podataka za odgovarajuću dimenziju.

3.4. Tekst i hipertekst

Ne mogu se svi tipovi podataka opisati u smislu dimenzionalnosti. U doba svetske mreže, jedan važan tip podataka je tekst i hipertekst, kao i sadržaj multimedijalnih veb stranica. Ovi tipovi podataka se razlikuju po tome što se ne mogu lako opisati brojevima, pa se većina standardnih tehnika vizualizacije ne može primeniti. U većini slučajeva, prvo je potrebna transformacija podataka u vektore opisa pre nego što se mogu koristiti tehnike vizualizacije. Primer za jednostavnu transformaciju je brojanje reči koje se često kombinuje sa analizom glavnih komponenti ili višedimenzionalnim skaliranjem.

3.5. Algoritmi i softver

Zapisi podataka često imaju neku vezu sa drugim podacima, oslanjaju se na sadržaj objavljen na vebu. Grafikoni se široko koriste za predstavljanje takvih međuzavisnosti. Grafikon se sastoji od skupa objekata koji se nazivaju čvorovi i veza između ovih objekata koji se nazivaju ivice. Primeri su međusobni odnosi e-pošte među ljudima, njihovo ponašanje pri kupovini, struktura datoteka na čvrstom disku ili hiperveze na svetskoj mreži. Postoji niz specifičnih tehnika vizualizacije koje se bave hijerarhijskim i grafičkim podacima.

4. TEHNIKE VIZUELIZACIJE

Postoji veliki broj tehnika vizualizacije koje se mogu koristiti za vizualizaciju podataka. Pored standardnih 2D/3D tehnika, kao što su X-Y (x-y-z) grafikoni, trakasti grafikoni, linijski grafikoni itd., Postoji niz sofisticiranijih tehnika vizualizacije. Časovi odgovaraju osnovnim principima vizualizacije koji se mogu kombinovati radi implementacije određenog sistema vizualizacije [2].

4.1. Standardni 2D/3D prikaz

Najčešće korišćen prikaz. Uglavnom predstavlja krafikone, kockaste prikaze grafikona u 3 dimenzije.

4.2. Geometrijski transformisan prikaz

Geometrijski transformisane tehnike prikaza imaju za cilj pronalaženje „zanimljivih“ transformacija višedimenzionalnih skupova podataka. Klasa tehnika geometrijskog prikaza uključuje tehnike iz istraživačke statistike kao što su matrice raspršenog grafikona i tehnike koje se mogu podvesti pod izraz „traženje projekcije“. Druge tehnike geometrijskog projektovanja uključuju tužilačke poglede (eng. Prosection Views), Hyperslice i dobro poznatu tehniku vizualizacije paralelnih koordinata. Tehnika paralelnih koordinata preslikava k-dimenzionalni prostor u dve dimenzije ekrana korišćenjem k ekvivalentnih osa koje su paralelne sa jednom od osa prikaza. Osi odgovaraju dimenzijama i linearno su skalirane od minimalne do maksimalne vrednosti odgovarajuće dimenzije. Svaka stavka podataka predstavljena je kao poligonalna linija, koja seče svaku osu u onoj tački koja odgovara vrednosti razmatranih dimenzija.

4.3. Prikaz zasnovan na ikonama

Druga klasa tehnika istraživanja vizuelnih podataka su ikonične tehnike prikaza. Ideja je mapiranje vrednosti atributa višedimenzionalne stavke podataka sa karakteristikama ikone. Ikone se mogu proizvoljno definisati: To mogu biti mala lica, ikone zvezda, ikone u obliku štapića, ikone u boji i TileBars (Slika 7). Vizualizacija se generiše preslikavanjem vrednosti atributa svakog zapisa podataka na karakteristike ikona. U slučaju tehnike štapne figure, na primer, dve dimenzije se mapiraju u dimenzije ekrana, a preostale dimenzije u uglove i/ili dužinu ekstremiteta ikone figure štapa. Ako su stavke podataka relativno guste u odnosu na dve dimenzije ekrana, rezultujuća vizualizacija predstavlja obrasce teksture koji se razlikuju u skladu sa karakteristikama podataka i stoga se mogu otkriti predpažnjom.

4.4. Prikaz sa gustim pikselima

Osnovna ideja tehnika gustog piksela je mapiranje vrednosti svake dimenzije u obojeni piksel i grupisanje piksela koji pripadaju svakoj dimenziji u susedna područja. Pošto generalno ekrani sa gustim pikselima koriste jedan piksel po vrednosti podataka, tehnike omogućavaju vizualizaciju najveće moguće količine podataka na trenutnim ekranima (do oko 1.000.000 vrednosti podataka). Ako je svaka vrednost podataka predstavljena jednim pikselom, glavno pitanje je kako rasporediti piksele na ekranu. Tehnike gustog piksela koriste različite aranžmane u različite svrhe. Raspoređivanjem piksela na odgovarajući način, rezultirajuća vizualizacija pruža detaljne informacije o lokalnim korelacijama, zavisnostima i žarišnim tačkama.

4.5. Naslagani prikaz

Tehnike naslaganog prikaza prilagođene su za hijerarhijski prikaz podataka podeljenih na particije. U slučaju višedimenzionalnih podataka, dimenzije podataka koje će se koristiti za particioniranje podataka i izgradnju hijerarhije moraju biti odgovarajuće odabrane. Primer tehnike naslaganog prikaza je Dimenzijsko slaganje.

5. TEHNIKE INTERAKCIJE I IZOBLIČENJA

Pored tehnike vizualizacije, za efikasno istraživanje podataka potrebno je koristiti i neke tehnike interakcije i izobličenja. Tehnike interakcije omogućavaju analitičaru podataka da direktno stupi u interakciju sa vizualizacijama i dinamički menja vizualizacije u skladu sa ciljevima istraživanja, a takođe omogućavaju povezivanje i kombinovanje više nezavisnih vizualizacija. Tehnike izobličenja pomažu u procesu istraživanja podataka pružajući sredstva za fokusiranje na detalje uz očuvanje pregleda podataka. Osnovna ideja tehnika izobličenja je prikazati delove podataka sa visokim nivoom detalja, dok su drugi prikazani sa nižim nivoom detalja. Razlikujemo termine dinamički i interaktivni u zavisnosti od toga da li se promene vizualizacija vrše automatski ili ručno (direktnom interakcijom korisnika).

5.1. Dinamička projekcija

Osnovna ideja dinamičkih projekcija je dinamička promena projekcija radi istraživanja višedimenzionalnog skupa podataka. Broj mogućih projekcija je eksponencijalan u broju dimenzija, to jest nerešiv je za velike dimenzionalnosti. Prikazane sekvence projekcija mogu biti nasumične, ručne, unapred izračunate ili zasnovane na podacima

5.2. Interaktivno filtriranje

U istraživanju velikih skupova podataka važno je interaktivno podeliti skup podataka na segmente i fokusirati se na zanimljive podskupove. Ovo se može uraditi direktnim izborom željenog podskupa (pregledavanjem) ili specifikacijom svojstava željenog podskupa (upiti). Pretraživanje je veoma teško za veoma velike skupove podataka i upiti često ne daju željene rezultate. Stoga su razvijene brojne tehnike interakcije za poboljšanje interaktivnog filtriranja u istraživanju podataka.

5.3. Interaktivno zumiranje

Zumiranje je dobro poznata tehnika koja se široko koristi u brojnim aplikacijama. U radu sa velikim količinama podataka, važno je predstaviti podatke u visoko sabijenom obliku kako bi se obezbedio pregled podataka, ali istovremeno omogućio promenljiv prikaz podataka u različitim rezolucijama. Zumiranje ne znači samo da se objekti podataka prikazuju veći, već takođe znači da se prikaz podataka automatski menja kako bi predstavio više detalja o višim nivoima zumiranja. Objekti mogu, na primer, biti predstavljeni kao pojedinačni pikseli na niskom nivou zumiranja, kao ikone na srednjem nivou zumiranja i kao označeni objekti u visokoj rezoluciji. Zanimljiv primer primene ideje zumiranja na velike skupove tabelarnih podataka je pristup TableLens.

5.4. Interaktivno krivljenje

Interaktivne tehnike izobličenja podržavaju proces istraživanja podataka čuvajući pregled podataka tokom operacija bušenja. Osnovna ideja je prikazati delove podataka sa visokim nivoom detalja, dok su drugi prikazani sa nižim nivoom detalja. Popularne tehnike izobličenja su hiperbolične i sferne distorzije koje se često koriste na hijerarhijama ili grafikonima, ali se mogu primeniti i na bilo koju drugu tehniku vizualizacije.

5.5. Interaktivno povezivanje i četkanje

Postoji mnogo mogućnosti za vizualizaciju višedimenzionalnih podataka, ali svi oni imaju određenu snagu i neke slabosti. Ideja povezivanja i četkanja je kombinovanje različitih metoda vizualizacije kako bi se prevazišli nedostaci pojedinih tehnika. Na primer, tabele različitih projekcija mogu se kombinovati bojenjem i povezivanjem podskupa tačaka u svim projekcijama. Na sličan način, povezivanje i četkanje se mogu primeniti na vizualizacije generisane svim gore opisanim tehnikama vizuelizacije. Kao rezultat toga, brušene tačke su istaknute u svim vizualizacijama, što omogućava otkrivanje zavisnosti i korelacija. Interaktivne promene napravljene u jednoj vizualizaciji automatski se odražavaju u drugim vizualizacijama.

6. PRIMENA SA BIG DATA PODACIMA

Pretraživanje podataka je metoda analize velikih količina podataka u nastojanju da se otkriju odnosi, tj. Veze između podataka. Ove veze se slažu sa definicijom da moraju biti smisleni jer vode do nekoliko prednosti, najčešće finansijske. Podaci su obično kvantitativni, posebno ako se uzme u obzir eksponencijalni razvoj podataka odnosno big data. Vizuelno pretraživanje podataka je prikaz podataka u okviru grafikona, slike i slično. Oni su napravljeni kao vizuelni prikaz informacija.

7. PRAKTIČNA PRIMENA

Ova studija slučaja pretraživanja vizuelnih podataka zasnovana je na otvorenim podacima koje je objavio Centar za životnu sredinu Helsinkija.

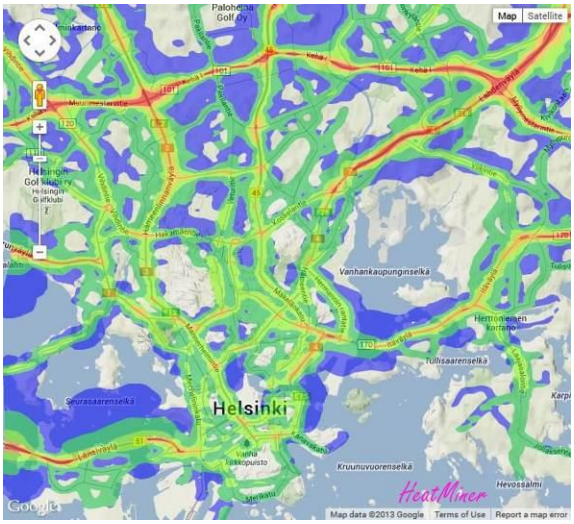
Originalni skup podataka dostupan je na zvaničnoj stranici regije Helsinki i sadrži detaljne informacije o nivoima gradske buke. Detaljan opis podataka i povezanih istraživanja mogu se pronaći iz zbirnog izveštaja koji je dostupan na veb stranicama projekta studije buke (uglavnom na finskom) [4].

Skup podataka o buci odličan je primer otvorenih podataka koji bi mogli biti vrlo vredni običnim ljudima. Na primer, porodice koje traže miran kraj u Helsinkiju verovatno bi želele da provere nivo buke u saobraćaju pre nego što kupe kuću. Nažalost, dostupne informacije su date u prilično teškom formatu za brzi pregled. Zato je napravljen ovaj primer sa toplotnim mapama kako bi na google mapu izgenerisali date podatke i njihovom vizuelizacijom pojednostavili prikaz.

Toplotna karta sa slike 1 prikazuje prosečne nivoe saobraćajne buke (LAeq) tokom dana (između 07č-22č) dok slika 2 prikazuje prosečan nivo buke tokom noći.

Boje toplotne karte ukazuju na prosečne nivoe buke, tako da su crvena područja najbučnija.

Slika 1. *Saobraćajna buka tokom dana (07č-22č)* [5]



Slika 2. *Saobraćajna buka tokom noći (22č-07č)* [5]

Nakon prikaza obe mape, preko dana i preko noći, možemo zaključiti sledeće: Saobraćaj u noćnim satima je manje bučan (što je očekivano), imamo mrežu glavnih saobraćajnica koje su preko dana veoma bučne i delove koje su čak i preko dana u najmirnijoj boji – plavoj. Ako se vratimo na primer sa početka, u kom očekujemo da bi ova mapa koristila ljudima koji biraju nekretninu u mirnijem delu, oni bez domenskog znanja mogu da prepoznaju delove grada i tako dođu do željene nekretnine, uz pomoć vizualnog pretraživanja podatka.

8. ZAKLJUČAK

Istraživanje velikih skupova podataka važan je, ali težak problem. Tehnike vizualizacije informacija mogu pomoći u rešavanju problema. Istraživanje vizuelnih podataka ima veliki potencijal i mnoge aplikacije, poput otkrivanja prevara i pretraživanja podataka, koristiće tehnologiju vizualizacije informacija za poboljšanu analizu podataka. Budući rad će uključivati usku integraciju tehnika vizualizacije sa tradicionalnim tehnikama iz disciplina kao što su statistika, mašinsko učenje, istraživanje operacija i simulacija. Integracija tehnika vizualizacije i ovih ustaljenijih metoda kombinovala bi brze algoritme automatskog rudarenja podataka sa intuitivnom snagom ljudskog uma, poboljšavajući kvalitet i brzinu procesa iskopavanja vizuelnih podataka. Viusalne tehnike rudarenja podataka takođe moraju biti čvrsto integrisane sa sistemima koji se koriste za upravljanje ogromnom količinom relacionih i polustrukturiranih informacija, uključujući upravljanje bazama podataka i sisteme skladišta podataka. Krajnji cilj je unositi moć tehnologije vizualizacije na svaku radnu površinu kako bi se omogućilo bolje, brže i intuitivnije istraživanje veoma velikih resursa podataka. Ovo neće biti samo vredno u ekonomskom smislu, već će i stimulisati i oduševiti korisnika.

9. LITERATURA

[1] D.A.Keim, „IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS“, 2002.
[2] D. A. KEIM, „Visual Data-Mining Techniques“. 2002.
[3] http://cloudnsci.fi/wiki/index.php?n=HeatMiner.RateOfIndebtedness (pristupljeno u oktobru 2021.)
[4] http://cloudnsci.fi/wiki/index.php?n=HeatMiner.TraficNoiseInHelsinki (pristupljeno u oktobru 2021.)
[5] http://cloudnsci.fi/wiki/index.php?n=HeatMiner.TraficAccidentsInHelsinki (pristupljeno u oktobru 2021.)
[6] http://cloudnsci.fi/wiki/index.php?n=HeatMiner.DeerCrashesInFinland (pristupljeno u oktobru 2021.)
[7] http://cloudnsci.fi/wiki/index.php?n=HeatMiner.VisitorLocationHeatmaps (pristupljeno u oktobru 2021.)
[8] E. Karn, „Visualization of Real Time Data Driven Systems using D3 Visualization Technique“.
[9] D. Asimov, “The grand tour: A tool for viewing multidimensional data“, 1985.
[10] S. J. Simoff1, M. H. Böhlen, and A. Mazeika, „Visual Data Mining: An Introduction and Overview“ ,2007.

Kratka biografija:

Tanja Radojčić je rođena 29.04.1996. godine u Novom Sadu. Završila je srednju školu gimnaziju „Isidora Sekulić“ u Novom Sadu 2015. godine. Fakultet tehničih nauka u Novom Sadu je upisala 2015. godine. Završila je osnovne akademske studije na Fakultetu tehničkih nauka 2019. godine. Završila je master akademske studije na Fakultetu tehničkih nauka 2021. godine, smer Primenjeno softversko inženjerstvo sa prosekom 9,20.

Зборник радова Факултета техничких наука, Нови Сад

UDK: 007:004.056
DOI: https://doi.org/10.24867/16BE35Milovanovic

ОБЕЗБЕЂИВАЊЕ ИЗВОРНОГ КОДА КОЈИ ОБРАЂУЈЕ ПЛАТФОРМА ЗА АНАЛИЗУ КВАЛИТЕТА КОДА

SECURING SOURCE CODE PROCESSED BY A CODE QUALITY ANALYSIS PLATFORM

Милан Миловановић, *Факултет техничких наука, Нови Сад*

Област – СОФТВЕРСКО ИНЖЕЊЕРСТВО И ИНФОРМАЦИОНЕ ТЕХНОЛОГИЈЕ

Кратак садржај – *У раду је приказан модел претњи за платформу за анализу изворног кода. Такође је представљен дизајн за безбедно унапређење платформе који укључује обфускацију, употребу дигиталних сертификата и виртуелне приватне мреже.*

Кључне речи: *моделовање претњи, безбедност, поверљивост, обфускација*

Abstract – *The paper describes a threat model of a platform designed to analyze source code. The design for secure platform enhancement that includes obfuscation, the use of digital certificates, and a virtual private network was presented.*

Keywords: *threat modeling, security, confidentiality, obfuscation*

1. УВОД

У данашње време, изворни код као интелектуална својина представља најбитнију имовину многих организација. Доспеће пословних тајни у погрешне руке може изазвати огромну материјалну штету. Из тог разлога је заштита поверљивости корисничких података кључан аспект који треба размотрити при изради софтвера за анализу изворног кода. У овом раду ће, са аспекта безбедности, бити извршена анализа софтверског решења за анализу квалитета изворног кода – Clean CaDET платформе. Акценат је стављен на заштиту изворног кода који платформа обрађује.

Следеће поглавље ће бити посвећено безбедносној анализи дизајна и моделовању претњи на основу креираног дијаграма тока података. У трећем поглављу ће бити представљене митигације за претходно поменуте претње. Последње поглавље закључује рад и наводи предлоге за унапређење.

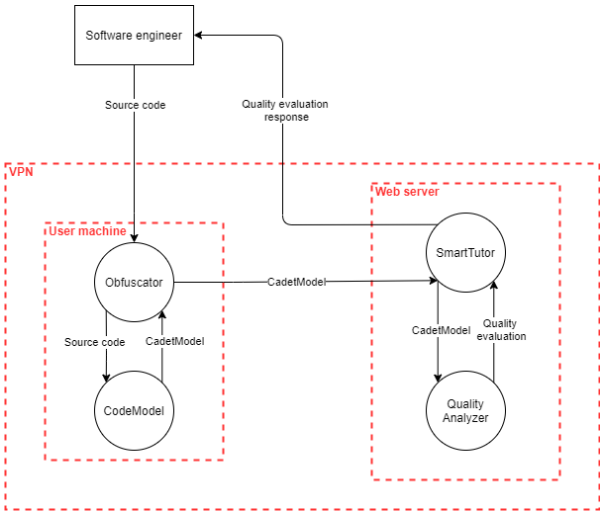
2. БЕЗБЕДНОСНА АНАЛИЗА ДИЗАЈНА

Примарна функционалност Clean CaDET платформе је детекција проблема унутар изворног кода (енгл. *code smells*) неког програма користећи моделе вештачке интелигенције. Затим, кориснику нуди персонализоване предлоге у облику образовног садржаја који ће му помоћи у решавању идентификованих проблема. Платформа има улогу дигиталног асистента софтверским инжењерима и интегрише се у њихова развојна окружења како би анализирала изворни код [1].

С обзиром да се у великом делу софтверских компанија изворни код сматра пословном тајном, а анализа се врши тако што се сам изворни код шаље са корисничких машина на Clean CaDET сервере, изворни код се може дефинисати кључним ресурсом у систему и потребно га је заштити од пада у погрешне руке.

Clean CaDET платформа ради са апстрактним моделом података који се креира парсирањем изворног кода – CaDETModel. У процесу парсирања игнорише се екстерни код (на пример, код из System простора имена) и креирају се објекти само од класа које су послате на Clean CaDET платформу [2]

Слика 1. приказује дијаграм тока података Clean CaDET платформе.



Слика 1 – *дијаграм тока података система у коме се CaDETModel парсира на корисничкој машини*

Парсирање изворног кода и креирање апстрактног модела се врши на корисничкој машини, затим се над SourceCode пољима елемената модела примењују трансформације обфускације. Обфускован модел се шаље на сервер и над њим се врши анализа квалитета.

2.1. Моделовање претњи

Shostack [3] моделовање претњи дефинише као употребу апстракција које помажу у размишљању о ризицима. Идеја која стоји иза моделовања претњи је

разумевање потенцијалних безбедносних претњи систему, утврђивање ризика и успостављање одговарајућих митигација. За идентификовање претњи користиће се метода под називом STRIDE.

У табели 1. налази се приказ идентификованих и класификованих претњи, као и делови система на које се поменуте претње односе. Уз сваку претњу је представљен и кратак опис претње.

Табела 1 - *Категоризација потенцијалних претњи*

ИД	Категорија	Мета	Опис
1.1	*Spoofing / Information disclosure*	Корисничка машина	Нападач може да се лажно представи као сервер, што омогућава да се изворни код са корисничке машине шаље директно нападачу
1.2	*Spoofing*	Сервер	Нападач може да се представи као обичнан корисник и оствари комуникацију са сервером.
1.3	*Information disclosure*	Комуникација са сервером	Нападач може да чита изворни код који се шаље на анализу ка серверу тако што прислушкује комуникацију између клијента и сервера преко мреже (*Man-in-the-middle* напад [4])
1.4	*Denial-of-service*	Сервер	Нападач може да оптерети сервере слањем великог броја захтева за анализу изворног кода
1.5	*Information disclosure*	Сервер	Инсајдер унутар Clean CaDET система може да дође у посед података користећи неки програм за читање радне меморије
1.6	*Repudiation*	Сервер	Инсајдер унутар Clean CaDET система може да порекне да је вршио малициозне радње над машинама

3.2 Решавање претњи

У табели 2. приказани су потенцијални начини на који ће се идентификоване претње решити.

Табела 2.- *Потенцијални начини за решавање претњи*

ИД	ИД Претње	Митигација
2.1	1.1	Аутентификација обе стране путем дигиталних сертификата приликом комуникације између клијента и сервера
	1.2	
2.2	1.3	Имплементација HTTPS (енгл. *Hypertext Transfer Protocol Secure*) протокола [5] у комуникацији између клијента и сервера
2.3	1.4	Остваривање везе између клијента и сервера путем виртуелне приватне мреже (енгл. *Virtual Private Network* – VPN [6])
2.4	1.5	Генерисање лог записа, анализа и управљање логовима
2.5	1.6	Имплементација механизама заштите лог датотека

3. ДИЗАЈН ЗА БЕЗБЕДНО УНАПРЕЂЕЊЕ ПЛАТФОРМЕ

У овом поглављу ће конкретније бити представљене митигације за претходно поменуте претње. Биће описан начин комуникације између клијентског и серверског дела платформе и кратак опис инфраструктуре јавних кључева. Затим ће бити приказане смернице за правилно конфигурисање виртуелне приватне мреже. На крају поглавља ће се анализирати појединачна поља апстрактног модела података са циљем да се за свако проблематично поље пронађу адекватне мере прикривања.

3.1. Безбедна комуникација између клијента и сервера

Сва комуникација између клијента и сервера је шифрована уз помоћ TLS (енгл. Transport Layer Security) протокола [7]. Ради повећања безбедности система, није довољно да клијентска апликација зна коме шаље изворни код, већ и да сервер зна од кога изворни код заправо пристиже. Зато се за комуникацију између клијената и сервера користи клијентска аутентификација [8].

3.2. Употреба инфраструктуре јавних кључева

Да би се осигурала безбедна комуникација између актера у Clean CaDET систему потребно је имплементирати инфраструктуру јавних кључева (у даљем тексту PKI).

3.2.1. Издавање сертификата

Свакој машини која користи услуге Clean CaDET система ће бити потребан приватни и јавни кључ. Уколико се приватни и јавни кључ генеришу на серверима и шаљу клијенту, увек ће постојати шанса да ће нападач на неки начин доћи у посед приватног кључа. Најбољи начин да се избегну проблеми са транспортом приватног кључа јесте да се он уопште

не транспортује преко мреже, већ да се генерише на корисничкој машини и никада не напушта систем.

За поступак пријаве и провере валидности података је задужено регистрационо тело (енгл. *registration authority* [9]) унутар PKI. Захтев за издавање сертификата се може послати путем форме на сајту осигураним TLS протоколом, а генерисани сертификат се кориснику може послати електронском поштом користећи S-MIME протокол [10]. Овај процес се такође може аутоматизовати тако што ће клијентска апликација генерисати парове кључева, послати захтев за издавање сертификата, а затим генерисани сертификат и приватни кључ сачувати у сигурном складишту на машини.

3.3. Употреба виртуелне приватне мреже

Део површине напада Clean CaDET система представљају IP адресе сервера. Како би се избегли *Denial-of-service* напади на Clean CaDET сервере, потребно је смањити површину напада скривањем IP адреса од нападача. Ово се може постићи употребом виртуелних приватних мрежа.

Да би виртуелна приватна мрежа донела предности у безбедности, потребно ју је правилно конфигурисати. Центар за националну сајбер безбедност Уједињеног Краљевства (енгл. *National Cyber Security Centre* – NCSC) [11] препоручује неколико смерница приликом конфигурације виртуелних приватних мрежа [12]:

- Користити IPSec (енгл. *Internet Protocol Security*) протокол [13] како би се добила флексибилност у избору између широког спектра интероперабилних производа;
- Користити аутентификацију базирану на сертификатима;
- Ако је могуће, користити изворни (енгл. *native*) клијент за своје платформе;
- Користити VPN који се аутоматски повезује, тако да корисници уређаја не морају ручно да га укључују;
- Избегавати *split tunneling* да би се смањио ризик од цурења података изван VPN-а [12].;
- Тестирати више VPN провајдера како би се открило који је најбољи за потребе организације и који је добољно робустан и отпоран на нападе.

3.4. Обфускација апстрактног модела података

Анализом апстрактног модела података се дошло до закључка да је могуће анонимизовати или уклонити нека поља у моделу, без утицаја на резултате анализе квалитета изворног кода. Такође, на основу комплексности и количине података које захтева, процес анализе је било могуће поделити на три засебна нивоа:

- Основна анализа,
- Анализа средње комплексности и
- Напредна анализа.

Могуће је извршити основну анализу квалитета изворног кода користећи само метрике. Ова анализа се састоји од примене простих правила на пристигле метрике. Количина информација која може процурити при анализи метрика је минимална, јер метрике не откривају ништа о семантици саме класе. Такође су игнорисане везе ка осталим класама, те је немогуће направити граф зависности класа. Анализа средње комплексности би захтевала поља на основу којих је могуће саставити граф зависности класа. Ова анализа би донела квалитетније резултате, међутим, и количина информација које би процуреле би била већа. Крајње, уколико се не поставе никаква ограничења на апстрактни модел података, над њим ће бити могуће извршити најнапредније анализе, али долази до потенцијалне ситуације где све информације падну у руке нападача, јер се шаље комплетан, непромењен изворни код класе или њених чланова. Најкорисније решење би било да корисник има опцију да изабере једну од три понуђене врсте анализе, у зависности од његових сигурносних потреба.

3.4.1. Анонимизација имена

Независно од нивоа анализе који је у питању, због повратних информација је на сервер потребно слати имена чланова апстрактног модела података. С обзиром да изворно име класе са собом носи потенцијал за цурење информација, потребно га је некако анонимизовати. Табела 3 наводи називе класа и поља која представљају имена чланова.

Табела 3 - *Поља која је потребно анонимизовати*

Класа	Поље
`CaDETClass`	`Name`
`CaDETClass`	`FullName`
`CaDETField`	`Name`
`CaDETMember`	`Name`

Над горе поменутим пољима ће бити извршене трансформације хеширања користећи алгоритам SHA256 (*Secure Hash Alghoritm 2*) [14]. Анонимизација се ослања на постојање посебне структуре података на клијенту – CaDETHashTable, која врши хеширање података и чува податке у облику **кључ: вредност**.

Процес анонимизације имена је следећи: приликом формирања апстрактног модела података, када се наиђе на поље поменуто у **Error! Reference source not found.**, израчунаће се SHA256 хеш вредност тог поља. Оригинална вредност поља се мења хеш вредношћу, а у CaDETHashTable се додаје нови елемент чији је кључ израчунати хеш, а вредност је оригинална вредност поља.

4. ЗАКЉУЧАК

У овом раду је извршена безбедносна анализа и креиран је модел претњи за софтверско решење за анализу квалитета изворног кода. Представљен је дизајн за безбедно унапређење Clean CaDET платформе, где је приказан процес издавања дигиталних сертификата у оквиру инфраструктуре

јавних кључева, представљен начин комуникације између клијентског и серверског дела система и наведене смернице за правилно конфигурисање виртуелне приватне мреже. На основу детаљне анализа апстрактног модела података су пронађена конкретна поља која је потребно анонимизовати.

Ради додатног повећања безбедности Clean CaDET платформе, даље истраживање би се могло усмерити у саму анализу безбедносних митигација, на пример, дубље истраживање проблема који настају приликом коришћења инфраструктуре јавних кључева у самом решењу.

5. ЛИТЕРАТУРА

[1] „Clean-CaDET: Who is it for?," [На мрежи]. Available: https://github.com/Clean-CaDET/platform#what-is-the-problem. [Последњи приступ 5 Октобар 2021].

[2] „Clean CaDET Code Model," [На мрежи]. Available: https://github.com/Clean-CaDET/platform/wiki/Module-Code-Model. [Последњи приступ 07 Август 2021].

[3] A. Shostack, Threat Modeling: Designing for Security, John Wiley & Sons, 2014.

[4] „What is MITM (Man in the Middle) Attack," [На мрежи]. Available: https://www.imperva.com/learn/application-security/man-in-the-middle-attack-mitm/. [Последњи приступ 29 Септембар 2021].

[5] „RFC 2818 – HTTP Over TLS," [На мрежи]. Available: https://datatracker.ietf.org/doc/html/rfc2818. [Последњи приступ 29 Септембар 2021].

[6] „What is a VPN," [На мрежи]. Available: https://www.kaspersky.com/resource-center/definitions/what-is-a-vpn. [Последњи приступ 1 Октобар 2021].

[7] „RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2," [На мрежи]. Available: https://tools.ietf.org/html/rfc5246. [Последњи приступ 29 Септембар 2021].

[8] Cloudfare, „Introducing TLS with Client Authentication," [На мрежи]. Available: https://blog.cloudflare.com/introducing-tls-client-auth/. [Последњи приступ 29 Септембар 2021].

[9] „What is a Registration Authority," [На мрежи]. Available: https://www.primekey.com/wiki/what-is-a-registration-authority/. [Последњи приступ 30 Септембар 2020].

[10] „Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0," [На мрежи]. Available: https://datatracker.ietf.org/doc/html/rfc8551. [Последњи приступ 30 Септембар 2021].

[11] „National Cyber Security Centre," [На мрежи]. Available: https://www.ncsc.gov.uk/. [Последњи приступ 1 Октобар 2021].

[12] „Virtual Private Networks (VPNs)," [На мрежи]. Available: https://www.ncsc.gov.uk/collection/device-security-guidance/infrastructure/virtual-private-networks. [Последњи приступ 1 Oktobar 2021].

[13] „RFC 2406 - IP Encapsulating Security Payload," [На мрежи]. Available: https://datatracker.ietf.org/doc/html/rfc2406. [Последњи приступ 1 Октобар 2021].

[14] W. Penard и T. v. Werkhoven, „On the Secure Hash Algorithm," [На мрежи]. Available: https://web.archive.org/web/20160330153520/http://www.staff.science.uu.nl/~werkh108/docs/study/Y5_07_08/infocry/project/Cryp08.pdf. [Последњи приступ 2 Октобар 2021].

[15] A. Chuvakin, K. Schmidt и C. Phillips, Logging and Log Management: The Authoritative Guide to Understanding the Concepts Surrounding Logging and Log Management, Syngress, 2013.

Кратка биографија:

Милан Миловановић рођен је 1. априла 1997. године у Смедереву. 2016 године постаје студент на Факултету техничких наука у оквиру Универзитета у Новом Саду, смер Софтверско инжењерство и информационе технологије. 2020. године је стекао диплому основних студија и уписао мастер студије на смеру Софтверско инжењерство и информационе технологије.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.42:004.738.12
DOI: https://doi.org/10.24867/16BE36Ljubic

ФОРЕНЗИКА ЕЛЕКТРОНСКЕ ПОШТЕ

E-MAIL FORENSICS

Јанко Љубић, *Факултет техничких наука, Нови Сад*

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – *Рад представља истраживање у области дигиталне форензике електронске поште. Чињеница је да је значајан број електронских порука данас малициозног карактера и да често постоји потреба за утврђивањем идентитета и аутентичности учесника у конверзацији. Циљ рада био је преглед и поређење постојећих форензичких техника и алата. Анализирано је више техника и алата. Закључене су предности и недостаци форензике мејлова као и чињеница да постоји потреба за њеним унапређивањем.*

Кључне речи: *Форензика, дигитални докази, електронска пошта*

Abstract – *This work presents research in a field of the Digital Email Forensics. Today, it is a well-known fact that the significant number of emails are malicious in nature. There is often a need to identify and authenticate the conversation participants, which is being done by use of email forensics. The aim of this work is a presentation and comparison of existent forensics methods and tools. Multiple forensics techniques and tools were analyzed. In the end, the current advantages and disadvantages of email forensics were concluded, alongside with a fact of existing need for its improvement and further development.*

Keywords: *Forensics, digital evidence, email*

1. УВОД

Приличан број електронских порука данас јесте малициозног карактера. Пошиљаоци имају за циљ да се лажно представе или путем мејла инфилтрирају у систем примаоца и тиме нанесу штету.

Такође, често постоји потреба за утврђивањем аутентичности пошиљаоца. Одговоре на ова питања даје форензика електронске поште. У овом раду су упоређене релевантне технике и алати, приказане су њихове предности и мане као и услови у којима их ваља или не ваља користити.

2. ЕЛЕКТРОНСКА ПОШТА

Једна електронска порука (у даљем тексту мејл) се шаље од пошиљаоца примаоцу, или примаоцима и састоји се од више саставних делова од којих сваки носи информације корисне за одређивање садржаја поруке као и идентификације учесника у комуникацији или информације о самој поруци – мејлу. Главне делове мејла представљају заглавље и тело. Заглавље (табела 1) садржи информације у форми парова, кључ вредност, које указују на адресу пошиљаоца, адресу примаоца, информације о времену слања и слично. Тело садржи суштинску поруку мејла коју је креирао пошиљалац.

Табела 1. *Важнија поља у заглављу мејла*

Назив поља	Опис
From	Мејл адреса и опционо име аутора мејла.
To	Мејл адреса/адресе, и опционо име/имена прималаца мејла.
Cc	Carbon Copy – мејл адреса/адресе корисника који ће добити копију мејла.
Bcc	Blind Carbon Copy – мејл адреса/адресе корисника који су осталима невидљиви, а добиће своју копију мејла.
Subject	Кратки наслов мејла.
Date	Временски тренутак када је мејл креиран.
Reply-To	Мејл адреса на коју би требало послати одговор.
Message-ID	Аутоматски генерисано поље које представља јединствени идентификатор поруке.
Recieved	Информација генерисана од стране мејл сервера који су руковали поруком, у обрнутом редоследу, а која служи праћењу кретања поруке кроз мрежу.

Пут мејла од пошиљаоца до примаоца се може разликовати у зависности од архитектуре система електронске поште.

Тип 1 – архитектура у којој нема интернета већ су и пошиљалац и прималац повезани на исти, дељени систем који прослеђује поруке. Мејлови се креирају и читају уз помоћ агентских апликација.

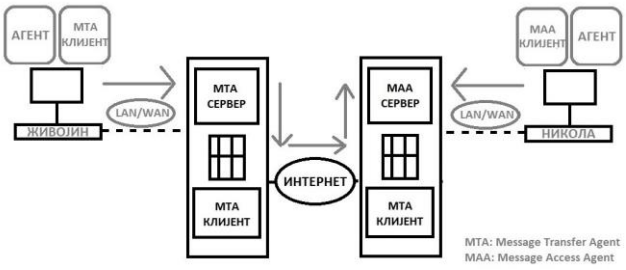
Тип 2 – пошиљалац и прималац више не деле исти систем већ се мејл шаље путем интернета од једног сервера до другог. Пошиљалац користи агентски програм да креира и пошаље мејл а прималац користи агентски програм да приступи мејловима који се

НАПОМЕНА:
Овај рад проистекао је из мастер рада чији ментор је био др Стеван Гостојић, ванр. проф.

налазе у поштанском сандучету његовог сервера електронске поште.

Тип 3 – прималац је и даље директно повезан на свој сервер електронске поште али је пошиљалац сада повезан преко LAN/WAN (LAN – Local Area Network, WAN – Wide Area Network) мреже.

Тип 4 – данас најчешће сретани тип архитектуре. И пошиљалац и прималац су на сервере електронске поште повезани путем LAN/WAN конекције [1] (слика 1).



Слика 1: Најчешћа архитектура мејл система

Протоколи размене мејлова представљају стандарде размене информација између корисничких агентских апликација и сервера електронске поште. Разликујемо више протокола по функционалности и месту примене, а као најзаступљенији истичу се SMTP, MIME, POP3, IMAP4, HTTP, Microsoft Exchange.

3. ТЕХНИКЕ ФОРЕНЗИКЕ ЕЛЕКТРОНСКЕ ПОШТЕ

Дигитална форензика представља процес коришћења ваљаних научних метода за идентификовање, прикупљање, чување, анализу и презентовање дигиталних доказа добијених из дигиталних извора а за потребе лакшег и бољег разумевања и реконструкције догађаја који могу представљати кривично дело [2]. Дигитална форензика се грана у различитим правцима па тако разликујемо форензику масовне и радне меморије, форензику оперативних система, форензику мобилних уређаја и тако даље. Једна од грана дигиталне форензике јесте и форензика рачунарских мрежа. У овом случају, дигитални докази преносе се путем рачунарских мрежа. Електронска порука, односно мејл, представља један од таквих дигиталних доказа.

3.1. Идентификовање доказа

Мејл се идентификује тако што се тражи свуда по мрежи, јер је могуће да је у локалу обрисан али да се његова копија и даље чува у мејл серверу, на пример. Поред мејла, дигитални доказ представљају и логови са уређаја мејл система.

3.2. Прикупљање доказа

Да би се докази обрадили морају се прикупити на ваљан начин који подразумева да ће стање на уређајима са којих се докази прикупљају бити неизмењено, као и да ће докази бити очувани и копирани у истом стању у ком су и пронађени. Такође је важно забележити сваку од радњи и идентификовати особе које су радњу преузеле како би се обезбедио ланац доказа и знало ко је, у ком тренутку и на који начин утицао на и прикупљао дигиталне доказе. Медиј на ком ће се чувати прикупљени подаци мора пре почетка фазе прикупљања и чувања бити форензички чист.

Када се посматра електронска порука и прикупља као доказ, мора се водити рачуна на то у ком се формату порука преузима. Порука се треба преузети у свом изворном формату, а не претварати у формат који одговара истражитељима. Често се прави копија целокупног сандучета за шта постоји више техника.

- Истраживање сервера - Мејлови који су обрисани са клијента (било пошиљаоца или примаоца) и чији опоравак није могућ, могу се затражити од мејл сервера пошто већина њих чува копије након што мејл доставе. Даље, логови сервера могу помоћи да се пронађе адреса рачунара одговорног за покретање мејл трансакције. Неки од сервера могу чувати информације о власнику поштанског сандучета као што је на пример број кредитне картице власника, што може много помоћи у проналажењу идентитета особе [3].
- Истраживање уређаја на мрежи - Рутери, свичеви или неки заштитни зидови и њихови логови.
- Тактика мамца - Уколико је мејл адреса са којег је примљена порука која се испитује права, истражитељи могу послати мамац мејл пошиљаоцу. Такав мејл садржи „img“ елемент чији се извор слике (енгл. source - src) налази на неком HTTP серверу.

3.3. Чување доказа

Да би се у било ком тренутку могло тврдити да су изнети докази веродостојни, мора се осигурати то да се у току процеса дигиталне форензике они ни на који начин не мењају, као и да се зна у ком тренутку су докази били предмет рада које особе (ланац доказа).

Неки од принципа којих се треба држати приликом обављања процеса дигиталне форензике су чување оригиналних доказа, фотографисање физичких доказа, фотографисање екрана приликом приступа дигиталним доказима, чување дупликата комплетних доказа, вршење хеш тест анализе са циљем утврђивања аутентичности претходно копираних доказа, чување ланца доказа (документовање датума, времена и свих осталих информација приликом пријема доказа на обраду).

3.4. Прегледање доказа

Докази прикупљени у претходној фази представљају сировину из које треба издвојити суштину. Суштину представљају докази који се из неког разлога сматрају релевантнијим извором информација за оповргавање или потврђивање кривичног тела, па се тако из скупине свих електронских порука из пријемног сандучета може издвојити секвенца око критичног мејла како би се извршила анализа њихових заглавља, серијских бројева и садржаја.

- Иницијално прегледање и предпроцесирање - Издвајање одређених мејлова из скупа

форматираног као ost или pst фајл може се обавити коришћењем различитих алата.

- Опоравак обрисаних података - Фаза прегледања доказа обухвата евентуалну реконструкцију обрисаних доказа, на пример мејлова који су обрисани у локалу али се и даље налазе на серверу. Обрисани мејлови су углавном доступни за опоравак у неком року. Након тог рока, опоравак је могућ уз помоћ неког од доступних алата који ће прегледати сачувани .ost, .pst или mbox фајл [4].
- Откључавање и дешифровање доказа

3.5. Анализа доказа

Фаза анализе подразумева процесуирање информација које се односе на објекат истраге са циљем одређивања чињеница о неком догађају, важност доказа и одговорност особе или особа.

- Претрага по тексту и шаблонима
- Анализа заглавља мејла - Срж форензичког истраживања електронске поште. Сваки мејл има тачно једно заглавље које је структурално подељено на поља у форми кључ-вредност. Заглавље мејла садржи главне информације које одређују ко је, коме, када, како (којим протоколом), са које адресе и шта послао. Нека од важнијих поља заглавља која се анализирају су *Recieved, X-Received, Received-SPF, Delivered-To, Return-Path, Authentication-Results, DKIM-Signature, Message-ID.*
- Утврђивање временског тока
- Визуализација

4. АЛАТИ ЗА ФОРЕНЗИКУ ЕЛЕКТРОНСКЕ ПОШТЕ

Од велике скупине алата издвојени су они за чије коришћење није било неопходно плаћање већ постоје верзије које је бесплатно могуће користити у ограниченом или неограниченом временском периоду. Такође, за алате за које је било неопходно издвојити новчана средства коришћене су информације из других истраживања, документација и слично.

4.1. Autopsy 4.19.1

Бесплатни алат отвореног кода који подржава рад и у интернет и у режиму без интернета. Врши аквизицију података са локалних дискова, фајлова и фолдера као и слика дискова. Подржава више мејл формата а функционише на више оперативних система. Може да анализира екстерне уређаје а подржава и напредне опције претраге и визуелизације, опоравка података и извоза у више различитих формата.

4.2. Aid4Mail Professional Trial 4.7

Бесплатна пробна верзија алата чији код није доступан јавности способан је да врши аквизицију података са локалног система, с тим што може да анализира искључиво *Windows* оперативни систем.

Подржава велики број формата мејлова које може проналазити и на екстерним уређајима и извозити у више различитих формата. Постоји опција опоравка података и претраге али не и нарочите визуелизације.

4.3. MailXaminer 4.9.3

Бесплатна пробна верзија чији је код такође задржан од стране произвођача нуди аквизицију података и са локалне машине и са интернета уз подршку мноштва мејл формата. Функционише искључиво на *Windows* оперативном систему с тим што може анализирати и екстерне уређаје. Подржава претрагу, визуализацију и опоравак података. Функционалност извоза података одлично је покривена.

4.4. AccessData FTK Imager 4.5.0.3

Алат који функционише само у режиму без интернета и може да врши аквизицију података са локалне машине али и са екстерних уређаја. Подржава доста мејл формата и покрива остале важније функционалности укључујући и претрагу, опоравак, визуализацију и извоз.

4.5. BitCurator Environment

Линукс дистрибуција отвореног кода који врши аквизицију локалне машине и свих типова мејл архива. Ради са екстерним уређајима и омогућава претрагу, визуализацију, опоравак и извоз података.

4.6. Paraben E-Mail Examiner

Затворени алат који се наплаћује и пружа офлајн услуге обраде преко 750 мејл формата на *Windows* оперативном систему. Не подржава рад са екстерним уређајима али омогућава визуализацију, опоравак и извоз података.

4.7. EnCase

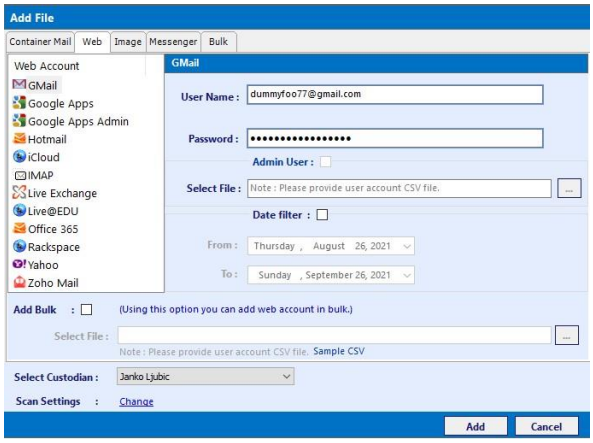
Затворени алат који се наплаћује и омогућава рад у режиму са и без интернета. Аквизиција података се обавља са интернета или локалне машине а подржано је преко 600 различитих формата мејлова које је могуће претраживати, опорављати, визуализовати и извозити у склопу више различитих оперативних система.

5. СТУДИЈА СЛУЧАЈА

Неке од ових техника и алата демонстрирани су на хипотетичком случају. Замислимо да постоји сумња да је са налога „dummyfoo77@gmail.com“ послата електронска порука чија садржина представља средство преваре и покушај лажног представљања. Задатак је извршити вештачење приложеног рачунара тако што ће се утврдити: да ли је преко уређаја остварен приступ налогу са адресом „dummyfoo77@gmail.com“, да ли се на том налогу налазе електронске поруке, садржај тих порука, да ли су поруке аутентичне, адресе на које су поруке послате и друге околности у вези настанка тих порука.

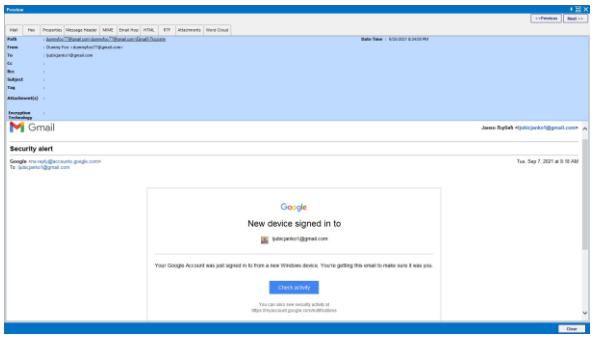
Идентификован је лаптоп уређај и забележен његов модел, изглед као и стање. Прикупљени су докази тако што се уз помоћ алата *SysTools MailXaminer v4.9.3* приступило мејл налогу осумњиченог (слика 2). Прикупљени докази сачувани су у непромењеном

облику на екстерној меморији која је претходно форензички очишћена.



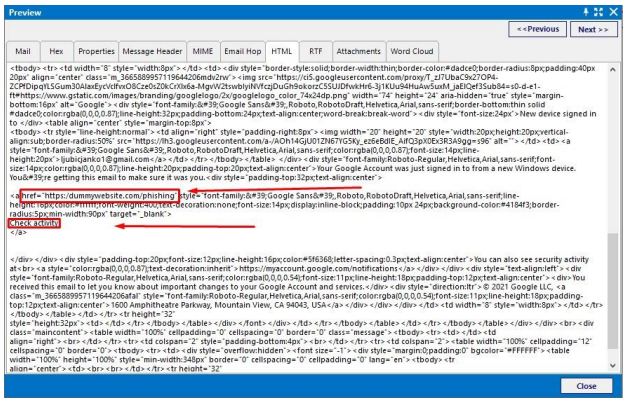
Слика 2: Приступање налогу из алата

Приликом прегледања доказа издвојен је мејл са карактеристикама које одговарају опису датом у хипотетичком случају (слика 3).



Слика 3: Пронађени мејл

Порука је анализирана и примећено је да порука садржи дугме које шаље корисника на нежељену и лажну локацију са циљем крађе података (слика 4). Такође је утврђен идентитет пошиљаоца.



Слика 4: Анализирани садржај мејла

Докази су презентовани у виду налаза који се прилаже суду.

6. ЗАКЉУЧАК

Са растом броја корисника електронске поште, расте и број потенцијалних жртава, али и број потенцијалних починилаца кривичних дела која се извршавају уз помоћ или преко електронских порука. Постоји потреба да се починиоци на научно важан начин пронађу и повежу са кривичним делом да би се могли санкционисати.

Коришћење проверених и устаљених техника смањује могућност прављења грешака које у правном или неком другом смислу могу да омету истрагу и долажење до истине.

Проблем већине постојећих техника и алата је тај што су артефакти које они анализирају опште познати и због тога погодни као тачке којима би у будућности могло бити манипулисано, поготово уколико се узме у обзир брзина развоја антифорензичких и криминалних алата и техника. Тиме би се могла довести у питање веродостојност тврдњи које су у претходном периоду важиле за истините, што у правној пракси није прихватљиво.

Како би се умањио утицај растуће гране антифорензике, неопходно је константно унапређивати и осавремењавати постојеће форензичке технике и алате, као и системе електронске поште који у многим случајевима допуштају манипулацију која може бити малициозног карактера.

7. ЛИТЕРАТУРА

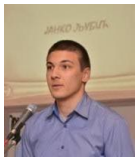
[1] G Chhabra, D Bajwa, Review of E-mail System, Security Protocols and Email Forensics. International Journal of Computer Science & Communication Networks (IJCSCN), Vol.5, Issue 3, pp.201-211, 2015.

[2] Andre Arnes, Digital Forensics, Norweigan University of Technology and Science, Norway, 2018.

[3] M. Tariq Banday, Techniques and Tools for Forensic Investigation of E-mail, University of Kashmir India, 2011.

[4] Tracy King, "How to Retrieve Emails from Gmail, Outlook, Hotmail, and Yahoo", 2021. Доступно на https://www.easeus.com/file-recovery/recover-deleted-email-files.html [посећено 05. септембра 2021.]

Кратка биографија:



Јанко Љубић рођен је 02.02.1998. године у Врању. Мастер рад на Факултету техничких наука из области Дигиталне форензике одбранио је 2021. године. контакт: janko.ljubic@uns.ac.com

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.31
DOI: https://doi.org/10.24867/16BE37Kozomora

KONCEPT HIBRIDNOG ELEKTRIČNOG SKLADIŠTENJA ENERGIJE SA SUPERKONDENZATORIMA U ELEKTRIČNIM VOZILIMA

CONCEPT OF HYBRID ELECTRIC STORAGE OF ENERGY WITH SUPERCAPACITORS IN ELECTRIC VEHICLES

Marko Kozomora, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Savremeni hibridni sistem za skladištenje električne energije u električnim vozilima koji se sastoji od litijum-jonskih baterija i superkondenzatora je opisan u ovom radu. Predstavljena je paralelna topologija potpuno aktivnog hibridnog sistema, kao i menadžment energije za datu topologiju i simulicija.*

Ključne reči: *Električna vozila, Hibridni sistem za skladištenje električne energije, Baterija, Superkondenzator*

Abstract – *A modern hybrid electric storage system in electric vehicles with lithium-ion batteries and supercapacitors is described in this work. The parallel topology of a fully active hybrid system is presented, as well as energy management for a given topology and simulation.*

Keywords: *Electric vehicles, Hybrid electric storage system , Battery, Supercapacitor*

1. UVOD

Vozila koja su zasnovana na nafti, zbog svojih mana i negativnog uticaja na životnu sredinu, imaju tedenciju da u bliskoj budućnosti budu zamenjena električnim vozilima (EV).
Glavni nedostatak električnog vozila jeste njegova jedinica za skladištenje zasnovana na baterijama, koja zbog trenutnog stanja razvoja električno vozilo čini nekompetetnim na tržištu ili nudi električno vozilo sa smanjenim performansama što ga čini manje prihvatljivim za potrošače.
Relativno kratak vek trajanja, velika osetljivost na ambijentalne uslove, opasnosti po životnu sredinu i relativno ograničena izlazna snaga samo su neki od nedostataka trenutne tehnologije baterija. Međutim, nedavno se pojavila nova vrsta tehnologije za skladištenje električne energije koja se pokazala kao obećavajući dodatak tehnologiji baterija.
Dug vek trajanja i velika gustina snage čine tehnologiju superkondenzatora održivim i sigurnim dodatkom baterijama, nudeći mogućnost da električna vozila učini konkurentnijim na tržištu [1].

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Bane Popadić, docent.

2. HIBRIDNI SISTEM ZA SKLADIŠTENJE ELEKTRIČNE ENERGIJE

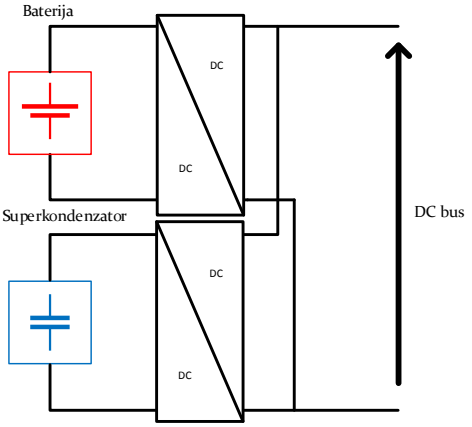
Iako su prošlih decenija postignuti značajni pomaci radi poboljšanja performansi baterije, glavni problem dolazi od vršne upotrebe. Čak i u malim elektronskim uređajima oštećenje baterije nastaje iznenadnom upotrebom energije iz baterija. Ovakva situacija je stalna u EV, pogotovo što razni faktori, kao što su stil vožnje i put, uzrokuju brze promene potrebne snage koja se uzima/predaje sistemu za skladištenje električne energije.
Superkondenzatori su slični elektrohemijski sistemi za skladištenje električne energije, ali glavna razlika je u tome što imaju visoku sposobnost brzog punjenja tj. pražnjenja. Superkondenzatori se ne mogu samostalno koristiti kao izvor električne energije u EV iz razloga što imaju malu gustinu energije. Ipak superkondenzatori su dobre opcije za isporuke iznenadne snage koje zahteva EV.
Kombinacija baterija i superkondenzatora poznatija je kao hibridni sistem za skladištenje električne energije. Kada se koristi hibridni sistem za skladištenje električne energije štetni efekat struje u baterijama se smanjuje i produžava se životni vek baterije. Ovo je suštinski aspekt za upotrebu hibridnog sistema za skladištenje električne energije. Takođe upotreba hibridnog sistema povećava efikasnost EV skladištenjem rekuperativne energije iz kočnica tokom usporavanja EV [2].
Adekvatno realizovan hibridni sistem za skladištenje električne energije može imati veoma veliki uticaj na ukupne performanse EV i na taj način pomaže njihovom prodiranju na tržište. Prednosti hibridnih sistema su: velika gustina energije, veliki broj životnih ciklusa, znatno manja težina od samih baterija i velika gustina snage.

2.1. Paralelna topologija potpuno aktivnog hibridnog sistema

Potpuno aktivna topologija ima najbolji kotrolni algoritam među topologijama hibridnog sistema za skladištenje električne energije, zahvaljujući DC/DC pretvaračima koji su povezani na jednosmerno međukolo radi razdvajanja baterije i superkondenzatora [3-4].
Paralelna topologija potpuno aktivnog hibridnog sistema je sistem koji se sastoji od baterije i superkondenzatora povezanih preko bidirekcionih DC/DC pretvarača na jednosmerno međukolo [5].
Razlog za primenu ovakve topologije je ta da je punjenje tj. pražnjenje baterije u potpunosti kontrolisano. DC/DC pretvarači služe da pretvore radni napon baterije i superkondenzatora u veći napon za potrebe jednosmernog

međukola sa kojeg se napaja trofazni invertor (~560 V, ako je linijski napon pogona 400 V). Bidirekcioni DC/DC pretvarači se koriste jer se pretpostavlja da su oba smera protoka energije dozvoljena, tako da i baterija i superkondenzator mogu da apsorbuju višak energije iz DC međukola ili da služe kao izvor energije u slučaju potrebe.

Cilj potpuno aktivne paralelne topologije hibridnog sistema za skladištenje električne energije je da udovolji različitim zahtevima opterećenja regulacijom snage deljene između baterije i superkondenzatora. Kontrolom bidirekcionih DC/DC pretvarača postiže se raspodela snage između njih.



Slika 1. *Paralelna topologija potpuno aktivnog hibridnog sistema*

3. MENADŽMENT ENERGIJE

Na osnovu zahteva vozača i stila vožnje, potreba za električnom energijom iz hibridnog sistema za skladištenje električne energije je izuzetno promenljiva i nepredvidljiva.

Zbog ovakvih faktora je potreban što bolji i efikasniji menadžment energije. Zahvaljujući bidirekcionim DC/DC pretvaračima ima se potpuno aktivna topologija, sa njihovom kontrolom baterijski deo i deo sa superkondenzatorom mogu da imaju ulogu izvora energije i mogu apsorbuju energiju na kontrolisan i veoma efikasan način.

Algoritam je prikazan na slici 2. iz više segmenata. Pre objašnjenja kako ovaj algoritam funkcioniše, potrebno je definisati stanja napunjenosti (SOC) baterije i superkondenzatora po nivoima:

Tabela 1. *Stanja napunjenosti baterije i superkondenzatora po nivoima*

SOC[%]	<30%	>30% i <90%	>90%
Baterija	0	1	2
Superkondenzator	0	1	2

Zbog istih oznaka za novoe stanja napunjenosti, prva cifra se odnosi na stanje napunjenosti baterije, dok se druga cifra odnosi na stanje napunjenosti superkondenzatora.

Na slici 2.a) se nalazi upravljački algoritam ukoliko je stanje napunjenosti sistema za skladištenje električne energije: **1 1** .

Prvo se proverava da li je promena brzine vozila u toku vremena veća od 0.5 m/s^2, ako je manja sva energije koja je potrebna električnom motoru se uzima iz baterije ili se sva energija iz električnog motora isporučuje bateriji. Ukoliko je promena brzine vozila u toku vremena veća od 0.5 m/s^2, energija koju baterija uzima ili isporučuje električnom motoru ostaje ista kao što je bila neposredno pre povećanja promene brzine iznad 0.5 m/s^2. Razlika između tražene energije i energije koja se uzima ili daje bateriji je energija koja se uzima ili isporučuje superkondenzatoru.

Na slici 2.b) se nalazi upravljački algoritam ukoliko je stanje napunjenosti sistema za skladištenje električne energije: **0 0** . Zbog nedovoljne napunjenosti baterije i superkondenzatora ovakav sistem za skladištenje električne energije nije u mogućnosti da daje energiju električnom motoru.

Opet se prvo proverava da li je promena brzine vozila u toku vremena veća od 0.5 m/s^2, ako je manja sva energija koja se može dati sistemu za skladištenje električne energije, isporučuje se bateriji. Ukoliko je promena brzine vozila u toku vremena veća od 0.5 m/s^2, energija koja se isporučuje bateriji ostaje ista kao što je bila neposredno pre povećanja promene brzine iznad 0.5 m/s^2. Razlika između energije koja se vraća u sistem za skladištenje i energije koja se isporučuje bateriji je energija koja se isporučuje superkondenzatoru.

Na slici 2.c) se nalazi upravljački algoritam ukoliko je stanje napunjenosti sistema za skladištenje električne energije: **1 0** ili **2 0** ili **2 1**

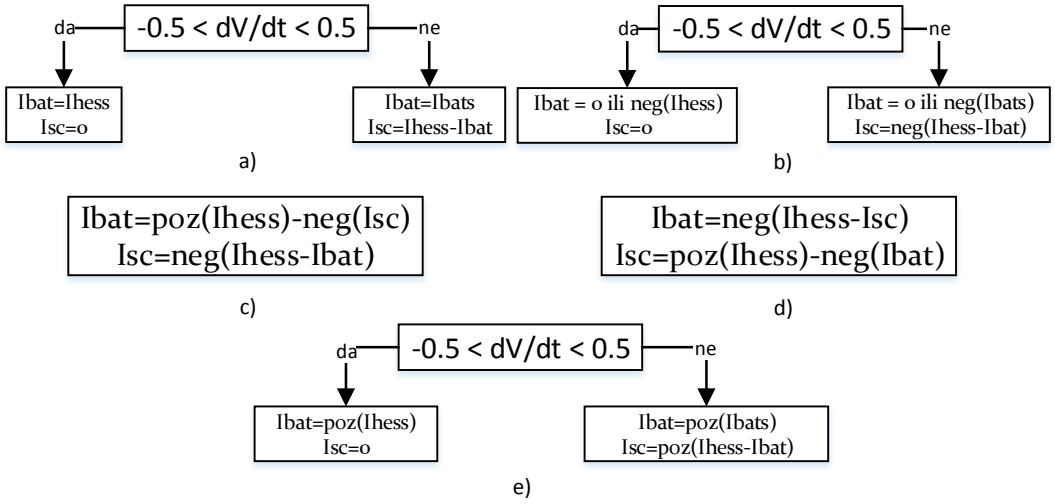
Kao što je navedeno u tabeli 1, jasno je da je u ovom slučaju stanje napunjenosti baterije za minimum jedan nivo veće od stanja napunjenosti superkondenzatora.

1 0 - nivo napunjenosti baterije je u stanju normalnog rada, dok se superkondenzator u ovom slučaju može smatrati praznim. Zbog nemogućnosti superkondenzatora da isporučuje energiju, sva zahtevana energije za električni motor mora biti isporučena od strane baterije. Ukoliko ima viška energije u bateriji onda se ona koristi i za upotrebu punjenja superkondenzatora. Energija kojom se puni superkondenzator se može dobiti iz baterije, iz električnog motora ili iz baterije i iz električnog motora zajedno.

2 0 - nivo napunjenosti baterije se nalazi u prepunjenom stanju, dok se superkondenzator kao i u prethodnom primeru može smatrati praznim. U ovakvom slučaju se sva potrebna energija za rad električnog motora kao i za punjenje superkondenzatora mora u svakom momentu uzimati iz baterije. Ne sme se doći u stanje u kojem će baterija dostići maksimalno dozvoljenu vrednost napunjenosti.

2 1 - nivo napunjenosti baterije se nalazi u prepunjenom stanju, dok se superkondenzator nalazi u stanju normalne napunjenosti.

Slično kao što je bilo u prethodnom slučaju, baterija se mora što pre isprazniti do normalnog stanja napunjenosti i mora da svu potrebnu energiju isporučuje električnom motoru i superkondenzatoru (moguće punjenje superkondenzatora dokle god mu je stanje napunjenosti ispod 90 %). Međutim, ukoliko je potrebna energija u ovom slučaju veća od energije koju može da isporuči baterija, ostatak energije se može uzeti iz superkondenzatora.



Slika 2. *Algoritam upravljanja sistema za skladištenje električne energije*

Na slici 2.d) se nalazi upravljački algoritam ukoliko je stanje napunjenosti sistema za skladištenje električne energije: **0 1** ili **0 2** ili **1 2**. U ovom slučaju stanje napunjenosti superkondenzatora je za minimum jedan nivo veće od stanja napunjenosti baterije. Algoritam upravljanja je srazmeran prethodnom slučaju, samo što su baterija i superkondenzator zamenili uloge.

Na slici 2.e) se nalazi upravljački algoritam ukoliko je stanje napunjenosti sistema za skladištenje električne energije: **2 2** .

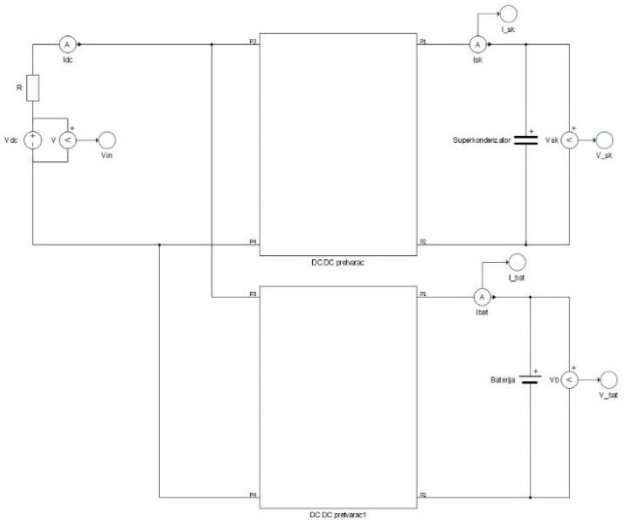
U ovom slučaju stanja napunjenosti baterije i superkondenzatora se nalaze u najvećem nivou, prepunjeni su. Zbog prepunjenosti, baterija i superkondenzator imaju samo mogućnost isporučivanja potrebne energije električnom motoru, bez opcije apsorbovanja energije.

Prvo se proverava da li je promena brzine vozila u toku vremena veća od 0.5 m/s^2, ako je manja sva energije koja je potrebna električnom motoru se uzima iz baterije. Ukoliko je promena brzine vozila u toku vremena veća od 0.5 m/s^2, energija koju baterija isporučuje električnom motoru ostaje ista kao što je bila neposredno pre povećanja promene brzine iznad 0.5 m/s^2. Razlika između tražene energije i energije koja se uzima iz baterije je energija koja se uzima iz superkondenzatora.

4. SIMULACIJA HIBRIDNOG SISTEMA ZA SKLADIŠTENJE ELEKTRIČNE ENERGIJE

U ovom poglavlju je opisan simulacioni deo rada koji je podeljen na softverski i hardverski deo. Softverski deo koji se odnosi upravljanje datim pretvaračima je realizovan u MATLAB/SIMULINK radnom okruženju, u ovom radu korišćena je MATLAB R2021a verzija. Hardverski deo koji se odnosi na hardver datog hibridnog sistema za skladištenje električne energije je realizon u Typhoon HIL radnom okruženju, u ovom radu korišćena je Typhoon HIL V2021.2 verzija kao i Typhoon HIL602+ uređaj.

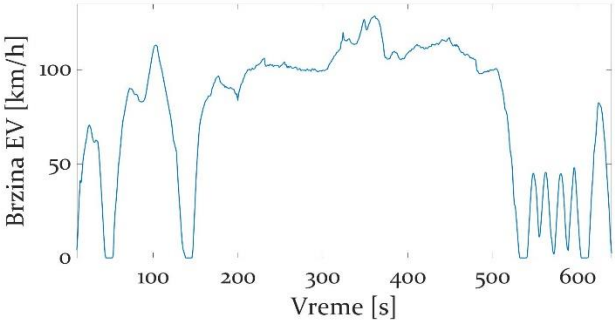
Hardver koji je realizovan u Typhoon HIL radnom okruženju (slika 3.) sastoji se od: Li-Ion baterije, superkondenzatora, dva trofazna bidirekciona DC/DC pretvarača i naponskog izvora jednosmernom međukola na koje je sistem povezan.



Slika 3. *Hardverski deo simulacije*

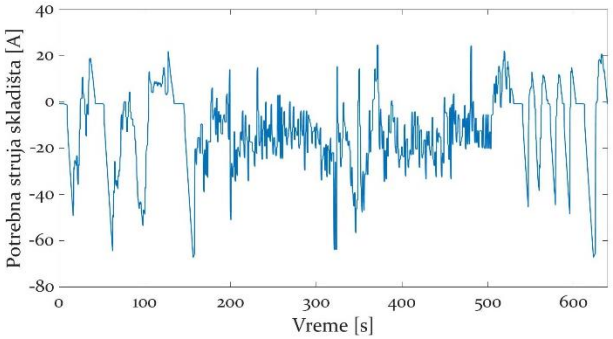
Softverski deo simulacije obuhvata menadžment energije i strujnu regulaciju trofaznih bidirekcionih DC/DC pretvarača.

U simulaciji je prikazano ponašanje hibridnog sistema za skladištenje električne energije za određeni profil vožnje (slika 4.) električnog vozila.



Slika 4. *Profil vožnje električnog vozila*

Za ovakav profil vožnje iz hibridnog sistema za skladištenje električne energije mora da se obezbedi struja prikazana na slici 5. koja se predaje/uzima jednosmernom međukolu.

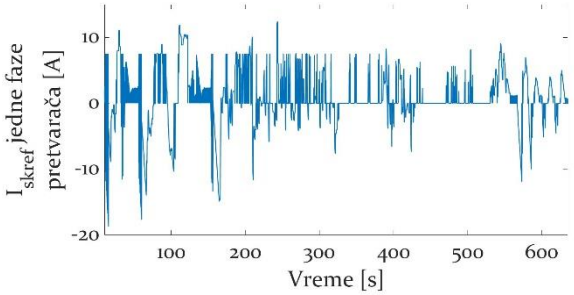


Slika 5. *Potrebna struja skladišta*

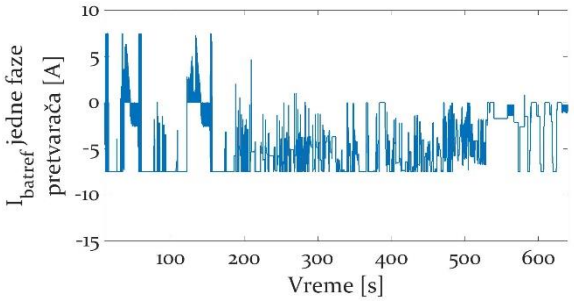
Ukoliko je potrebna struja skladišta negativne vrednosti to znači da se struja uzima iz skladišta, ukoliko je pozivitne vrednosti znači da se predaje skladištu (tj. u pitanju je regenerativno kočenje).

Na slici 5. mogu se primetiti tačni periodi povećane oscilacije potrebne struja skladišta, jer tada vozilo naglo ubrzava/usporava, što se može videti na slici 4.

Zahvaljujući već opisanom menadžmentu energije, na slikama 6. i 7. prikazane su podeljene reference struja bidirekcionog DC/DC pretvarača baterija i superkondenzatora. Svaka faza pretvarača prima trećinu potrebne struje koja je potrebna iz baterije ili superkondenzatora.



Slika 6. *Referenca struje pretvarača superkondenzatora*

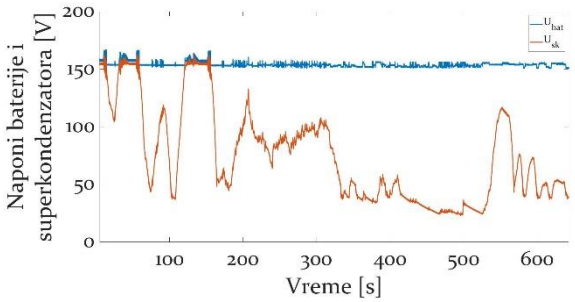


Slika 7. *Referenca struje pretvarača baterije*

Na svaku veliku promenu brzine, struja se veoma brzo menja i osciluje. Da se bateriji ne bi smanjio životni vek i broj ciklusa, menadžment energije određuje bateriji da ima što ustaljeniju referencu struje. Iz tog razloga, superkondenzator je zadužen za velike promene potrebne struje skladišta.

Zbog brzih usporavanja, jako malo regenerativne energije se vraća bateriji tokom simuliranog profila vožnje, što se može primeti na slikama 4. i 7.

Na slici 8. prikazani su naponi baterije i superkondenzatora. Vozilo je krenulo sa napunjenim hibridnim sistemom za skladištenje električne energije. Iako se baterija skoro ceo put prazni, sa slike 8. se može zaključiti da se baterija sporo prazni. Sa druge strane, tačno se primeti da se superkondenzator višestruko brže puni/prazni od baterije.



Slika 8. *Napon baterije i superkondenzatora*

5. ZAKLJUČAK

U radu je opisan hibridni sistem za skladištenje električne energije paralelne potpuno aktivne topologije i njegov menadžent energije.

Na kraju je opisana simulacija i dobijeni rezultati koji se podudaraju sa očekivanim rezultatima koji bi se desili u stvarnosti. Takođe, dokazano je da predstavljeni menadžment energije u rezultatima simulacije radi kao što je teorijski opisano.

6. LITERATURA

[1] Nikola Vukajlović et al, "Comparative analysis of the supercapacitor influence on lithium battery cycle life in electric vehicle energy storage", JOURNAL OF ENERGY STORAGE, Vol. 31, 2020.

[2] Lia Kouchachvili, "Hybrid battery/supercapacitor energy storage system for the electric vehicles", Journal of Power Sources, Vol. 374, pp. 237-248, 2018.

[3] Changle Xiang, Yangzi Wang, Sideng Hu, Weida Wang, "A New Topology and Control Strategy for a Hybrid Battery-Ultracapacitor Energy Storage System", Energies, Vol. 7, pp. 2874-2896, 2014.

[4] Q. Zhang, G. Li, "Experimental Study on A Semi-active Battery-Supercapacitor Hybrid Energy Storage System for Electric Vehicle Application", IEEE Transactions on Power Electronics, Vol. 35, Issue: 1, Jan. 2020.

[5] Alon Kuperman, Ilan Aharon, "Battery-ultracapacitor hybrids for pulsed current loads", Renewable and Sustainable Energy Reviews, Vol. 15, Issue: 2, 2011.

Kratka biografija:

Marko Kozomora rođen je u Beogradu 1997. godine. Osnovne akademske studije na Fakultetu tehničkih nauka u Novom Sadu iz oblasti Elektrotehnike i računarstva -Energetska elektronika i električne mašine završio je 2020.godine.
kontakt:
marko.kozomora97@gmail.com

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.85
DOI: https://doi.org/10.24867/16BE38Ilic

PREDIKCIJA POPULARNOSTI OBJAVA NA SAJTU 9GAG KORIŠĆENJEM MULTIMODALNIH PODATAKA SA FOKUSOM NA TEKSTUALNE PODATKE

POPULARITY PREDICTION OF 9GAG POSTS WITH USAGE OF MULTIMODAL DATA AND FOCUS ON TEXTUAL DATA

Nikola Ilić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Predikcija popularnosti objava na sajtu 9gag bi mogla da bude od značaja za prepoznavanje i analizu faktora koji su bitni kod popularnosti sadržaja na društvenim mrežama. Ova istraživanja mogu biti korisna pre svega za oblast marketinga. U ovom radu predstavljena je arhitektura sistema za predikciju popularnosti objava. Fokus je bio na analizi onih elemenata objava koje se tiču teksta objave (komentara, naslova,...). Isprobana su dva načina, od prvi predstavlja testiranje više različitih klasifikatora, a drugi stekovanje klasifikatora. Na kraju su predstavljeni i analizirani rezultati, a predložena su i dalja unapređenja sistema.*

Ključne reči: *Analiza sentimenta, obrada teksta, predikcija popularnosti*

Abstract – *Popularity prediction of 9gag posts could help recognize and analyze key elements of popular content on social media. This type of research could be necessary for areas such as marketing. This paper presents the system's architecture for 9gag post popularity prediction, where the focus is on analyzing the textual elements of the posts. Two approaches were used - training multiple different classifiers and classifier staking. Results are presented and analyzed at the end, and future work of the system is discussed.*

Keywords: *Sentiment analysis, text mining, popularity prediction*

1. UVOD

Svaka objava na *9gag-u* [1] ima određene karakteristike koje je prate. Neke su napravljene po određenom, prethodno ustanovljenom šablonu (takozvani *meme*). Neke su potpuno originalne. Kod nekih je glavni element slika objave, kod nekih tekst, dok je kod nekih bitna kombinacija ta dva. Sve objave mogu biti ocenjene pozitivno i negativno i na svakoj objavi se može ostaviti komentar, što doprinosi određenoj popularnosti objave. Popularnost objave uzrokuje da se objava nađe u određenoj kategoriji na sajtu na osnovu toga koliko je popularna.

Problem kojim se bavi ovaj rad jeste predikcija popularnosti neke objave na sajtu *9gag*. Odnosno, analizira se koji sve elementi koji karakterišu jednu objavu utiču na popularnost objave i u kolikoj meri. Ovo bi moglo da bude od značaja u oblasti marketinga, prvenstveno u onim delovima koji se bave reklamiranjem, kao i predstavljanjem određenog zabavnog sadržaja kako na društvenim mrežama, tako i onlajn uopšte.

Ideja prikazana u ovom radu je da se prvo izvrši ekstrakcija obeležja u vidu elemenata objave za koje je pretpostavka da su od značaja kada je u pitanju njena popularnost. Ekstrahovana obeležja se koriste za treniranje regresionog modela za predikciju popularnosti objava.

U cilju treniranja modela prikupljeni su podaci sa sajta *9gag* od kojih su glavni bili: naslov objave, *hashtag*-ovi, komentari, kao i podaci u vezi sa komentarima (broj komentara i sl.). Ovde predstavljeni klasifikator teksta je deo većeg sistema i korišćen je u kombinaciji sa klasifikatorom koji radi na osnovu slike za koji su takođe bila ekstrahovana obeležja od značaja.

Nakon ekstrakcije obeležja, ona su iskorišćena za treniranje regresionih modela. Obeležja su kombinovana na dva načina. Prvi način je da se sva obeležja konkateniraju i proslede modelu. Drugi način kombinovanja obeležja je stekovanje (eng. *stacking*) modela. Za ciljno obeležje (popularnost objave) korišćen je odnos pozitivnih i negativnih glasova neke objave.

Glavno zapažanje kada su u pitanju konačni rezultati predikcije je to da modeli nisu uspeli dovoljno dobro da se obuče na ekstrahovanim obeležjima. Za to postoji više razloga. Neki od njih su činjenica da se model u velikoj meri oslanja na broj komentara kao obeležje i da je utvrđeno da vrlo često sam naslov nema skoro nikakve veze sa samom objavom.

U poglavlju 2 će biti izložena postojeća literatura koja se bavi sličnim problemom. U poglavlju 3 biće govora o skupu podataka koji je korišćen, a u poglavlju 4 će biti detaljnije pojašnjenje arhitekture sistema i njegova implementacija. U poglavlju broj 5 će biti opisan postupak verifikacije i prikaz rezultata. Konačno, u poglavlju 6 iznesen je zaključak rada i opisane su neke ideje i mogućnosti za dalji rad koje bi mogle da doprinesu unapređenju postojećeg rešenja.

2. SRODNA ISTRAŽIVANJA

U toku pregleda literature relevantne za ovaj rad nije pronađen ni jedan rad koji se bavi problemom predikcije popularnosti objava na sajtu 9gag. Zato će u ovom poglavlju biti izložen prikaz radova koji se bave

---

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

problemom predikcije popularnosti na drugim društvenim mrežama, kao i radovi u kojima je fokus na tekst kao glavni aspekt rešavanja problema kojim se bave ti radovi.

Cilj rada [2] jeste predviđanje popularnosti objava na društvenoj mreži *Flickr*[1]. Opisani pristup prilikom predikcije koristi vizuelna obeležja, koja se dobijaju analizom slike, obeležja dobijena analizom teksta objave i društvena obeležja, poput prosečnog broja pregleda korisnika koji je objavu postavio. Korišćen je postojeći SMP-TI skup podataka [3], koji ukupno sadrži 432 hiljade objava sa društvene mreže *Flickr*. Za analizu kontekstnih i društvenih informacija korišćeni su *Random Forest* algoritam i namenski kreirana konvolutivna neuronska mreža sa šest slojeva. Analiza naslova i tagova se bazirala na upotrebi rečnika, a sentiment opisa je određen pomoću *Stanford CoreNLP*[2] biblioteke. Sentiment se ocenjuje sa pet opisnih ocena na osnovu čega se izvodi ocena opisa. Ovakvom analizom se došlo do 15 obeležja, koja se potom normalizuju i vrši se predviđanje popularnosti objave pomoću konvolutivnog modela. Za evaluaciju rešenja korišćeni su Spirmanov koeficijent korelacije, srednja apsolutna greška i srednja kvadratna greška. Zaključak opisanog rada je da je korišćenje multimodalnog pristupa bilo opravdano. Rad [2] se može smatrati najsličniji ovom istraživanju i zbog toga je dao značajne putokaze na koji način se mogu prikupljati i koristiti multimodalni podaci. Analiza objekata na slikama, naslova, tagova i sentimenta vršene su na sličan način i u ovom istraživanju.

Cilj rada [4] jeste da se utvrdi šta sve u okviru objave koja se odnosi na brend brze hrane na društvenoj mreži *Instagram*[3] utiče na to da objava bude popularnija. U radu su korišćeni parametri koje su autori nazvali *engagement parameters*, od kojih se neki odnose na sliku, a neki na tekst. S obzirom na to da je za ovaj rad relevantno samo to kako su iskorišćena obeležja u vezi sa tekstom, neće biti reči o obeležjima i metodama u vezi sa slikom. Prikupljeno je 75 hiljada objava koji se odnose na šest poznatih lanaca brze hrane i od njih je sačinjen skup podataka. Kao obeležja u vezi sa tekstom korišćeni su komentari, naslovi i *hashtag*-ovi nad kojima je izvršena analiza sentimenta. Obrada teksta je rađena slično kao i u ovom radu, gde je analiza sentimenta bila primenjena samo na komentare.

Sličan zaključak i značaj za ovo istraživanje ima rad [5], u kojem se predviđa popularnost objava na *Reddit*-u na osnovu prvih deset komentara. Problem je posmatran kao binarna klasifikacija, zbog, kako autori navode, čudne distribucije ciljnog obeležja *score*, koje predstavlja razliku pozitivnih i negativnih reakcija. Kao obeležja za klasifikaciju korišćeno je više obeležja koji se odnose na sentiment komentara, kao i dužina komentara. Prikupljeno je oko 2220 primera koji čine skup podataka. Obeležija koja su izvučena su: *maximum*, *minimum*, *lower-half average*, *upper-half average* i *average* ocena dobijena analizom sentimenta komentara. Analiza sentimenta je vršena pomoću *Stanford NLTK* biblioteke. Slično kao u [2] i ova biblioteka daje rejting za svaku rečenicu, a rejtinzi su: veoma negativno, negativno, neutralno, pozitivno i veoma pozitivno. Svakom od rejtinga je data brojna vrednost i to redom –2, -1, 0, 1, 2. Svaki komentar je dobio svoju sentiment ocenu na osnovu srednje vrednosti sentiment ocena svih rečenica u tom komentaru. Razlika pristupa predstavljenog u radu [5] i pristupa prikazanog u ovom radu je ta da je u ovom radu korišćena procentualna vrednost za meru ocene, gde se kao krajnja ocena sentimenta uzima ona koja ima najveću procentualnu vrednost. Rezultati rada [5] su dobijeni primenom nekoliko metoda, a to su: *K-Neighbors*, *Perceptron*, *Support Vector Machine* i *Logistic Regression.* Pored značaja ranih reakcija na neku objavu za njenu popularnost, rad [5] pokazuje da sentiment komentara treba uzeti u razmatranje prilikom predikcije popularnosti objava.

Cilj rada [6] jeste predikcija popularnosti onlajn peticija putem multimodalnog dubokog regresionog modela. Skup podataka korišćen u radu [6] sastoji se od oko 75 hiljada peticija prikupljenih sa sajta „Avaaz.org“. Slično kao i u ovom radu, autori evaluiraju model na osnovu tri klase modela: tekstualni model, model slike i kombinovani model slike i teksta. Modeli korišćeni za tekst su CNN regresioni model i BERT regresioni model. Eksperimenti su vršeni na skupu podataka koji je bio nasumično podeljen na trening, validacioni i test skup. Takođe, valja napomenuti i da su eksperimenti posebno vršeni na skupu podataka koji je sačinjen samo od podataka na engleskom jeziku, a posebno vršeni na skupu podataka od nekoliko različitih jezika. Srednja apsolutna procentualna greška (eng. *mean absolute percentage error* - MAPE) i Spirmanov koeficijent korelacije su korišćeni za računanje evaluacije. Autori rada [6] na osnovu dobijenih rezultata zaključuju da tekstualni model ima nešto bolje performanse u odnosu na model slike, dok kombinovani model ima nešto bolje rezultate od model teksta. Na osnovu Spirmanovog koeficijenta korelacije vrednosti za tekstualni BERT model su $\rho = 0.385$, tekstualni CNN model $\rho = 0.363$, a dok su vrednosti za model slike $\rho = 0.235$. Spirmanov koeficijent korelacije za kombinovani model je $\rho = 0.405$. Ovde naznačeni rezultati su dobijeni na osnovu skupa podataka koji se sastoji samo od podataka na engleskom jeziku. S obzirom na nešto bolje performanse BERT modela u odnosu na CNN model, za dodatnu analizu sentimenta komentara u ovom radu odabran je BERT model.

3. SKUP PODATAKA

Kako nije pronađen relevantan postojeći skup podataka, za potrebe ovog istraživanja formiran je novi, prikupljanjem podataka (eng. *scrape*) direktno sa sajta *9gag*. Prikupljene su informacije o pojedinačnim objavama, kao i informacije o svim njihovim komentarima.

Svaka objava sadrži sledeće korisne informacije:

- broj komentara,
- *timestamp* kreiranja objave,
- broj negativnih ocena (eng. *downvote*),
- broj pozitivnih ocena (eng. u*pvote*),
- URL ka slici,
- sekcija u kojoj se nalazi – *hot*, *trending* ili *fresh*,
- naslov,
- tip objave – da li je u pitanju slika ili animacija,

[1] https://www.flickr.com/
[2] https://stanfordnlp.github.io/CoreNLP/
[3] https://www.instagram.com/

Tabela 1. *Rezultati za prvi pristup na test skupu*

Metod / Mera	*Baseline*	Linearna regresija	*Elastic net*	SVR	*Random forest*	*Voting*
R^2	-0.0307	0.1332	0.1367	0.4362	0.4515	**0.4918**
ρ	/	0.431	0.4606	0.6534	0.6586	**0.6935**

Tabela 2. *Rezultati za drugi pristup na test skupu*

Podaci / Mera	Informacije sa slika	Sentiment komentara – Core NLP	Sentiment komentara - BERT	Ključne reči u naslovu	Konačna predikcija
R^2	0.0211	0.1443	0.1498	0.0014	0.4843
P	0.1255	0.3877	**0.4117**	0.002	**0.6782**

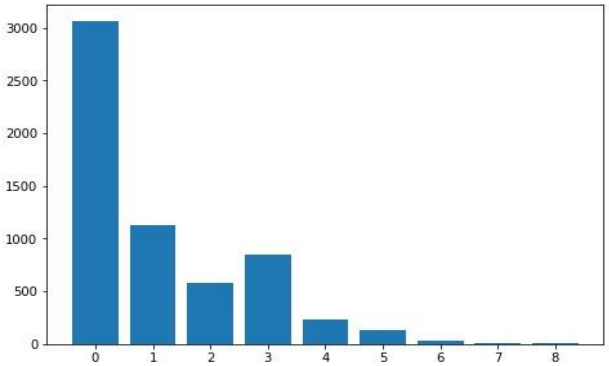
- URL ka originalnoj objavi na sajtu
- Informacije o tagovima.

Prikupljanje komentara za pojedinačne objave analogno je prikupljanju informacija o objavama. Svaki komentar sadrži sledeće korisne informacije:

- broj potkomentara,
- broj pozitivnih ocena (eng. *like*)
- broj negativnih ocena (eng. *dislike*)
- tekst komentara - ako je u pitanju slika ili animacija (GIF) u ovom polju se nalazi URL
- *permalink* – veza ka objavi kojoj komentar pripada (URL)

4. SPECIFIKACIJA I IMPLEMENTACIJA SISTEMA

Za analizu tekstualnih podataka, pretpostavka, utemeljena u analizu tekstualnih podataka u srodnim istraživanjima iz poglavlja 2, je da bi od značaja za popularnost objave bili naslovi, *hashtag*-ovi i komentari objava. Isprobano je nekoliko metoda kako bi se iz podataka pribavile informacije potrebne za izvođenje predikcije popularnosti objava.



Slika 1. *Broj objava koje imaju od 0 do 8 hashtag-ova*

4.1. Analiza *hashtag*-ova

Posmatran je odnos između broja *hashtag*-ova po objavi i broja pozitivnih, odnosno negativnih ocena, kao i broja komentara. Broj *hashtag*-ova postavljenih na pojedinačnim objavama kreće se od 0 do 8. Od 6038 prikupljenih objava, njih 3067 nije imalo nijedan hashtag, dok svega 5 objava ima 8 *hashtag*-ova. Celokupna raspodela *hashtag*-ova na ovom skupu podataka prikazana je na slici 1. Prosečan broj pozitivnih, odnosno, negativnih ocena objava sa istim brojem *hashtag*-ova varira sa tolerancijom od ±10%. Slična je situacija i sa prosečnim brojem komentara kod objava sa istim brojem *hashtag*-ova. Odstupanje od ove tolerancije nastaje kod objava sa 7 i 8 *hashtag*-ova. Međutim, ni ovde se ne da primetiti određena zavisnost. Zaključak je da je broj takvih objava previše mali, pa da zbog toga dolazi do odstupanja.

Poređena je i popularnost objava sa *hashtag*-ovima i bez njih i nema primetne razlike u njihovom odnosu. Iz svega navedenog dolazi se do zaključka da broj *hashtag*-ova ni na koji način ne utiče na popularnost objave.

4.2. Ekstrakcija ključnih reči iz naslova

Ključne reči su izvlačene TFIDF metodom na korpusu podataka koji je obuhvatao naslove svih dobavljenih objava.

Kod njihovog odabira, primećeno je da se najfrekventnije reči iz hiljadu najbolje ocenjenih objava javljaju i kao najfrekventnije reči u hiljadu najlošije ocenjenih objava, pa se tu već moglo zapaziti da one neće imati veliki značaj u konačnoj predikciji.

4.3. Određivanje sentimenta komentara

Analiza sentimenta komentara je vršena pomoću *Stanford CoreNLP* API-ja [5] za analizu i obradu prirodnog jezika, a dodatno je korišćen i pretrenirani BERT model. Rezultati koji se dobijaju pomoću *CoreNLP* API-ja su na skali od 0 do 4, gde je 0–veoma negativno, 1-negativno, 2-neutralno, 3-pozitivno i 4-veoma pozitivno. Svaka od tih pet ocena dobija procentualno vrednost, gde ukupan zbir čini 100%, a ocena koja ima najveću procentualnu vrednost se uzima kao krajnja vrednost sentimenta.

Veliki broj komentara je ocenjen kao neutralan, čak preko 80% na celom skupu, dok se taj broj kod komentara koji pripadaju samo *trending* sekciji kretao do 90,5%.

Drugi pristup analizi sentimenta komentara se zasniva na korišćenju BERT jezičkog modela (eng. *language model*), pretreniranog na skupu podataka koji se sastoji od komentara sa platforme *Google Play*[4] i koji ima tačnost od oko 88% na test skupu, prilikom određivanja sentimenta. Prilikom treniranja kao optimizator je korišćen *AdamW*, sa preporučenim parametrima iz rada [8], i *cross-entropy* kao *loss* funkcija. Kompletan model se, osim od BERT-a, sastoji i od *dropout* sloja, kako bi se sprečio *overfitting*, i linearnog izlaznog sloja sa tri izlaza. Korišćenjem ovakvog modela se dobija sentiment svakog komentara, a

[4] https://play.google.com/

kumulativan sentiment komentara na neku objavu se dobija uprosečavanjem predikcija za pojedinačne komentare, pri čemu su komentari prepoznati kao pozitivni označeni brojem 1, neutralni sa 0, a negativni sa -1. Najveći broj komentara je ponovo prepoznat kao neutralan, njih nešto manje od 21 hiljade, broj pozitivnih je nešto manji od jedanaest hiljada, a negativnih nešto preko trinaest hiljada.

4.4. Opis celokupnog sistema

Celokupan sistem predstavlja kombinaciju različitih klasifikatora slike i teksta.
Isprobana su dva različita načina dobijanja konačne predikcije modela, a to su:

1. prosleđivanje svih obeležja modelima koji se treniraju
2. stekovanje modela zasnovanih na različitim skupovima obeležja (engl. *stacking*)

Što se tiče prvog načina, isprobano je nekoliko modela. Modeli koji su se pokazali najbolje bili su: SVR, ansambl regresionih stabala i *voting* model.

Drugi pristup se sastoji od posebne predikcije popularnosti objava za podatke iz različitih izvora. Tri posebna ansambla regresionih stabala predviđaju popularnost svake objave, a potom su dobijene vrednosti, zajedno sa brojem komentara i tipom objave, korišćene kao obeležja za novi model koji donosi konačnu procenu o popularnosti objave. Model koji donosi konačnu procenu je takođe ansambl regresionih stabala. Rezultati se mogu videti u tabeli 2.

5. VERIFIKACIJA I REZULTATI

Podela skupa podataka na trening skup i test skup je vršena tako što je 80 % skupa izdvojeno za trening, a 20 % za test. Ova podela je važila za sve modele.
Najbolje rezultate kod prvog pristupa daje model koji konačnu predikciju donosi na osnovu glasova *Random forest* i SVR metoda. Kod pojedinačnih metoda najbolje rezultate ima *Random forest*, ali ni SVR ne zaostaje mnogo. Linearna regresija i *Elastic net* daju značajno lošije rezultate, ali i te metode imaju određenu prediktivnu moć. Rezultati se mogu videti u tabeli 1.
Konačna predikcija zasnovana na stekovanju modela je nešto lošija nego kod najuspešnije metode prvog pristupa, ali ne značajno. Rezultati drugog pristupa se mogu videti u tabeli 2.
Dalji rad na usavršavanju performansi ovih modela bi mogao da se sastoji od poboljšanja analiza vršenih na tekstualnim podacima. Pre svega na tome da se reši problem sarkazma kada je u pitanju sentiment komentara.

6. ZAKLJUČAK

Problem koji se rešavao u ovom radu tiče se predikcije popularnosti objava na sajtu 9gag. Rešavanje ovog problema moglo bi da bude od značaja u daljim istraživanjima u oblasti marketinga, pre svega reklamiranja i predlaganja određenog zabavnog sadržaja putem društvenih mreža.

Problem je rešavan tako što je vršena predikcija popularnosti neke objave, gde su za modele koji su trenirani korišćena obeležja dobijena analizom elemenata od značaja za neku objavu.

Za rezultate se može primetiti da nisu loši, ali ne i u kojoj meri su dobri, jer ne postoje referentne vrednosti, odnosno, rezultati iz nekih drugih dovoljno sličnih istraživanja sa kojima bi mogli da se uporede.

Nedostaci ovog rešenja kada je reč o tekstualnim podacima bi bili nemogućnost metoda da prepoznaju sarkazam, kako u naslovima, tako i u komentarima. Buduća proširenja i poboljšanja sistema mogla bi da se vrše na onim delovima tekstualne analize koji su dali najbolje rezultate, pokušajem analize teksta metodama koje nisu isprobane, a potencijalno bi mogle da daju bolje rezultate.

7. LITERATURA

[1] 9gag - https://9gag.com/
[2] MEGHAWAT, Mayank, et al. A multimodal approach to predict social media popularity. In: 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR). IEEE, 2018. p. 190-195.
[3] SMP-T1 in ACM Multimedia Grand Challenge. https://social-media-prediction.github.io/MM17PredictionChallenge/leaderboard.html, 2017
[4] MAZLOOM, Masoud, et al. Multimodal popularity prediction of brand-related social media posts. In: Proceedings of the 24th ACM international conference on Multimedia. 2016. p. 197-201.
[5] Stanford Core NLP - https://stanfordnlp.github.io/CoreNLP/
[6] TERENTIEV, Andrei; TEMPEST, Alanna. Predicting Reddit Post Popularity Via Initial Commentary. nd): n. pag, 2014.
[7] Kitayama, Kotaro et al. "Popularity Prediction of Online Petitions using a Multimodal DeepRegression Model." ***ALTA*** (2020).
[8] DEVLIN, Jacob, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

Kratka biografija:

Nikola Ilić rođen je 9.1.1996. godine u Novom Sadu. Osnovnu školu „Miroslav Antić" u Futogu, završio je 2011. godine. Iste godine upisao je gimnaziju „Isidora Sekulić" u Novom Sadu, na prirodno-matematičkom smeru, koju je završio sa odličnim uspehom. Osnovne akademske studije na Fakultetu tehničkih nauka u Novom Sadu, smer računarstvo i automatika, upisuje 2015. godine i završava 2019. godine sa prosečnom ocenom 8.57. Odbranio je diplomski rad na temu „Veb servisi i njihova implementacija upotrebom .NET tehnologije".

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/16BE39Cvejic

PREPOZNAVANJE IMENOVANIH ENTITETA U SRPSKOM JEZIKU POMOĆU TRANSFORMER ARHITEKTURE

NAMED ENTITY RECOGNITION FOR SERBIAN LANGUAGE WITH TRANSFORMER ARHITECTURE

Andrija Cvejić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Za treniranje neuronskih mreži za obradu prirodnog jezika već postoje ustaljeni šabloni i principi isprobani nad engleskim jezikom. Prirodni sled događaja jeste istraživanje i razvijanje oblasti za druge jezike. U ovom radu predstavljena je arhitektura modela za prepoznavanje imenovanih entiteta u srpskom jeziku. Ulaz model je prirodno pisan jezik. Istrenirani model daje kao izlaz verovatnoće pripadnosti reči imenovanoj kategoriji. Predloženi su koraci za poboljšanje i dalji razvoj oblasti.*

Ključne reči: *NLP, NER, Obrada prirodnog jezika, Prepoznavanje imenovanih entiteta, BERT, RoBERTa*

Abstract – *When training neural networks for natural language processing, there are already common methods and practises tried and validated on the English language. Current research is the natural sequence of development and is focused on improving models for non English languages. In this paper, we present a model architecture used for named entity recognition of the Serbian language. The model's input is natural text. The trained model's outputs are category probability of each word for each named entity.*

Keywords: *NLP, NER, natural language processing, named entity recognition, BERT, RoBERTa*

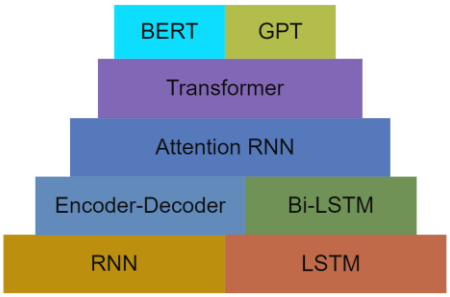
1. UVOD

Znatan deo komunikacije u savremenom društvu odvija se putem interneta, u obliku tekstualnog, video ili audio zapisa. Zahvaljujući tome, bitna odlika modernog sveta postala je povećana količina informacija, koje se prenose preko interneta.

Zbog toga javljaju se prvi pokušaji da se računarske tehnologije upotrebe u svrhe obrade teksta. Grana nauke koja se bavi takvim i sličnim problemima naziva se obrada prirodnog jezika (eng. *Natural Language Processing - NLP*).

Postoje različite tehnike za pronalaženje imenovanih entiteta; među prvima pojavile su se heurističke metode i ručno napravljena pravila (eng. *rule based*). Međutim, iakosu u izvesnim instancama pružale visoku tačnost, ove tehnike su bile preskupe i zahtevale su previše vremena za implementaciju, pri čemu i dalje nisu bile efikasne u opštim, već samo u pojedinačnim slučajevima.

Heurističke metode nasledili su kasnije statistički modeli, poput skrivenih Markovljevih lanaca (eng. *Hidden Markov Model – HMM*), koji su davali dobre rezultate, ali su zahtevali stručnjake koji bi mogli da izračunaju odgovarajuće verovatnoće za svaki jezik.



Slika 1. Hijerarhija nadogradnji NLP modela

Ovakvi pristupi imaju nekoliko ograničenja koja sprečavaju njihovo korišćenje u praksi. Iz tog razloga se prešlo na metode mašinskog učenja (eng. *Machine Learning - ML*) koje su se pokazale kao efikasnije u pronalaženju imenovanih entiteta [1]. Na slici 1. se može videti istorija napredka i nadogradnje za *ML* arhitekture. Do značajnog unapređenja u obradi prirodnog jezika dolazi sa pojavom Transformer arhitekture [2], koja se zasniva na pretpostavci da nam rekurentne neuronske mreže (RNN) nisu potrebne, već da nam je dovoljan samo aspect „pažnje“ (eng. attention). Ova nova arhitektura dala je izvanredne rezultate na planu prevođenja jezika.

U odnosu na RNN, ova arhitektura unapređuje razumevanje konteksta, ubrzava obučavanje i podržava rad sa neograničenim sekvencama, pri čemu očuvava međuzavisnost i poznavanje prethodne rečenice pomoću pažnje. To znači da se u ovom modelu može pozivati na reči u dalekoj prošlosti ili uticati na odabir trenutne reči. Istorijska nadogradnja modela može se videti na slici 1.

S obzirom na znatan napredak u oblasti obrade prirodnog jezika, pristup koji je korišćen u radu zasniva se na arhitekturi sa trenutno najboljim rešenjima kao *BERT* (eng. *Bidirectional Encoder Representations from Transformers - BERT*) [3] arhitekturi za reprezentaciju jezika [4], s tim da su uzeta u obzir i sitna unapređenja osnovnog modela, koja daju nešto bolje rezultate kao RoBERTa (*Robustly Optimized BERT Pretraining Approach*) [4]. Pored toga, predstavljena rešenja u ovome radu proizilaze iz očiglednog nedostatka obučenih modela za obradu imenovanih entiteta u srpskom jeziku [5]. Odabrana je HuggingFace infrastrukture za veb pristup modelima i

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

skupovima podataka (eng. dataset) iz navedenih razloga, u radu je korišćena BERT arhitektura, primenjena na srpski jezik, pomoću biblioteke HuggingFace specijalizovane za NLP [6][14]. Takođe se koriste algoritmi za tokenizaciju koji vrši transformaciju prosleđenih reči u vektore brojeva, konkretno o tokenizaciji pod-reči zvanoj šifrovanja bajtparova (eng. Byte Pair Encoding – BPE). Za reprezentaciju teksta u vektorskom prostoru, u radu je korišćena kontekstualna RoBERTa arhitektura.

Evaluacija prepoznavanje imenovanih entiteta izvršena je nad skupom podataka HR500k, koji je predstavljen kod [7] i korišćen kod [8], sa uporednom analizom performansi primenjenog BERT modela u istoj studiji. Pored toga, u ovom radu se primenjuje nadogradnja BERT modela zvana RoBERTa predstavljena kod [4].

U narednom poglavlju detaljnije je predstavljena postojeća literatura, korišćene tehnike opisane su u poglavlju 3, a poglavlje 4 sadrži analizu dobijenih rezultata i mogućih poboljšanja. Poglavlje 5 predstavlja sumarizaciju celog rada.

2. PRETHODNA REŠENJA

U ovom poglavlju biće predstavljeni radovi koji su se bavili prepoznavanjem imenovanih entiteta u prirodnom jeziku, sa akcentom na primenu u srpskom jeziku. U tom smislu, ukazaćemo na skupove podataka i tehnike koje su korišćene, kao i na njihove rezultate dobijene evaluacijom. Pored toga, biće objašnjen način na koji naš rad izvlači inspiraciju iz navedenih radova. Opisani radovi su dati hronološkim redosledom, gde je poseban akcenat stavljen na studiju koji su sačinili [9], a koja pružadobre smernice za celo polje i uputstva za rešavanje ovog problema. Takođe, od velikog značaja za našu problematiku su istraživanja koja su sproveli [11], zatim [10], kao i [12], rešavajući slične probleme u domenima sopstvenih jezika. Poseban značaj imaju studije koje su sačinili [13][8], jer su se autori bavili prepoznavanjem imenovanih entiteta u srpskom jeziku.

U radu [6] prezentuju HuggingFace biblioteku i platformu, čija je glavna prednost mogućnost deljenja obučenih modela i skupova podataka široj javnosti. Zbog toga, HuggingFace je postalo jedno od najpopularnijih okruženja u razvoju modela za obradu prirodnog jezika, kao i za mnoge specifične zadatke, poput prepoznavanja imenovanih entiteta. *HuggingFace* se prvenstveno bavi *Transformer* arhitekturama.

Međutim, dok je u ovoj oblasti zabeležen znatan napredak po pitanju dostupnosti obučenih modela, to ipak nije ostao slučaj za jezike koji nisu engleski, kako su pokazali Wolf, Debut, Sanh et al [11]. U svom radu autori obučavaju *BERT* model nazvan *CamemBERT*, primenjen na francuski jezik. Kao model, odabrana je nadograđena arhitektura za obučavanje, RoBERTa, uz pomoć HuggingFace biblioteke. Autori porede rezultate obučavanjana malom skupu podataka – primenjenog samo na francuski jezik, to jest jednojezični model (eng. mono-lingual model) – sa M-BERT (eng. BERT-base-multilingual-cased) više jezičnim modelom (eng. Multi-language model) obučavanja na većem skupu podataka. Zaključak ovoga rada sastoji se u tome da jednojezični model daje bolje rezultate od višejezičnog. Takođe, autori pokazuju da podaci preuzeti sa internet stranica daju bolje rezultate u odnosu na Wikipedia skupove podataka, i to sa unapređenjem od 1.5% na planu preciznosti.

Takođe rad [12] poredi višejezične modele M-BERT sa jednojezičnim modelima, u njihovom slučaju na nemačkom, švedskom, finskom, danskom i norveškom. Autori poređenje vrše na rasponu više zadataka, od generisanja teksta do specifičnih problema kao što je NER. Njihov rad takođe pokazujeda jednojezični modeli znatno bolje generišu tekst od M-BERT-a. Štaviše, autori smatraju da je M-BERT praktično beskoristan za generisanje teksta, posmatrano sa aspekta gramatičke i sintaksičke ispravnosti generisanih reči. Na pojedinačnom slučaju, NER performanse su bolje oko jedan do dva procenta kod jednojezičnih modela. Takođe, autori pokazuju da predikcija reči kod M-BERT jezičkog modela ima poteškoća sa skandinavskim jezicima zbog njihove kompleksnosti (morfološkog bogatstva) i manjeg skupa podataka za obučavanje.

U radu [8] su radu izdvojili tri ključna koraka za implementaciju svog rešenja: prvi se odnosi na odabir *BERT* modela; drugi na samo obučavanje modela da razume kontekst i gramatiku srpskog jezika, što je izvršeno na velikom skupu neobeleženih podataka; a treći na obučavanje obrade imenovanih entiteta (*NER*). Za *BERT* model autori koriste velike skupove podataka preuzete sa interneta: *srWaC[15]*, *hrWaC[15]*, *bsWaC[15]*, *cc100-hr[16] i cc100-sr[16]*. . Za obučavanje *NER* koristi se skup podataka *HR500k*. Tačnost koju su autori postigli u F1 meri varira od 0.90 do 0.96, u zavisnosti od test-skupa. Time su prikazani odlični rezultati u obradi imenovanih entiteta na srpskom jeziku, gde postoje mali skupovi podataka zbog *Transformer* arhitekture. Autori svoje rešenje porede sa drugim *M-BERT* modelom istreniranima na višejezičnim skupovima podataka.

3. METOD

U ovom poglavlju biće detaljno objašnjena postavka implementacije sistema za obučavanje modela na srpskom jeziku, kao i konkretan zadatak modela vezan za pronalaženje imenovanih entiteta

U prvom delu opisana je arhitektura rešenja. Drugi deo detaljno sagledava skupove podataka. Treći deo razjašnjava proces pretprocesiranja skupova podataka. Četvrti deo govori o obučavanju modela. Peti deo razmatra pitanja preuzimanja i ponovnog korišćenja obučenih modela u ovom radu.

3.1. Skup podataka

Skup podataka je podeljen u dva dela – (1) skup podataka za treniranje jezičkog modela i (2) skup podataka za prepoznavanje imenovanih entiteta.

Za treniranje jezičkog modela jezički model RoBERTa preuzeta su šest javno dostupna skupa podataka — Leipzig, OSCAR, srWac, hrWac, cc100-hr i cc100-sr. Prvi preuzeti srpsko-hrvatski Leipzig skup podataka je na latiničnom pismu, a podaci dolaze sa sajta Wikipedia, kao i sa .rs i .hr domena iz 2014. godine.

Drugi skup podataka je zvan OSCAR-sr, koji obuhvata samo srpski jezikna ćiriličnom pismu, dalje u radu obeležen kao OSCAR. On predstavlja prečišćenu verziju podataka preuzetih sa svih .rs domena od strane Common Crawl skupa podataka.

Treći skup podataka srWac, većinom je na latinici (16.7% ćirilica), i preuzet je sa .ba, .rs i .hr domena iz 2014. godine. Ovaj skup podataka izvorno ima veličinu 17 GB, međutim, nakon pretprocesiranja, sveden je na 3 GB.

Četvrti skup podataka hrWac, potpuno je na latinici, i preuzet je sa .hr domena iz 2014. godine. Ovaj skup podataka nakon pretprocesiranja je sveden na 6 GB.

Peti skup podataka cc100-sr, preuzet i prečišćen podaci iz 2018. godine sa Common Crawl. Koji samo sadrži ćirilicu.

Šesti skup podataka cc100-hr, preuzet i prečišćen podaci iz 2018. godine sa Common Crawl. Ovaj skup podataka je samo na latinici.

Ukupna veličina skupa podataka iznosi 28.848 milijarde tokena i 17.19 miliona rečenica, odnosno oko 41.5 GB podataka. Distibucija skupova podataka se može videti u tabeli 3.1.

Skup podataka za NER nazvan je HR500k[7] . On je spoj više skupova podataka – srpskog i hrvatskog skupa

Skup podataka	Prosečna dužina rečenice	broj tokena	broj rečenica
OSCAR	98	220 mil.	4 mil.
Leipzig	75	18 mil.	0.19 mil.
srWaC	38	490 mil.	13 mil.
hrWaC	21	1.4 mlrd.	67 mil.
cc100-sr	152	5.42 mlrd.	36 mil.
cc100-hr	148	21.3 mlrd.	143 mil.

Tabela 3.1 Skupovi podataka

podataka SETimes.RS i SETimes.HR. Skup sadrži 24800 reči i 500 hiljada tokena.

Obeležavanje imenovanih entiteta je urađeno prema IOB [17] (eng. Inside-Outside-Beginning) šemi označavanja. Opšti oblik takvog označavanja izgleda kao "prefiks-oznaka". Prefiks "I" u oznaci označava da se radi ocelini imenovanog entiteta, dok "B" prefiks indikuje da je reč o početku celine imenovanog entiteta. "B" prefiks je korišćen samo kada ga prati ista oznaka bez pojavljivanja "O" oznake između. Oznaka "O" označava da reč nije deo celine imenovanih entiteta. Ovaj način označavanja poznat je i pod nazivom BIO (eng. Beginning-Inside-Outside). Podržane oznake u korpusu su Osoba (PER), Organizacija (ORG), Lokacija (LOC) i Ostalo (MISC). Takvih oznaka u skupu podataka ima 11:'I-org', 'B-misc', 'B-per', 'B-deriv-per', 'B-org', 'B-loc', 'I-deriv-per', 'I-misc', 'I-loc', 'I-per' i 'O'. Oznaka koja zahteva dodatno objašnjenje je "deriv-per" - označava izvedenicu od imena osobe. U Tabeli 3.2 je distribucija skupa podataka za imenovane entitete HR500k prema kategorijama. U Tabeli 3.3 prikazana je distribucija rečenica i tokena za obuku, validaciju i testiranje.

Imenovani entitet	Broj tokena	Procenat skupa	Oznaka za kategoriju
Osoba	10,241	2.02%	PER
Izvedenica od osobe	319	0.06%	DERIV-PER
Lokacija	7,445	1.47%	LOC
Ogranizacija	11,216	2.21%	ORG
Razno	7,514	1.48%	MISC
Ukupno	36,735	7.25%	

Tabela 3.2 Distribucija imenovanih entiteta u skupu podataka *hr500k*

Veličina validacionog skupa iznosi 10%, a test-skupa 15% od obučavajućeg skupa, pri čemu svaka podela sadrži klase koje se prepoznaju za imenovane entitete

	Obučavanje	Validacija	Testiranje
Rečenica	20K	2K	2.7K
Tokena	550K	52K	68K

Tabela 3.3 Distribucija rečenica i tokena na skupovima podataka za obučavanje, validaciju i testiranje

3.2.Arhitektura

Arhitektura sistema je podeljena u dva dela – (1) obučavanje jezičkog modela za srpski jezik RoBERTa, u daljem toku rada nazvanim SRoBERTa, i (2) obučavanje glave modela za prepoznavanje imenovanih entiteta, u daljem toku rada nazvanim SRoBERTa-NER.

Jezički model poseduje RoBERTa arhitekturu, pa se, iz tog razloga, nećemo ponovo osvrtati na njene opšte odlike, već ćemo samo ukazati na parametre značajni za naš model, kao što su broj slojeva, broj glava za pažnju.

	M-BERT	SB-L	SB-base
Jezik	104 jezika	Sr + Hr	Srpski
Skup pod.	Wikipedia	WOL*	OL**
Obučavanja	MLM+NSP	MLM	MLM
Tokenizacija	WordPiece	BPE	BPE
Vokabular	110 hilj.	48 hilj.	48 hilj.
Dužina ulaza	512	514	514
Slojevi	12	6	6

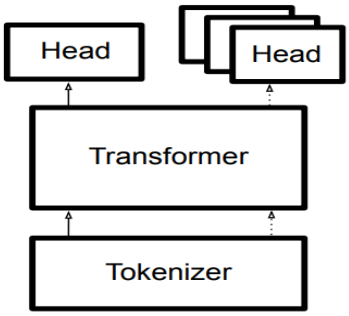
Tabela 3.4 Razlika između modela *M-BERT*, *SB-L* i *SB-base*

* WOL – podrazumeva spojena 3 izvora skupove podataka — srWac, OSCAR, Leipzig [20].

** OL- podrazumeva spojena 2 izvora skupa podataka — Oscar i Leipzig

Sam RoBERTa model je bitno uticao na odabir parametara, koji su dosta slični parametrima za BERT mrežu. Međutim, takvi parametri se i dalje znatno razlikuju od parametaramreže sa kojom vršimo poređenje, kakva je M-BERT. Sve razlike između parametara u navedenim modelima date su u Tabeli 3.4 (svi modeli SRoBERTa u tabeli su označeni sa SB). SB-XL ima istu konfiguraciju kao SB-L, ali koristi sve skupove podataka navedeni iz tabele 3.1. Takođe SB modeli se razlikuju po zadacima koje izvršavaju samo obučavanje modela maskiranog jezika (eng. Masked Language Model - MLM), dok BERT koristi i predikciju sledećih rečenica (eng. Next Sentence Prediction - NSP).

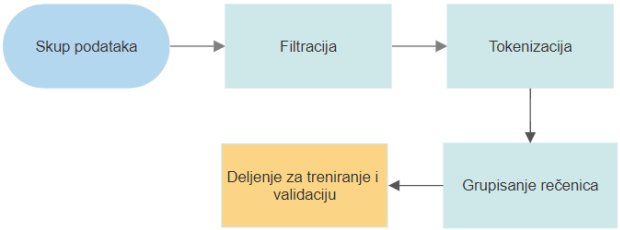
Naš model za prepoznavanje imenovanih entiteta je zasnovan na SRoBERTa modelu, gde se vrši dodavanje "glave" prema HuggingFace terminologiji, koja će da izvršava specifičan zadatak, kao što je ilustrovano na Slici 3.1. Ova glava je zadužena za NER i naziva se Token Classification, odnosno Roberta For Token Classification. Kao ulaz, model prima veličine embedovane dimenzionalnosti od E=768 vektora realnih brojeva. Glava NER modela se zasniva na jednom potpuno povezanom sloju koji za izlaz daje klase imenovanih entiteta. Klase imenovanih entitetase određuju pomoću softmax funkcije, koja utvrđuje najveću verovantnoću kojoj klasi dati entitet pripada.



Slika 3.1 Prikaz ponovne upotrebe jezičkog modela za više specifičnih zadataka [6]

3.3. Pretprocesiranje podataka za jezički model

Na Slici 3.2 predstavljeni su koraci u pretprocesiranju skupova podataka - filtracija, tokenizacija, grupisanje rečenica i deljenje za treniranje i validaciju.



Slika 3.2 Pretprocesiranje skupova podataka za jezički model

Prvi korak je podrazumevao filtraciju podataka, u ovom koraku ulaz je predstavljao originalan skup podataka, a izlaz su činile filtrirane rečenice u razdvojenim redovima. Filtracija podrazumeva izbacivanjem suvišnjih informacija (veb tagove i enkodovane veb tagove) i transformaciju u odgovarajući format. Drugi korak, tokenizacija, predstavlja proces prebacivanja podaka iz rečenica u nizove tokenizovanih informacija — niz tokena, niz ulazne pažnje i niz segmenata. Niz tokena predstavlja prebacivanje reči u niz tokena podreči dužine t; svaki token podreči je ceo broj. Reč će biti jedan token u slučaju da cela reč postoji u vokabularu. U slučaju da cela reč ne postoji u vokabularu, ona postaje niz tokena koji čine tu reč. Niz ulazne pažnje je neophodan da bi se token popunjavanja "[PAD]" razlikovao u odnosu na druge tokene prilikom ulaza, pri čemu evaluacija modela može da zanemari izlaz ovih tokena. Niz segmenata predstavlja niz celih brojeva, a vrednost se dodeljuje u odnosu na to kojoj rečenici token pripada. Na primer, ako token pripada prvoj rečenici, onda će niz na t poziciji imati vrednost 1.

Treći korak podrazumevao je grupisanje rečenica, što je bilo neophodno iz tri razloga. Prvo, da bi ulaz odgovarao predefinisanoj veličini ulaza u model, jer u prethodnom koraku tokenizacije nije vršeno odsecanje preko dužine ulaza niti popunjavanje do dužine ulaza. Drugo, uvidi do kojih su došli Kosec, Fu i Krell [18], pokazuju da je moguće udvostručiti brzinu obučavanja ukoliko se ne troši snaga računanja na „[PAD]" token. Takođe, Sun, Qiu, Xu et al. [19] su pokazali da se bolji rezultati mogu dobiti ako se pri klasfikaciji teksta ne vrši odsecanje na kraja, već je bolje sredinu. Autori su takođe pokazali da tekst pri kraju ima veći značaj po odlučivanje modela. Treće, da bi se moglo propustiti više reči u procesu obučavanja, pošto skupovi podataka sadrže i veoma duge rečenice.

Skup podataka hr500k ima 3 koraka za pretprocesiranje. Prvi korak se odnosi na transformaciju iz conllu formata u csv format, pri čemu se čuvaju informacije o rečenicama, njihovim odgovarajućim imenovanim entitetima i POS oznakama. Drugi se odnosi na tokenizaciju reči u skupu podataka. On podrazumeva da svi tokeni (pošto reči mogu da postanu više tokena) dobiju istu oznaku NER iz skupa podataka.

3.3. Obučavanje

U predloženom rešenju obučavana su četiri RoBERTa modela — *SRoBERTa* , *SRoBERTa-base* , *SRoBERTa-L* i *SRoBERTa-XL*.

Na Tabeli 3.5 su prikazani različiti parametri obučavanaja jezičkog modela.Najbolji *m*odel *SB-XL* je treniran 135h

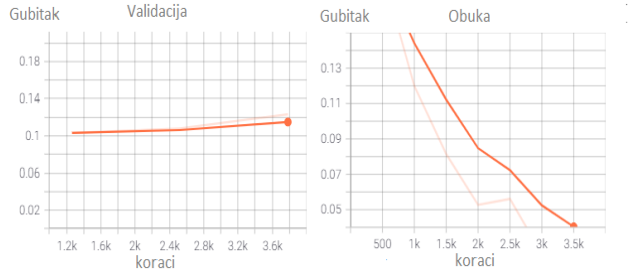
	SB	SB-base	SB-L	SB-XL
Batch size	94	180	48	50
Vreme	12 h	40h	30h	135h
Epoha	3	2	2	1
Skup Podataka	Leipzig	OL	WOL	WOLX
Broj parametra	120mil.	80mil.	80 mil.	80 mil.
Gubitak	3.2	2.9	2.6	2.2

Tabela 3.5 Razlike između obučavanih modela

na RTX 3070 sa *batch size* 50 sa gubitkom 2.2. U Tabeli 3.5 su prikazani različito obučavanje SB modela. Za ovo obučavanje su korišćeni svih 6 skupova podataka iz tabele 3.1. zajedno zvani WOLX.

U sklopu predloženog rešenja obučavana su četiri *NER* modela — *SB-NER*, *SB-base-NER*, *SB-L-NER*, *SB-XL-NER* [1] i *M-BERT-NER*. Svi modeli su obučavani na odgovarajućim *Transformer* modelima na istom skupu podataka.

Kada je reč o obučavanju modela za pojedinačni zadatak prepoznavanja imenovanih entiteta, situacija je nešto drugačija. Posle tri epohe gubitak na validacionom skupu uglavnom počinje da raste. Tada model kreće da se previše istrenira na skup podataka iz obučavajućeg seta (eng. Overfitting). Na Slici 5.1 možemo uočiti nizak obučavajući gubitak (desno), jer se kriva približava nuli, dok validacioni gubitak (levo) kreće polako da raste sa dužim obučavanjem. Dodatno obučavanje nije doprinelo boljim rezultatima.



Slika 3.3 Izgled obuke modela *SRoBERTa-L-NER*

4. REZULTATI I DISKUSIJA

U Tabeli 4.1 prikazani su rezultati obučavanja u predloženom rešenju za sve NER SB modele i M-BERT. Za evaluaciju rezultatasu korišćene prethodno opisane mere: f-mera, tačnost i odziv. F1 mera se povećava od *SRoBERTa-NER* do *M-BERT-NER* modela u rasponu od 0.66 do 0.86. Iz obučavanih modela se može primetit veliki uticaj povećanje skupa podataka sa Leipzig (18 mil. Tokena) na WOLX (28 mlrd.). Povećanje na SB-XL-NER sa jednom epohom treniranja postiže povećanu tačnost od 7%. Takođe, vidi se neophodna veličina skup podataka da se dostigne jednaka tačnost M-BERT-u. Tabele 4.2 i 4.3 prikazuju tačnost. Mada za dalje napredovanje, trebaju druge tehnike da se primene, kao što je povećavanje batch size,

Možemo zaključiti da su predložena rešenja u radu uspešno reprodukovana tačnost modela NER M-BERT čije je obučavanje opisano kod Ljubešića i Lauca [8] na hr500k skupu podataka. Međutim, naše rešenje nije uspelo poboljšati tačnost za pojedinačan zadatak sa jednojezičnim modelom, u meri u kojoj su to pomenuti autori postigli na drugim jezicima [10] [11][12].

Tokom treninga NER modela menjan je korak učenja (eng. learning rate), a najbolje rezultate ostvaruje korak između 1e-5 i 1.5e-5 za NER mrežu. Za evaluaciju NER modela korišćen je skup podataka koji su predstavili Ljubešić, Agić, Klubickaet al [7].

[1] https://huggingface.co/Andrija/SRoBERTa-XL-NER

Ime Modela	Preciznost	Odziv	F1 mera
SB-NER	0.65	0.67	0.66
SB-base-NER	0.72	0.73	0.73
SB-L-NER	0.79	0.80	0.80
SB-XL-NER	0.85	0.86	0.86
M-BERT-NER	0.87	0.86	0.86

Tabela 4.1 Rezultati metrika prilikom obučavanja NER modela

Imenovani entitet	SB	SB-base	SB-L	SB-XL	M-BERT
LOC	0.69	0.75	0.82	0.83	0.86
ORG	0.53	0.65	0.72	0.80	0.84
MISC	0.45	0.42	0.50	0.60	0.66
PER	0.61	0.72	0.80	0.85	0.86
D-PER	0.32	0.48	0.70	0.78	0.83
Ukupno	**0.56**	**0.66**	**0.73**	**0.80**	**0.83**

Tabela 4.2 F1 mera *NER* modela na test-delu skupa podataka

Imenovani entitet	SB	SB-base	SB-L	SB-XL	M-BERT
LOC	0.69	0.75	0.82	0.83	0.86
ORG	0.53	0.65	0.72	0.80	0.84
MISC	0.45	0.42	0.50	0.60	0.66
PER	0.61	0.72	0.80	0.85	0.86
D-PER	0.32	0.48	0.70	0.78	0.83
Ukupno	**0.56**	**0.66**	**0.73**	**0.80**	**0.83**

Tabela 4.3 F1 mera *NER* modela na test-delu skupa podataka

5. ZAKLJUČAK

U ovom radu predstavljen je model za prepoznavanje imenovanih entiteta u srpskom jeziku. Dobijeni rezultati nisu neposredno napredovali prepoznavanja imenovanih entiteta, ali daju zadovoljavajuće NER rezultate za jezički model u odnosu na M-BERT. Premda zabeležen je uspeh na planu razvoja jednog jezički modela u odnosu na M-BERT, koji bolje rešava problem skrivenih reči. U tom smislu, rezultati ovog rada nadovezuju se na rezultate koje su izveli Rönnqvist, Kanerva, Salakoski et al. [12],

tvrdeći da je višejezični model M-BERT praktično beskoristan za generisanje reči sa aspekta njihove semantičke i gramatičke ispravnosti.

U procesu izrade rešenja, identifikovano je nekoliko problema koji se javljaju u prepoznavanju imenovanih entiteta. Jedan od najvažnijih predstavlja nedovoljno radne memorije na grafičkim karticama, što se kod jezičkog modela manifestuje na tri plana - batch size, veličinu arhitekture (dubinu slojeva u mreži) i veličinu vokabulara. Smernice za dalje istraživanje svode se na povećanje vremena obuke jezičkog model, što je neophodno jer se tekstovi na srpskom jeziku javljaju na dva pisma, latinici i ćirlici, pa time zahtevaju više vremena za obuku kroz adekvatan skup podataka. Jednom kada se to postigne, biće moguće ostvariti bolju reprezentacija konteksta i samim time bolje rezultate NER zadatka. Istovremeno, ovo polazište potvrđuje pretpostavku da skupovi podataka srodnih jezika značajno doprinose izgradnji sveobuhvatnijeg konteksta. U daljem istraživanju ove smernice mogu biti korisne za opšti razvoj obrade prirodnog jezika, kao i, konretno, za razvoj jezičkih i NER modela.

6. LITERATURA

[1] Mansouri, A., Affendey, L. S., & Mamat, A. (2008). Named entity recognition approaches. International Journal of Computer Science and Network Security, 8(2), 339-344.

[2] 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).

[3] 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.

[4] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

[5] Krstev, C., Vitas, D., Obradović, I., & Utvić, M. (2011, July). E-dictionaries and Finite-state automata for the recognition of named entities. In Proceedings of the 9th International Workshop on Finite State Methods and Natural Language Processing (pp. 48-56).

[6] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2019). HuggingFace's Transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771.

[7] Ljubešić, N., Agić, Ž., Klubicka, F., Batanović, V., & Erjavec, T. (2018). hr500k–A Reference Training Corpus of Croatian.

[8] Ljubešić, N., & Lauc, D. (2021, April). BERTić-The Transformer Language Model for Bosnian, Croatian, Montenegrin and Serbian. In Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing (pp. 37-42).

[9] Torfi, A., Shirvani, R. A., Keneshloo, Y., Tavvaf, N., & Fox, E. A. (2020). Natural language processing advancements by deep learning: A survey. arXiv preprint arXiv:2003.01200.

[10] Virtanen, A., Kanerva, J., Ilo, R., Luoma, J., Luotolahti, J., Salakoski, T., ... & Pyysalo, S. (2019). Multilingual is not enough: BERT for Finnish. arXiv preprint arXiv:1912.07076.

[11] Martin, L., Muller, B., Suárez, P. J. O., Dupont, Y., Romary, L., de La Clergerie, É. V., ... & Sagot, B. (2019). Camembert: a tasty french language model. arXiv preprint arXiv:1911.03894.

[12] Rönnqvist, S., Kanerva, J., Salakoski, T., & Ginter, F. (2019). Is multilingual BERT fluent in language generation?. arXiv preprint arXiv:1910.03806.

[13] Šandrih, B., Krstev, C., & Stanković, R. (2019, September). Development and evaluation of three named entity recognition systems for serbian-the case of personal names. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019) (pp. 1060-1068).

[14] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2019). HuggingFace's Transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771.

[15] Ljubešić, N., & Klubička, F. (2014, April). {bs, hr, sr} wac-web corpora of Bosnian, Croatian and Serbian. In Proceedings of the 9th Web as Corpus Workshop (WaC-9) (pp. 29-35).

[16] 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.

[17] Ramshaw, L., & Marcus, M. P. (1995). Text chunking using transformation-based learn.

[18] Kosec, M., Fu, S., & Krell, M. M. (2021). Packing: Towards 2x NLP BERT Acceleration. arXiv preprint arXiv:2107.02027.

[19] Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019, October). How to fine-tune bert for text classification?. In China National Conference on Chinese Computational Linguistics (pp. 194-206). Springer, Cham.

[20] Goldhahn, D., Eckart, T., & Quasthoff, U. (2012, May). Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. In LREC (Vol. 29, pp. 31-43).

Kratka biografija:

Andrija Cvejić rođen je 1996. godine u Novom Sadu. Osnovne akademske studije završio je 2019. godine na Fakultetu tehničkih nauka, na kom brani i master rad 2021. godine iz oblasti Elektrotehnike i računarstva – Primenjene računarske nauke i informatika.
kontakt: andrijacvejic@gmail.com

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.85
DOI: https://doi.org/10.24867/16BE40Cvejic

ANALIZA RAZLIČITIH MODELA ZA PREPOZNAVANJE IMENOVANIH ENTITETA NA SRPSKOM JEZIKU

COMPARISON OF DIFFERENT APPROACHES TO NAMED ENTITY RECOGNITION IN SERBIAN LANGUAGE

Aleksandar Cvejić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Zadatak ovog rada jeste analiza različitih pristupa za prepoznavanje imenovanih entiteta (Named Entity Recognition, NER) u srpskom jeziku. Rad poredi performanse Conditional Random Fields (CRF) modela i transformer modela na NER zadatku. Kao transformer modeli korišćeni su BERT, DistilBERT i Electra transformer modeli. CRF se direktno trenira za NER zadatku, dok se transformer modeli treniraju u 3 koraka: (1) treniranje tokenizera, (2) pretreniranje generalnog jezičkog modela i (3) dotreniranje na NER zadatku. U radu se prikazuju rezultati više konfiguracija CRF modela, treniranim na različitim karakteristikama i rezultati transformer modela.*

Ključne reči: *NLP, NER, Transformeri, CRF,* prepoznavanje tokena

Abstract – *This paper aims to analyze different approaches to Named Entity Recognition (NER) in the Serbian language. The paper compares the performance of the CRF model to newer transformer models on the NER task. Applied transformer models are BERT, DistilBERT, and Electra. In the performed experiments, multiple configurations of CRF, trained on different attributes, are compared to transformer models.*

Keywords: *NLP, NER, Transformers, CRF, token classfication*

1. UVOD

NLP (eng. *Natural Language Processing*) se kao oblast razvila zahvaljujući dostupnosti veće količine podataka, kao i unapređenim u tehnikama procesiranja teksta. Naročito se povećava dostupnost nestrukturiranih podataka, zbog toga što je sve veći broj sistema digitalizovan, i veći broj ljudi imaju pristup internetu.

Ovaj rad rešava problem prepoznavanja imenovanih entiteta (eng. *Named Entity Recognition* – NER). Glavni cilj NER je prepoznavanje reči (ili skupa reči) u tekstu koje predstavljaju imenovane entitete (osobe, lokacije, organizacije i druge). Iako ovaj zadatak samostalno nije od većeg komercijalnog značaja, on predstavlja važan deo širih sistema za obradu teksta. Jedno od mesta gde bi NER model mogao primeniti zajedno sa analizom sentimenta (eng. *sentiment analysis*) jeste indirektna predikcije berze [1].

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

Zadatak ovog rada jeste analiza različitih pristupa za NER u srpskom jeziku. Jedan prikazani pristup je primena CRF [2] (eng. *Conditional Random Fields*) modela koji se tradicionalno koristi za ovaj zadatak. Isprobano je više različitih konfiguracija CRF modela, gde svaka konfiguracija predstavlja drugačiju kombinacija karakteristika korišćenih za predstavljanje podataka. CRF se pokazao kao *state-of-the-art* [3] pri obradi jezika koji nemaju bogat skup podataka (eng. *dataset*), kao što je srpski jezik. Pored ovoga, za rešavanje NER zadatka korišćeni su i dotrenirani BERT [4], DistilBERT [5] i Electra [6] transformer modeli. CRF se direktno trenira za NER zadatak, dok se transformer modeli treniraju u tri koraka: treniranje tokenizera, pretreniranje generalnog jezičkog modela (eng. *pre-training*) i dotreniranje na NER zadatku. Modeli istrenirani u radu se koriste originalne parametre i načine obučavanja kao i originalni autori transformer modela [4]–[6] i CRF rada [3].

Modeli predstavljeni u ovom radu trenirani su na jednom neanotiranom skupu podataka *oscar* [7] i dva anotirana skupa podataka, *SETimes.SR* [8] i *wikiann* [9]. Poređenja se vrše na *SETimes.SR* skupu podataka zbog upotrebe istog skupa podataka u drugim radovima [10], [11]. *Wikiann* je korišćen kao dodatni skup podataka za transformer modele, radi provere hipoteze o količini potrebnih podataka za treniranje ovako ogromnih modela.

Transformer modeli implementirani u ovom radu su javno dostupni, zajedno sa tokenizerima, generalnim jezičkim modelima i specijalizovanim modelima za NER na Huggingface [12] platformi[1].

U poglavlju 2 predstavljena je postojeća literatura, koja se bavi NER problemom. U poglavlju 3 su predstavljeni korišćeni podaci i treniranje modela. U poglavlju 4 su predstavljeni rezultati i diskusija. Poglavlje 5 predstavlja zaključak rada.

2. RELEVANTNA LITERATURA

U ovom poglavlju biće predstavljeni radovi u kojima se koriste statističke modele mašinskog učenja i moderne transformer modele za rešavanja NER problema. Biće predstavljena tačnost modela, kao i skupovi podataka nad kojima su modeli trenirani. Cilj je da se stekne okvirna predstava o oblasti, jer direktno poređenje nije moguće na različitim skupovima podataka. Ovaj rad je baziran na dve grupe studija: one koje se bave standardnim metodama za NER [3], [13], i one koje se bave transformer modelima za NER [11], [14]. Pored ovih radova, koriste se tehnike

[1] https://huggingface.co/Aleksandar

treniranja opisane u originalnim radovima o transformerima [15].

Pristup korišćen u ovom radu za odabir skupa svojstava (eng. *feature set*) je sličan pristupu u radu [3], a takođe se razmatra i isti pristup uključivanja morfoloških svojstava reči (eng. *parts of speech tags*) kod CRF modela. Autori rada [3] razmatraju NER performanse pomoću CRF modela nad češkim, kao i nad drugim jezicima. Pored toga, autori razmatraju prednost predloženog CRF modela ukoliko se koriste nadklase kod imenovanih entiteta. Kako autori u rada [3] primećuju, dodavanje morfoloških svojstava nema velikog uticaja na performanse NER zadatka, a zahteva dodatne informacije od modela. Postignuta f1 mera rada za češki jezik po *CoNLL* standardu za evaluaciju iznosi 74.08% nad malim i ograničenim češkim korpusom. Slična je situacija sa obeženim podacima za srpski jezik i dosta jezika (mali broj jezika je u povoljnoj grupi da ima pristup velikoj količini labeliranih podataka) [16].

Transformer rad [11] koji ćemo razmatrati jeste i najbitniji, jer potiče od autora samog skupa podataka koji je korišćen za evaluaciju NER modela u ovom radu. U radu se obučava *WordPiece tokenizer* (sa 32 hiljade tokena u rečniku), prvobitno predstavljen u upotrebi sa transformerima u *Google*-ovom sistemu [17]. Autori treniraju BERT transformer model (BERTić) na NER zadatku sa novim zadatkom pretreniranja generalnog jezičkog modela (korak pre dotreniranja na NER zadatak). Pristup treniranja generalnog jezičkog modela koji koriste autori rada [11] će biti primenjen i u ovom radu.

U radu [11] je prvi put uvedeno korišćenje *Electra* transformera [6]. Autori objašnjavaju da je razlog za odabir ovog načina predtreniranja (treniranje generalnog jezičkog modela – eng. *pre-training*) sadržan u boljim rezultatima po pitanju brzine konvergencije i manjoj količini resursa zahtevanih prilikom obučavanja samog transformera. Pretreniranje se postiže učenjem drugog zadatka, gde je cilj da transformer odgovori da li reč pripada originalnom tekstu ili je generisana pomoću generatora. Autori biraju ovaj pristup zbog toga što koriste manji skup podataka od ukupno 8.4 milijardi tokena (reči i ostali karakteri), koji je dosta manji od originalnog BERT modela na engleskom jeziku sa 2500 milijardi tokena. Promenjen način obučavanja se opisuje u 3.4 potpoglavlju. Ukupan broj tokena u skupu podataka korišćenim za treniranje u radu se dobija kombinacijom više jezika i više skupova podataka za jedan jezik. Autori postižu rezultate od 92.02% u f1 meri, nakon pet epoha treniranja na *SETimes.SR* i 87.92% u f1 meri na *ReLDI-sr* [18] skupu podataka. Drugi skup podataka je javno dostupan, ali bez NER tagova u trenutku izrade rada, zbog čega nije uzet u obzir u ovom radu[2].

Rešenje u ovom radu se razlikuje od dosadašnjih radova jer ograničava skup podataka za učenje transformera na jedno-jezičke modele, bez inicijalizacije modela na engleskom jeziku, radi čistog poređenja trenutnog stanja jedno-jezičkog modela na srpskom jeziku. Transformer modeli se porede sa 23 različite konfiguracije CRF modela.

[2] Nakon izrade rada, NER postale su dostupne anotacije na https://huggingface.co/datasets/classla/reldi_sr

3. METODOLOGIJA

U narednim poglavljima izloženi su skup podataka, tokenizer, arhitekture modela i trening modela.

3.1. Skup podataka

U radu se koriste 3 skupa podataka, neanotirani *oscar* [7] skup podataka i dva labelirana skupa podataka, *SETimes.SR* [8] i *wikiann* [9]. *Oscar* sadrži 645,747 jedinstvenih rečenica sa ukupno 207.5 miliona jedinstvenih tokena (*uhshuffled_deduplicated_sr*). *SETimes.SR* skup podataka sadrži 86,726 manualno anotiranih tokena, dok ima 3177 rečenica u trening skupu, 395 rečenica u validacionom skupu i 319 u testom skupu. *Wikiann* skup podataka sadrži 30 hiljada u trening skupu i po 10 hiljada za validacioni i testni skup podataka.

Anotirani skupovi podataka poštuju unutra-spolja-početak (eng. *inside-outside-beginning* – IOB) princip anotiranja podataka. Moguće klase kod *SETimes.SR*, sa obe varijante početak (B) i unutra (I), su: Osoba, Organizacija, Lokacija, Ostalo (eng. *misc*), izvedena osoba (eng. *deriv-per*) i klasa izvan skupa tagova (*O*). *Wikiann* skup podataka ima smanjeni broj klasa u odnosu na *SETimes.SR*, nema *B-misc*, *I-misc* i *B-deriv-per*.

Treba napomenuti da su svi skupovi podataka korišćeni u ovom radu samo na sprskom jeziku. Generalno se očekuje da kombinovanje sličnih jezika dovodi do unapređenja performansi [16].

3.2. WordPiece tokenizer

Sva tri transformer modela proučena u ovom radu koriste *WordPiece* način reprezentacije podataka. Ovaj algoritam je predstavljen u radu [19], ali je suštinski veoma sličan GPT-2 upotrebi sa *Byte-Pair Encoding* (BPE).

WordPiece se inicijalizuje tako što prvobitno uključi sve prisutne karaktere u rečnik (vokabular) u trening skupu (u ovom slučaju *oscar*), a potom vremenom uči delove reči pomoću pravila spajanja prethodno postojećih delova. Ideja je da se izabere *WordPiece* model koji rezultuje minimalnim brojem delova reči (eng. *word pieces*) u celom trening skupu (uz dodatne specijalne simbole). U odnosu na BPE, *WordPiece* tokenizer povećava verovatnoću pojave podataka nakon što su dodati u rečnik, umesto frekvencije reči, koja se koristi u BPE-u. Ovo je ekvivalentno pronalaženju para simbola, čija je verovatnoća veća u spojenoj formi u poređenju sa zbirom individualnih verovatnoća. Ovo znači da algoritam proverava da li vredi spojiti dva simbola (npr. „a“ i „b“), na osnovu računanja gubitaka koji nastaju spajanjem („ab“). Pristup je sličan *Unigram* principu odlučivanja modela koje tokene treba da spoji, ali razlika je u tome što je *WordPiece* „pohlepan“ (eng. *greedy*) algoritam, koji pokušava da nađe najbolje parove za spajanje u svakom koraku iteracije.

3.3. Tradicionalni model - CRF algoritam

CRF algoritam je baziran na neusmerenim grafičkim modelima. Jednostavni CRF lanci su kao *state-of-the-art* za NER zadatak među statističkim modelima mašinskog učenja. Mogu se koristiti i kao poslednji sloj transformera [14], [20]. CRF je definisan i analiziran u [2], [3], [13], koje treba konsultovati za više detalja.

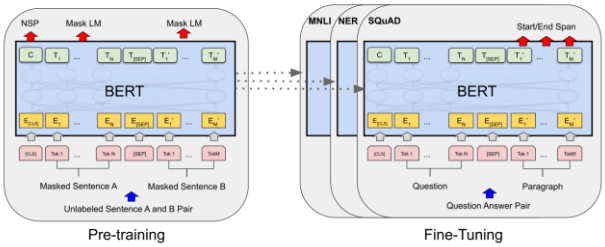
U ovom radu će se trenirati kombinacija različitih karakteristika po uzoru na [21], poput *word.lower()*, *word[-3:]*, *word.isupper()*, itd.

3.4. Transformer model – BERT

Bidirectional Encoder Representations from Transformers, odnosno BERT [4] je transformer model za modelovanje reprezentacije jezika. Postoje dva koraka u BERT implementaciji opisana u radu koji uvodi model: predtreniranje (eng. *pre-traning*) i dotreniranje (eng. *fine-tuning)*, kao što je prikazano na Slici 1.

Prvi korak podrazumeva treniranje modela uz pomoć nenadgledanog učenja na većim skupovima podataka. Ovo podrazumeva treniranje *WordPeice* tokenizera sa rečnikom od 30 000 tokena, nakon čega se trenira generalni jezički model na istom skupu nelabeliranih podataka.

Na kraju se koristi jedan od anotiranih skupova podataka da se istrenira na NER zadatku.



Slika 1. Prikaz dva koraka prisutna kod BERT transformera. Levo: prvi korak nenadgledanog obučavanja; desno: dotreniranje na specifičnom zadatku. Preuzeto iz [4].

Prilikom treniranje generalnog jezičkog modela pomoću nendagledanog učenja koristi se maskirani pristup, gde se umesto delova ulazne sekvence koristi [MASK] token za koji model treba da pretpostavi koji je token treba da bude.

3.5. Transformer model - DistilBERT

DistilBERT [5] transformer kao glavnu ideju koristi destilaciju znanja (eng. *knowledge distillation* [22]). Ovo podrazumeva specijalni režim obučavanja kompaktnog modela (studenta) u odnosu na veći (učitelj) model. Manji model je naučen da reprodukuje ponašanje već istreniranog celog modela. Rad pokazuje da se sa 40% manje parametara može održati 97% performansi mereno BLUE metrikom evaluacije (u odnosu na BERT model). Ovo znači da DistilBERT ima ukupno 66 miliona parametra, u odnosu na 110 miliona u BERT modelu.

3.6. Transformer model – Electra

ELECTRA [6] („*Efficiently Learning an Encoder that Classifies Token Representations Accurately*“) je transformer model koji menja način obučavanja od maskiranog pristupa do novog zadatka, gde se pogađa da li je token zamenjen ili pripada originalnom korpusu. Ovo je postignuto tako što se koristi jedna mala pomoćna mreža (generator). Prilikom treniranja transformer modela, menja se deo tokena tako što se uzimaju uzorci iz generator modela, dok se sam generator trenira istovremeno dok treniramo glavni model.

4. REZULTATI I DISKUSIJA

Prilikom izrade rada istrenirano sa konačnim konfiguracijama ukupno tri generalna jezička modela i šest specifičnih NER modela za transformere i 23 konfiguracije CRF modela.

Tabela 1 prikazuje rezultate treniranja specifičnog NER zadatka u 20 epoha dotreniranja. Bolji rezultati većeg skupa podataka *wikiann* ukazuju na manjak labeliranih podataka kod *SETimes.SR* skupa podataka.

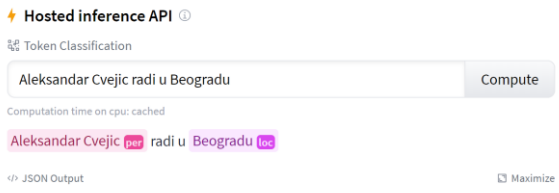
Model	**F1**	**Odziv**
BERT – *setimes.sr*	83.45%	84.65%
DistilBERT – *setimes.sr*	**84.92%**	**86.17%**
Electra – *setimes.sr*	83.46%	81.26%
BERT – *wikiann*	89.95%	90.82%
DistilBERT – *wikiann*	89.84%	**91.00%**
Electra – *wikiann*	**90.10%**	90.87%

Tabela 4.2 Rezultati dobijeni na dva skupa podataka sa različitim modelima.

Model	**Avg. F1**	**Makro F1**
BERT – *setimes.sr*	83.50%	-
DistilBERT – *setimes.sr*	84.44%	-
Electra – *setimes.sr*	81.13%	-
CRF – *setimes.sr* bez POS	84.2%	71.45%
CRF - *setimes.sr* bez isupper	88.58%	81.09%
BERTić [11] – *setimes.sr*	92.02%	-
CRF Janes [10] – *setimes*.sr	-	78.10%

Tabela 2 Rezultati dobijeni na dva skupa podataka sa različitim modelima.

Primer upotrebe modela je prikazan na slici 3. Rezultati konačnog CRF obučavanja, zajedno sa poređenjem sa rezultatima radova [10], [11] se mogu videti u tabeli 2. Najbolji CRF model je bolji od *Janes* modela [10], a lošiji od višejezičkog modela prikazanog u radu [11].



Slika 3 Primer upotrebe inference DistilBERT modela pomoću *HuggingFace API*-a, gde je zadatak pronalaženje imenovanih entiteta.

5. ZAKLJUČAK

U ovom radu predstavljeni su različiti modeli koji vrše prepoznavanje imenovanih entiteta (NER). Porede se tradicionalni pristupi NER problemu sa modernim transformer pristupima NER-u.

Rezultati dobijeni u radu imaju lošije performanse od *state-of-the-art* za engleski jezik i čak višejezičkog modela BERTić [11], ali treba imati u vidu količinu podataka korišćenu za treniranje. Razlike u transformer modelima su zanemarljive u odnosu na razlike između upotrebe većeg skupa podataka poput *wikiann*. Poznato je da veće mreže u NLP-u postižu bolje performanse, ali su potrebni podaci da se obuče takve velike mreže poput GPT-3.

6. LITERATURA

[1] R. P. Schumaker and H. Chen, "Textual analysis of stock market prediction using breaking financial news: The AZFin text system," *ACM Trans. Inf. Syst.*, vol. 27, no. 2, 2009, doi: 10.1145/1462198.1462204.

[2] lafferty john, M. Andrew, and C. N. P. Fernando, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," *ICML '01 Proc. Eighteenth Int. Conf. Mach. Learn.*, vol. 2001, no. June, pp. 282–289, 2001, doi: 10.29122/mipi.v11i1.2792.

[3] M. Konkol and M. Konopík, "CRF-based Czech named entity recognizer and consolidation of Czech NER research," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8082 LNAI, pp. 153–160, 2013, doi: 10.1007/978-3-642-40585-3_20.

[4] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, no. Mlm, pp. 4171–4186, 2019.

[5] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," pp. 2–6, 2019, [Online]. Available: http://arxiv.org/abs/1910.01108.

[6] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators," pp. 1–18, 2020, [Online]. Available: http://arxiv.org/abs/2003.10555.

[7] P. J. Ortiz Suárez, L. Romary, and B. Sagot, "A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages," pp. 1703–1714, 2020, doi: 10.18653/v1/2020.acl-main.156.

[8] V. Batanović, N. Ljubešić, T. Samardžić, and T. Erjavec, "Training corpus SETimes.SR 1.0." 2018, [Online]. Available: http://hdl.handle.net/11356/1200.

[9] A. Rahimi, Y. Li, and T. Cohn, "Massively multilingual transfer for NER," *ACL 2019 - 57th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf.*, pp. 151–164, 2020, doi: 10.18653/v1/p19-1015.

[10] D. Fišer, N. Ljubešić, and T. Erjavec, "The Janes project: language resources and tools for Slovene user generated content," *Lang. Resour. Eval.*, vol. 54, no. 1, pp. 223–246, 2020, doi: 10.1007/s10579-018-9425-z.

[11] N. Ljubešić and D. Lauc, "BERTić -- The Transformer Language Model for Bosnian, Croatian, Montenegrin and Serbian," *Proc. 8th Work. Balto-Slavic Nat. Lang. Process. (BSNLP 2021)*, no. 1, pp. 37–42, 2021, [Online]. Available: https://www.aclweb.org/anthology/2021.bsnlp-1.5.pdf.

[12] T. Wolf *et al.*, "Transformers: State-of-the-Art Natural Language Processing," pp. 38–45, 2020, doi: 10.18653/v1/2020.emnlp-demos.6.

[13] A. Cvejić, K. Grujić, A. Cvejić, M. Marković, and S. Gostojić, "Automatic Transformation of Plain-text Legislation into Machine-readable Format," *Proc. 11th Int. Conf. Inf. Soc. Technol.*, pp. 50–55, 2021.

[14] M. Arkhipov, M. Trofimova, Y. Kuratov, and A. Sorokin, "Tuning Multilingual Transformers for Language-Specific Named Entity Recognition," 2019, no. August, pp. 89–93, doi: 10.18653/v1/w19-3712.

[15] A. Vaswani *et al.*, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017.

[16] P. Joshi, S. Santy, A. Budhiraja, K. Bali, and M. Choudhury, "The State and Fate of Linguistic Diversity and Inclusion in the NLP World," pp. 6282–6293, 2020, doi: 10.18653/v1/2020.acl-main.560.

[17] Y. Wu *et al.*, "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation," pp. 1–23, 2016, [Online]. Available: http://arxiv.org/abs/1609.08144.

[18] M. Miličević and N. Ljubešić, "Tviterasi, tviteraši or twitteraši? Producing and analysing a normalised dataset of Croatian and Serbian tweets," *Slov. 2.0 empirical, Appl. Interdiscip. Res.*, vol. 4, no. 2, pp. 156–188, 2016, doi: 10.4312/slo2.0.2016.2.156-188.

[19] M. Schuster and K. Nakajima, "Japanese and Korean voice search," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 1, pp. 5149–5152, 2012, doi: 10.1109/ICASSP.2012.6289079.

[20] H. Yan, B. Deng, X. Li, and X. Qiu, "TENER: Adapting Transformer Encoder for Named Entity Recognition," 2019, [Online]. Available: http://arxiv.org/abs/1911.04474.

[21] M. Marcińczuk and M. Janicki, "Optimizing CRF-based model for proper name recognition in Polish texts," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012, vol. 7181 LNCS, no. PART 1, pp. 258–269, doi: 10.1007/978-3-642-28604-9_22.

[22] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," pp. 1–9, 2015, [Online]. Available: http://arxiv.org/abs/1503.02531.

Kratka biografija:

Aleksandar Cvejić rođen je 1996. godine u Novom Sadu. Osnovne akademske studije završio je 2019. godine na Fakultetu tehničkih nauka, na kom brani i master rad 2021. godine iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika.
kontakt: cvejicaca@gmail.com

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.85
DOI: https://doi.org/10.24867/16BE41Tomic

DETEKCIJA RESPIRATORNIH ANOMALIJA DUBOKIM UČENJEM

DEEP LEARNING DETECTION OF RESPIRATORY ANOMALIES

Nikola Tomić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Respiratorne bolesti izazivaju ogroman zdravstveni, ekonomski i društveni teret i treći su vodeći uzrok smrti širom sveta i značajan teret za javni zdravstveni sistem. Ulažu se značajni istraživački napori posvećeni poboljšanju rane dijagnoze i rutine praćenja pacijenata sa respiratornim oboljenjima, kako bi se omogućile pravovremene intervencije. Respiratorne anomalije koje se mogu prikupiti stetoskopom se klasifikuju kao diskontinualno (pucketanje) ili kontinuirano (piskanje). Takve anomalije u formatu snimljenog zvuka mogu se prikazati i vizuelno u formi spektrograma. Kako su konvolucione neuronske mreže dominatne u polju klasifikacije slika, moguće ih je obučiti i za detekciju respiratornih anomalija. U ovom radu predstavljeni pristup korišćenjem konvolucione neuronske mreže predstavlja značajno poboljšanje u odnosu na prethodne pristupe i dobijeni rezultati su među najboljima nad treniranim skupom podataka.*

Ključne reči: *CNN, ICBHI, Konvolucione neuronske mreže*

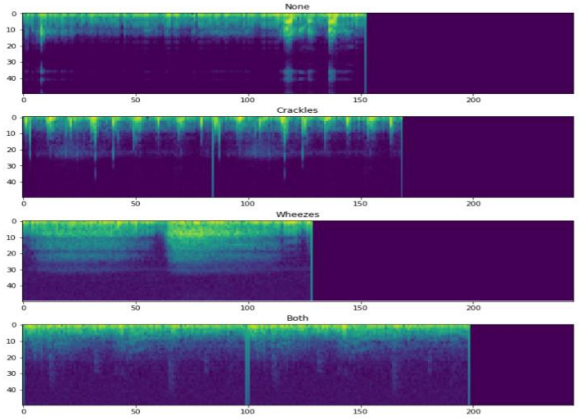
Abstract – *Respiratory diseases cause huge health, economic and social burden and are the third leading cause of death worldwide and a significant burden on the public health system. Therefore, significant research efforts have been made to improve the early diagnosis and routine monitoring of patients with respiratory diseases to enable timely interventions. Respiratory anomalies collected with a stethoscope are classified as discontinuous (cracking) or continuous (wheezing). Such anomalies in the format of recorded sound can also be displayed visually in the form of spectrograms. As convolutional neural networks are dominant in image classification, it is possible to train them to detect respiratory anomalies. The approach presented in this paper based on convolutional neural networks represents a significant improvement over previous approaches and the results obtained among the best on the trained data set.*

Keywords: *CNN, ICBHI, Convolutional Neural Networks*

1. UVOD

Sprovedena su mnoga istraživanja usredsređena na auskultaciju i karakteristike zvukova respiratornog sistema – kako su direktno povezani sa kretanjem vazduha, prome ne unutar plućnog tkiva i položaj sekreta unutar traheobronhijalnog stabla. To ih čini vrednim pokazateljem respiratornog zdravlja i respiratornih poremećaja [1].

Dijagnoza i praćenje zasnovano na auskultaciji respiratornih stanja u velikoj meri zavise od prisustva prigodnih zvukova i na izmenjenom prenosu karakteristike zida grudnog koša. Pucketanja su diskontinualne, eksplozivne i nemuzičke slučajnosti koji se često javljaju kod kardiorespiratornih oboljenja [3]. Obično se klasifikuju kao fini i grubi pucketavi po njihovom trajanju, glasnoći, tonu, trenutka pojave u respiratornom ciklusu, odnosa prema kašljanju i promeni pozicije osluskivanja [4]. Zvižduci su prigodni respiratorni zvuci koji obično traju duže više od 250 ms. Uobičajeni su klinički znak kod pacijenata sa opstruktivnim bolestima disajnih puteva kao što su astma i hronična opstruktivna plućna bolest [5]. Moguće klase predikcije i njihov prikaz u formatu spektrograma na mel skali prikazan je na slici 1.



Slika 1. *Primer spektrograma zvuka bez anomalija, pucketanja, zvižduka i pucketanja i zvižduka prikazani odozgo nadole u istom redosledu.*

Poslednjih godina postignuti su značajni rezultati u obradi slika pomoću neuronskih mreža. Ovo se delimično može pripisati odličnim performansama dubokih konvolucionih neuronskih mreža za detektovanje i transformaciju informacija na apstraktno visokom nivou. Kako se zvuk može prikazati u vidu spektograma kao dvodimenzionalni prikazi audio frekvencijskih spektara u jedinici vremena, moguće je analizirati i obraditi ga pomoću konvolucionih neuronskih mreža. U toku procesa optimizacije hiperparametara sistema razmatrane su dve različite arhitekture konvolucionih mreža. Arhitektura koja je specijalizovana za klasifikaciju zvuka, odnosno, ljudskog govora, davala je u proseku bolje rezultate u odnosu na

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

arhitekturu namenjenu za klasifikaciju slika kao sa *ImageNet* [2] takmičenja .

U narednom poglavlju detaljnije je predstavljena postojeća literatura, kao i poređenje sa našim rešenjem. Arhitektura modela i korišćene tehnike opisane su u poglavlju 3, a poglavlje 4 sadrži analizu dobijenih rezultata i mogućih poboljšanja. Poglavlje 5 predstavlja sumarizaciju celog rada.

2. PRETHODNA REŠENJA

U radu [6] predstavljen je metod zasnovan na skrivenim Markovljevim modelima (*Hidden Markov Models*, HMM) u kombinaciji sa Gausovim modelima smeše (*Gaussian Mixture Models*, GMM) za klasifikaciju respiratornih zvukova. Sličnosti u pogledu predprocesiranja signala u opisanom radu uključuju smanjenje uzorkovanja na 4 kHz i filtriranje frekvencija *bandpass* filterom kako bi se smanjio uticaj spoljnjih smetnji i šuma. Ulazne karakteristike zvuka su cepstralni koeficijenti mel-frekvencije (*Mel Frequency Cepstral Coefficients*, MFCC) ekstrahovani u opsegu između 50 Hz i 2000 Hz u kombinaciji sa njihovim prvim derivatima. Najbolji rezultat postignut u drugoj fazi evaluacije *ICBHI Challenge* je 39,56. Ovakva evaluacija je direktno uporediva sa pristupom predloženim u ovom radu jer sledi tačno specificirane podele test i trening skupa koje predlažu pravila ICBHI, kao i zvaničnu formulu bodovanja. Rad je osvojio drugo mesto na takmičenju, i predstavlja odličnu referentnu tačku za dalje napredovanje i upoređivanje sa *deep learning* pristupima.

U radu [7] je predložena konvoluciona neuronska mreža za detekciju piskanja nad plućnim zvukovima. Autori tvrde da postiže tačnost od 99% i 0,96 AUC u skupu podataka koji su prikupili. Takođe tvrde da pristup predstavlja trenutni *state-of-the-art* za detekciju piskanja i da je robustan na pomeranja po vremenskoj osi i na spoljašnje smetnje. Korišćeni podaci u radu [7] prikupljeni su iz nekoliko skupova podataka iz više izvora i više od osam puta manji od ICBHI skupa u pogledu broja respiratornih ciklusa. Akcenat u eksperimentisanju gde su se autori posvetili doprinosu augmentacije i uticaja na krajnje performanse. Predloženo rešenje je rezultovalo značajnim poboljšanjima i prednostima u pogledu robusnosti modela u odnosu na druge statističke modele kao što su vektorske mašine (SVM), K-najbližih suseda (KNN), Logistička regresija (LR), *Gradient Boosting Machine* (GBM), *Random Forests* (RF) itd.

U radu [8] za klasifikaciju respiratornih anomalija (pucketanja i zvižduka), takođe su evaluirani su pristupi zasnovani na konvolucionim neuronskim mrežama. Autori su testirali i ocenili performanse nekoliko arhitektura konvolucionih mreža koje su bile razvijane za potrebe klasifikacije slika nad skupom podataka *ImageNet*, kao što su*: AlexNet, VGG16* i *ResNet-50*, gde je *VGG16* arhitektura sa F1 merom od 0.55 značajno poboljšanje u odnosu na prethodne pristupe na ICBHI skupom podataka.

3. METOD

U narednim poglavljima izloženi su skup podataka, arhitektura i trening modela.

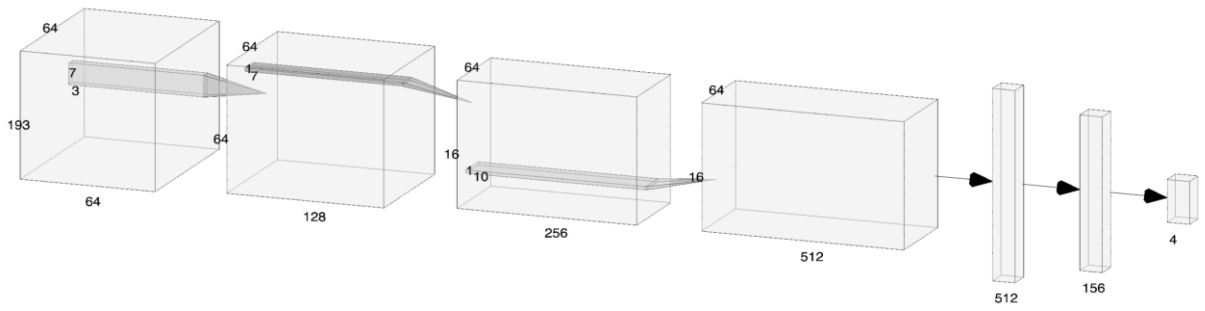
3.1. Skup podataka

Konferencija o bio medicinskoj i zdravstvenoj informatici (ICBHI) je prvi naučni izazov organizovan je sa glavnim ciljem razvoja algoritama sposobnih za karakterizaciju snimaka respiratornog zvuka izvedenih iz kliničkog i ne kliničkog okruženja. Bazu podataka kreirala su dva istraživačka tima u Portugalu i Grčkoj, a ona uključuje 920 snimaka pribavljenih od 126 pacijenata. Zabeleženo je ukupno 6898 ciklusa disanja. Baza podataka sastoji se od ukupno 5,5 sati snimaka od kojih 1864 sadrži pucketanje, 886 sadrži zvižduke, a 506 sadrži i pucketanje i piskanje. Svaki zvuk je sniman nad jednoj od sedam mogućih pozicija na gornjem delu tela pacijenta. Nivoi buke u nekim ciklusima disanja su visoki, što simulira stvarne životne uslove.

3.2. Arhitektura

Arhitektura konvolucione neuronske mreže inspirisana je drugo plasiranom arhitekturom rešenja na *Kaggle's Tensorflow speech recognition competition 2017* [19]. Glavna ideja ove arhitekture bila je stvaranje modela koji bi različito tretirao vreme i frekvenciju. Arhitektura je prikazana na slici 2.

Svaki blok prikazan na slici 2 predstavlja konvolucioni



Slika 2. *Arhitektura konvolucione neuronske mreže sa prikazanim dimenzijama svakog sloja i filtera*.

sloj praćen *max-pooling* slojem sa nacrtanim dimenzijama filtera. Poslednja tri sloja predstavljaju potpuno povezane slojeve.

Sledi opis svakog sloja iz opisane arhitekture – nazvan imenom klase, sa tačnim parametrima:

1. *Conv2D (64, [7,3])* - Filter šumova i izdvajanje osnovnih obeležja
2. *MaxPool ([1,3])* - Vraćanje na $N/3$ frekvencijske funkcije, gde *N* predstavlja broj mel filtera
3. *Conv2D (128, [1,7])* - Lokalno prepoznavanje šablona po frekventnom opsegu
4. *MaxPool ([1,4])* - Dozvoljava varijacije kod zvuka pacijenta [20]
5. *Conv2D (256 [1,10])* - Izdvajanje funkcija za svaki od frekvencijskih opsega različito i kompresija frekvencijske dimenzije
6. *Conv2D (512, [7,1])* - Traženje povezanih šablona u kratkom vremenskom interval
7. *GlobalMaxPool* - Povezivanje svih komponenti
8. *Dense (256) + Dropout* - Tumačenje obeležja
9. *Dense (4) + Softmax* - Izlazni sloj.

U datom opisu *Conv2D* predstavlja konvolucioni sloj, a *Dense* potpuno povezane slojeve.

3.3. Trening

Kompletan sistem (hiperparametri modela i proces augmentacije i predprocesiranja) deo je petlje optimizacije hiperparametara implementirane uz pomoć *Hyperopt python* biblioteke [10]. *Tree of Parzen Estimators* je odabrani algoritam koji se koristi za optimizaciju hiperparametara zbog većeg broja hiperparametara koji generišu široki prostor za pretragu. Funkcija koja se optimizuje je zvanična formula bodovanja *ICBHI 2017* [11]. Svaki eksperiment (skup hiperparametara) je obučavan maksimalno 35 epoha zbog velikog prostora pretrage i ograničenih resursa. Parametri treniranja kao što su stopa učenja (*learning rate*) i *batch size* variraju po vrednostima koje su izabrane empirijskim opsezima a konkretne vrednosti izabrane algoritmom optimizacije.

4. REZULTATI I DISKUSIJA

Model sa najboljim performansama dobijen optimizacijom hiperparametara pod nazivom *„iconic-flower-150“* prikazan u tabeli 1. Tabela 1 takođe predstavlja poređenja dobijenih rezultata sa najboljim radovima nad istim skupom podataka.

Iz kolona *sensitivity* i *specificity* tabele 1, može se zaključiti da model ima problem sa razlikovanjem zdravih respiratornih ciklusa od ostale tri oznake. Ovaj problem je posebno uočljiv kod piskanja i očekivan je kod dosta ređe klase pucketanja i piskanja, što je potvrđeno matricom konfuzije.

ICBHI 2017 je izuzetno izazovan skup podataka sa audio zapisima koji su snimljeni u ne idealnim, realnim uslovima sa dosta smetnji i šuma. Napredne tehnike za procesiranja digitalnog signala u smislu suzbijanja šuma bi sa velikom verovatnoćom bile od koristi u ovakvom okruženju. Ideja za potencijalno poboljšanje otpornosti mreže na šum jeste mogućnost treniranja generativne mreže (*Generative Adverserial Network*, GAN) koja bi proizvodila dinamički šum i ubacivala u snimke. Velika razlika u pacijentima (godine, veličina, pol, itd.) i različita oprema kojom su zapisi snimani čini ovaj skup podataka još izazovnijim. U ovakvom slučaju veća količina podataka bi svakako bila značajna u pogledu poboljšanja performansi *deep learning* modela, što se pokazalo i u eksperimentima sa agresivnijom augmentacijom retkih klasa. Korišćenjem biblioteke za optimizaciju hiperparametara uz veći broj eksperimenata otkriveni su značajniji parametri u celom sistemu, međutim, treba napomenuti da su se zbog vremenskih i računarskih ograničenja svi modeli trenirali maksimalno 35 epoha. U radu [8] model sa kojim su rezultati poređeni u tabeli 1 bio je treniran 150 epoha nad istim ICBHI skupom podataka. Imajući u vidu da su trenirani modeli i dalje konvergirali i funkcija greške se smanjivala, dužim treniranjem bi se potencijalno dobili još bolji rezultati.

5. ZAKLJUČAK

U ovom radu je predložen pristup dubokog učenja pri detekciji i klasifikovanju respiratornih anomalija. Respiratorne bolesti izazivaju ogroman zdravstveni, ekonomski i društveni teret, treći su vodeći uzrok smrti širom sveta i značajan teret za globalno zdravstveni sistem.

Poslednjih godina postignuti su značajni rezultati u obradi slika pomoću konvolucionih neuronskih mreža. Ovo se delimično može pripisati odličnim performansama dubokih konvolucionih neuronskih mreža za detektovanje i transformaciju informacija na apstraktnom nivou kod slika. Kako se zvuk može prikazati u vidu spektograma kao dvodimenzionalni prikazi audio frekvencijskih spektara u jedinici vremena, moguće je analizirati i obraditi ga pomoću konvolucionih neuronskih mreža. U ovom radu, kao i u relevantnoj literaturi, obavljena su poređenja i opisane prednosti *deep learning* pristupa ovom problemu u odnosu na tradicionalne statističke modele sa ručnim odabirom obeležja. Rezultati dobijeni u ovom radu trenutno predstavljaju najbolje rangirano rešenje po pravilima *ICBHI 2017* takmičenja. Rezultati su takođe upoređeni sa sličnim *deep learning* pristupom nad istim skupom podataka. Međutim, treba napomenuti da se u pomenutom radu nije koristio zvanični test skup za evaluaciju, sto može dovesti do optimističnih rezultata

Tabela 1. *Poređenje dobijenih rezultata u ovom radu (poslednji red) sa rezultatima iz radova [6] i [8].*

	Sensitivity	**Specificity**	**Precision**	**F1 score**	**ICBHI score**
Demir et al.	0.53	0.83	0.57	0.55	/
Jakovljević et al.	0.41	0.52	/	/	36.98
iconic-flower-150	**0.56**	**0.55**	**0.54**	**0.54**	**53.4**

kao što je napomenuto u radu [6]. Arhitektura koja je predložena u ovom rešenju inspirisana je konvolucionom neuronskom mrežom dizajniranom za prepoznavanje ljudskog govora. Eksperimentima u ovom radu pokazano je da takva specifična arhitektura ima prednosti u odnosu na *AlexNet* arhitekturu koja je dizajnirana pre svega za klasifikaciju slika. Ova prednost može se pripisati pažljivo osmišljenim slojevima i dimenzijama filtera koji različito tretiraju vremensku i frekvencijsku osu.

6. LITERATURA

[1] Gibson GJ, Loddenkemper R, Lundbäck B, Sibille Y. Respiratory health and disease in Europe: the new European Lung White Book. Eur Respir J. 2013;42:559–63.

[2] Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." International journal of computer vision 115.3 (2015): 211-252.

[3] Piirila P, Sovijarvi AR. Crackles: recording, analysis and clinical significance. Eur Respir J. Eur Respiratory Soc; 1995;8(12):2139–48.

[4] Sarkar M, Madabhavi I, Niranjan N, Dogra M. Auscultation of the respiratory system. Ann Thorac Med. Medknow Publications; 2015;10(3):158.

[5] Sovijarvi ARA. Characteristics of breath sounds and adventitious respiratory sounds. Eur Respir Rev. 2000;10:591–6.

[6] Jakovljević, Nikša, and Tatjana Lončar-Turukalo. "Hidden markov model based respiratory sound classification." International Conference on Biomedical and Health Informatics. Springer, Singapore, 2017.

[7] Kochetov, Kirill, et al. "Wheeze detection using convolutional neural networks." EPIA Conference on Artificial Intelligence. Springer, Cham, 2017.

[8] Demir, Fatih, Abdulkadir Sengur, and Varun Bajaj. "Convolutional neural networks based efficient approach for classification of lung diseases." Health information science and systems 8.1 (2020): 1-8.

[9] https://www.kaggle.com/c/tensorflow-speech-recognition-challenge

[10] https://hyperopt.github.io/hyperopt/

[11] https://bhichallenge.med.auth.gr/

Kratka biografija:

Nikola Tomić rođen je 1996. godine u Smederevu. Osnovne akademske studije završio je 2019. godine na Fakultetu tehničkih nauka, na kom brani i master rad 2021. godine iz oblasti Elektrotehnike i računarstva – Softversko inženjerstvo i informacione tehnologije.
kontakt: tomic.nikola.uns@gmail.com

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.85
DOI: https://doi.org/10.24867/16BE42Ogrizovic

AUTOMATSKA SUMARIZACIJA VESTI O KRIPTOVALUTAMA

AUTOMATIC SUMMARIZATION OF CRYPTOCURRENCY NEWS

Uroš Ogrizović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Kriptovalute su dosegle veliku popularnost 2021. godine. U ovom radu su evaluirani aktuelni transformer modeli na zadatku automatske sumarizacije vesti o kriptovalutama. Ulaz modela je sekvenca tokena koja predstavlja tekst vesti. Izlaz modela je sekvenca tokena koja predstavlja sumarizaciju vesti sa ulaza. Kao labela je korišćen naslov vesti. Predloženi su koraci za poboljšanje performansi modela.*

Ključne reči: *mašinsko učenje, transformer, automatska sumarizacija teksta*

Abstract – *This paper utilizes several popular transformer models to perform automatic text summarization of cryptocurrency news. The model's input is a sequence of tokens representing the content of the news, while the model's output is a sequence of tokens representing a summary of the input. The title of each piece of news was used as the label. The paper discusses the steps for further performance improvements.*

Keywords: *machine learning, transformer, automatic text summarization*

1. UVOD

Kriptovaluta je oblik digitalne imovine koja se koristi kao sredstvo razmene. Početkom 2009. godine, programer ili grupa programera pod pseudonim Satoši Nakamoto je predstavio Bitkoin, prvu decentralizovanu kriptovalutu. Od tada je stvoren veliki broj novih kriptovaluta.

Kriptovalute su doživele naglu ekspanziju 2017. godine, kada je njihova ukupna tržišna kapitalizacija porasla sa 11 na 177 milijardi dolara. Na početku 2017. godine, cena Bitkoina je iznosila 450 dolara, a na kraju iste godine 19500 dolara.

Novi rast popularnosti kriptovalute su doživele 2021. godine, kada su i konačno u većoj meri prihvaćene od renomiranih institucija poput kompanije *Visa*[1].

Sa druge strane, kriptovalute su i dalje veoma volatilne, na šta ukazuje činjenica da je vrednost Bitkoina na početku ove godine iznosila 29 hiljada dolara, nakon čega je porasla i dostigla vrhunac od preko 64 hiljade dolara u aprilu, da bi se zatim vratila na 30 hiljada dolara u julu, nakon čega je ponovo rasla do 50 hiljada u septembru.

Jedan od faktora koji je uticao na promenu vrednosti Bitkoina bili su tvitovi Ilona Maska: u prvom[2], iz marta, naveo je kako njegova kompanija Tesla prihvata Bitkoin kao sredstvo plaćanja (i pritom je Tesla kupila određenu količinu Bitkoina), dok je u drugom[3], iz maja, naveo kako Tesla više neće prihvatati Bitkoin kao sredstvo plaćanja jer ta kriptovaluta nije dobra po životnu sredinu. Mnogi su postupke Maska protumačili kao pokušaj manipulacije tržištem.

Oblast sumarizacije teksta je veoma izazovna za mašine zbog toga što one ne poseduju sposobnost razumevanja jezika na nivou na kom je poseduju ljudi. Sa druge strane, s obzirom na to da mašine mogu da čitaju tekst mnogo brže nego ljudi, razvijanje alata za sumarizaciju vesti o kriptovalutama omogućava brže sticanje saznanja o kriptovalutama.

2. PRETHODNA REŠENJA

Postoje dve tehnike sumarizacije teksta:

1. ekstraktivna - iz teksta se biraju najkvalitetnije rečenice po nekoj funkciji za vrednovanje značaja rečenice. Na primer, moguće je odrediti značaj rečenice kao sumu težina reči koje se u njoj nalaze, pri čemu težina reči može da odgovara frekventnosti te reči u tekstu. Izvučene rečenice se spajaju u celinu koja se prosleđuje kao ulaz modelu
2. apstraktivna - ne vrši kopiranje rečenica, već na vrši apstrahovanje suštine teksta. Ovaj pristup više podseća na ono što bi čovek uradio kada bi se od njega zatražilo da izvrši sumarizaciju teksta

U ovom poglavlju su predstavljena prethodna rešenja koja su se bavila apstraktivnom automatskom sumarizacijom teksta.

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

[1] Vest o prihvatanju kriptovaluti kao sredstava plaćanja od strane *Visa* kompanije, pristupljeno 15.10. 2021. https://www.forbes.com/sites/ninabambysheva/2021/03/29/visa-to-start-settling-transactions-with-bitcoin-partners-in-usdc/?sh=489b3d5f5228

[2] Tvit Ilona Maska o prihvatanju Bitkoina kao sredstva plaćanja, pristupljeno 15.10.2021. https://twitter.com/elonmusk/status/1374617643446063105

[3] Tvit Ilona Maska o prestanku prihvatanja Bitkoina kao sredstva plaćanja, pristupljeno 15.10.2021. https://twitter.com/elonmusk/status/1392602041025843203

U radu [1] su autori predstavili model BART (***B**idirectional **A**uto-**R**egressive **T**ransformer**s**). Cilj ovog rada jeste razvijanje autoenkodera za pretreniranje jezičkog modela (eng. *Language Model*), koji zatim može da se koristi za mnoštvo različitih zadataka, uključujući i sumarizaciju teksta. Glavna ideja se sastoji iz dva koraka: ”kvarenje“ teksta ubacivanjem šuma u isti pomoću proizvoljne funkcije i obučavanja modela da rekonstruiše originalni tekst. Korišćena je arhitektura neuronske mreže zasnovana na transformerima. Autori su dobili najbolje rezultate kada su promešali redosled rečenica u tekstu i iskoristili tačno jedan maskirni token da zamene nijednu, jednu ili više reči u tekstu. Na ovaj način, postignuta je generalizacija *masked language modeling* (MLM) i *next sentence prediction* (NSP) zadataka iz BERT [2] modela, jer model u ovom slučaju ne zna tačnu dužinu rečenice.

U radu [3] predstavljen je *sequence-to-sequence* (Seq2Seq) model ProphetNet zasnovan na transformerima koji uvodi samonadgledani zadatak koji se zove *future n-gram prediction*, kao i *n-stream self-attention* mehanizam. Pored klasičnog Seq2Seq modela koji optimizuje predviđanje za jedan korak unapred, ProphetNet takođe vrši predikciju n koraka unapred koristeći *future n-gram prediction* mehanizam, koji forsira model da ne vrši preprilagođavanje (*overfitting*) nad lokalnim korelacijama u tekstu. Odstupanje od klasičnog transformera prisutno je na strani dekodera, gde se, umesto predviđanja samo sledećeg tokena u svakom vremenskom koraku, vrši predviđanje narednih *n* tokena istovremeno.

Cilj rada [4] jeste uvođenje TLDR generisanja, nove vrste ekstremne sumarizacije naučnih radova, čiji je ulaz naučni rad, a izlaz jedna rečenica koja ga sumarizuje, i ta rečenica se naziva TLDR. Generisanje TLDR-a je alternativa apstraktu, koji se uvek nalazi na početku naučnog rada i predstavlja sumarizaciju tog rada. Za pisanje TLDR-a je potrebno ekspertsko znanje i poznavanje jezika specifičnog za domen (*domain-specific language*, DSL). TLDR-ovi se fokusiraju na ključne delove rada, izbacujući suvišne detalje. Imajući u vidu činjenicu da se broj godišnje objavljenih naučnih radova duplira svakih devet godina [5], evidentna je korist koju TLDR-ovi donose.

U radu [6] uveden je radni okvir koji pretvara sve probleme jezika zasnovane na tekstu u tekst-na-tekst (*text-to-text*) format, to jest, tekst je ulaz, i tekst je izlaz. Autori su na raznim zadacima iz oblasti obrade pri-rodnih jezika (*Natural Language Processing*, NLP) testirali različite arhitekture, skupove podataka, tehnike pretreniranja, pristupe transfer učenju i još drugih aspekata obrade prirodnog jezika. Postignuti su *state-of-the-art* rezultati na mnogim zadacima, poput sumarizacije teksta, odgovaranja na pitanja i klasifikacije teksta. Takođe, ovaj rad je uveo C4 skup podataka (*Colossal Clean Crawled Corpus*). Pristup koji podrazumeva tretiranje svakog problema kao tekst-na-tekst problem omogućava direktno korišćenje jednog modela sa istom funkcijom gubitka I istim vrednostima hiperparametara na više različitih zadataka. Modelu, koji je sličan originalnom transformer modelu iz rada [7], kao prefiks je prosleđivan naziv zadatka koji treba da izvrši. Na primer, za prevođenje rečenice: ”That is good.“ sa engleskog na nemački jezik, ulaz je: ”translate English to German: That is good.“, a izlaz je: ”Das ist gut.“, dok je prefiks za sumarizaciju: ”summarize:“.

Cilja rada [8] jeste vršenje apstraktivne sumarizacije rečenica pomoću pristupa sumarizaciji zasnovanog na pažnji koji autori nazivaju *Attention-Based Summarization* (ABS). Autori su kao model upotrebili kombinaciju arhitekture modela jezika (*language model*, LM) sa kontekstualnim enkoderom ulaza.

Model jezika je prilagođen po uzoru na rad [9]. Enkoder ulaza je modelovan po uzoru na enkoder zasnovan na pažnji (*attention-based encoder*) iz rada [10]. Ključna razlika između modela koršćenog u ovom radu i klasičnih modela jezika je što su enkoder i model jezika trenirani istovremeno.

Više različitih vrsta enkodera je testirano: *bag-of-words* (BOW), konvolutivni enkoder, kao i enkoder zasnovan na pažnji. Za treniranje je korišćen anotirani *Gigaword* skup podataka [11].

2.6 OBJAŠNJENJA METRIKA ZA EVALUACIJU SUMARIZACIJE TEKSTA

Po radu [12], tehnike evaluacije sumarizacija dele se na:

1. ekstrinzične – ove tehnike evaluiraju koliko dobro se sistem koji vrši sumarizaciju ponaša na nekom zadatku za koji je potrebna sposobnost vršenja sumarizacije, poput odgovaranja na pitanja sa više ponuđenih odgovora ili pretrage dokumenata po nekim kriterijumima.
2. intrinzične – ove tehnike vrše poređenje automatskih sumarizacija sa ”zlatnim standardom“, to jest, sa ljudskim sumarizacijama. Medu intrinzične metrike ubrajaju se: *Perplexity*, ROUGE, BLEU, METEOR. Prednost intrinzičnih metrika nad ekstrinzičnim je što programerima daju eksplicitnu povratnu informaciju o kvalitetu sumarizacija. Ekstrinzične metrike tu vrstu informacija pružaju implicitno, kroz rezultate postignute pri obavljanju nekog zadatka. Sa druge strane, nedostatak intrinzičnih metrika je što zahtevaju skup referentnih sumarizacija, to jest, anotiran skup podataka.

ROUGE-N je intrinzična metrika koja evaluira preklapanje N-grama između mašinske i ljudske (referentne) sumarizacije. Tačnije, računa se koliki procenat N-grama iz ljudske sumarizacije se nalazi u mašinskoj sumarizaciji. ROUGE-1 posmatra unigrame, to jest, reči, dok ROUGE-2 posmatra bigrame, to jest, parove reči.
BLEU je intrinzična metrika koja koristi modifikovanu verziju preciznosti da izračuna sličnost između dve rečenice. Razlika između preciznosti i modifikovane preciznosti je što modifikovana preciznost ne razmatra samo da li se neka reč pojavljuje u referentnim rečenicama, već i koliko puta se u njima pojavljuje. U praksi, korišćenje unigrama nije optimalno, pa se koriste N-grami. Empirijski je utvrđeno da korišćenje tetragrama (N=4) daje rezultate najsličnije ljudskom razmišljanju, pa se iz tog razloga BLEU-4 često koristi u praksi. Za

primere eksploatacije problema BLEU metrike pogledati poglavlje 3 rada [13].

3. SPECIFIKACIJA I IMPLEMENTACIJA

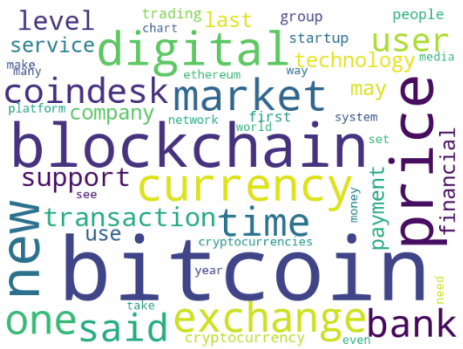
Cilj ovog rešenja je korišćenje T5 i BART modela za vršenje apstraktivne sumarizacije vesti o kriptovalutama. U potpoglavlju 3.1 su izloženi alati koji su korišćeni za implementaciju rešenja. T5 i BART modeli su dotrenirani nad skupom podataka koji je opisan u potpoglavlju 3.2, i njegove performanse su upoređene sa verzijom istog modela koja nije dotrenirana nad tim skupom podataka.

3.1 Korišćeni alati

Korišćeni su modeli iz okvira otvorenog koda *Hugging Face* u programskom jeziku *Python* verzije 3.8: $T5_{BASE}$, $BART_{BASE}$. Računar na kom su razvijani i trenirani modeli ima sledeće specifikacije: procesor *Intel Core i7-8750-H*, grafička kartica *Nvidia GeForce GTX 1050 Ti 3GB RAM*, 32GB radne memorije.

3.2 Skup podataka

Korišćen je skup vesti o kriptovalutama koji je dostupan na sajtu *Kaggle*[4]. Taj skup sadrži 40 hiljada vesti o kriptovalutama u periodu od 2013. do 2018. godine koje su preuzete sa popularnih sajtova, među kojima su CCN, CoinDesk i NewsBTC. Svaka vest sadrži sledećih sedam obeležja: URL, naslov, tekst tela vesti, HTML tela vesti, godina objavljivanja, ime autora, izvor. Autor skupa podataka je naveo da ovaj skup podataka može da se koristi kao merilo kvaliteta algoritama sumarizacije teksta. Sve do 2017. godine, Bitkoin je bio najpopularnija kriptovaluta. Stoga, ne čudi što je među tekstovima iz skupa podataka najučestalija reč ”Bitcoin“ sa preko 12000 pojavljivanja. Ona se pojavljuje u 80% tekstova i u 52% naslova iz skupa podataka. Na slici 1 je dat oblak reči sa 50 najfrekventnijih reči u tekstvoima iz skupa podataka.



Slika 1: *50 najfrekventnijih reči u tekstovima vesti iz skupa podataka*

U radu [4] naslov je korišćen kao pomoćni signal pri obučavanju, što ukazuje na to da se kvalitetan naslov makar u nekoj meri može smatrati adekvatnom sumarizacijom. Još jedan razlog zašto se naslov može smatrati adekvatnom sumarizacijom je taj što se 93% najčešćih reči iz tekstova tela vesti o kriptovalutama nalazi i u naslovima vesti. Ipak, jasno je da su naslovi naučnih radova većeg kvaliteta od naslova vesti o kriptovalutama, kako zbog toga što ih pišu stručnjaci iz oblasti, tako i zbog toga što nemaju potrebu za senzacionalizmom. Stoga, može se zaključiti da naslovi vesti iz skupa podataka ponekad jesu, a ponekad nisu adekvatna sumarizacija. U okviru potpoglavlja 4.1 je diskutovan primer gde naslov nije adekvatna sumarizacija vesti.

U proseku, tekst tela vesti sadrži 535,78 reči, dok je prosečna dužina naslova 8,85 reči.

4. EVALUACIJA REŠENJA I REZULTATI

Dotrenirana su dva modela - jedan za T5 arhitekturu, a drugi za BART arhitekturu. U pitanju su BASE verzije modela, koje su manje od LARGE verzija. Svaki od modela je treniran nad 10000 vesti, i evaluiran nad 2000 vesti pomoću tri metrike: ROUGE-1, ROUGE-2, BLEU. Takođe, verzije T5 i BART modela koje nisu dotrenirane nad vestima iz oblasti kriptovaluta su evaluirane nad istih 2000 vesti, i rezultati su upoređeni.

Ciljna sumarizacija, koja se sastoji iz jedne ili više rečenica, generisana je na osnovu teksta tela vesti. Kao labela je korišćen naslov. Dužina ulazne sekvence je pri dotreniranju ograničena na 50 tokena. Za *batch size* je korišćena vrednost 64. Treniranje je vršeno u tri epohe, na procesoru, zbog toga što nije bilo dostupno dovoljno radne memorije na grafičkoj kartici.

Pri pretprocesiranju teksta uklonjeni su karakteri šuma i veći broj razmaka je zamenjen jednim razmakom. Prebacivanje teksta u mala slova nije poboljšalo rezultate, kao ni lematizacija. U tabeli 1 dat je prikaz rezultata evaluacije modela za ulaznu sekvencu ograničenu na dužinu od 50 tokena.

metrika / model	ROUGE-1	ROUGE-2	BLEU-4
T5*	31,40	**12,99**	**0,099**
BART*	**31,71**	12,89	0,097
T5	23,41	7,16	0,029
BART	20,97	6,84	0,025

Tabela 1: Vrednost metrike po modelu. * označava dotrenirane verzije modela.

Iz rezultata je uočljivo da je dotreniranje modela doprinelo poboljšanju performansi. Takođe, rad [13] navodi da ne postoje statistički značajne razlike u performansama između T5 i BART modela, što je u skladu sa rezultatima iz tabele 1.

Empirijski su ispitane različite vrednosti za dužinu ulazne i dužinu izlazne sekvence. S obzirom na to da su dobijeni isti rezultati za ograničenje dužine ulazne sekvence na 50, 200 i 500 tokena, radi uštede resursa je izabrana najmanja od ispitanih vrednosti za ograničenje dužine sekvence.

Tri metode kojima bi ove performanse mogle da se poboljšaju su:

[4] Skup vesti o kriptovalutama, pristupljeno 15.10.2021. https://www.kaggle.com/kashnitsky/news-about-major-cryptocurrencies-20132018-40k/version/2

1. korišćenje LARGE verzija modela umesto BASE verzija
2. korišćenje kvalitetnijih referentnih sumarizacija, koje se mogu generisati pomoću neke SOTA arhitekture kao što je PEGASUS, koja je uvedena u radu [14].
3. Vršenje rekurizvne sumarizacije, po uzoru na rad [15]

4.1 Analiza grešaka modela

Performansama modela škodi činjenica da je naslov korišćen kao referentna sumarizacija. Na primer, za pojedine vesti model je proizveo smislenu sumarizaciju, ali naslov nije smislen, i zbog toga je ostvarena mala vrednost za metriku evaluacije. Jedan takav primer dat je na slici 2.

text: summarize:the emblem of the peoples' republic of china. editor's note: btc china has been dropped by tenpay, the 2nd largest third party payment processor in china after alipay, and is currently using yeepay (an obscure smaller third party payment processor) for deposits and withdrawals. btc china is also located in a special zone in shanghai and it is my opinion that btc china will continue legal operations in the best interest of their users; however, the waters in the meantime will be tumultuous. the timeline is this: by chinese new years, chinese bitcoin exchanges will have to have found new ways for users to deposit and withdraw. virtual commodity payment processors anyone? the bitcoin chart remains somewhat precarious following the latest round of news and rumours out of china. all markets hate uncertainty, so,

title: translation of sina.com article by cryptocoinsnews editor, caleb chen,

summary: chinese Bitcoin Exchanges Will Have to Find New Ways to Deposit and Refund,

rouge1: 0.0

Slika 2: *Greška T5 modela usled neadekvatnog naslova*

Prisutni su i primeri gde je model napravio grešku u sumarizaciji zbog toga što nije video dovoljan deo teksta vesti.

Uzrok najvećeg broja loše ocenjenih sumarizacija kada je u pitanju BLEU metrika jeste činjenica da su sumarizacije mnogo duže od naslova, te ne daju visoku vrednost metrike, iako su veoma smislene.

5. ZAKLJUČAK

U ovom radu su evaluirani aktuelni transformer modeli na zadatku automatske sumarizacije vesti o kriptovalutama. Korišćen je javno dostupan skup podataka sa sajta *Kaggle*. Ulaz modela je sekvenca tokena koja predstavlja tekst vesti. Izlaz modela je sekvenca tokena koja predstavlja sumarizaciju vesti sa ulaza. Kao labela je korišćen naslov vesti. Za evaluaciju su korišćene ROUGE i BLEU metrike. Rezultati pokazuju da je dotreniranje modela doprinelo poboljšanju njihovih performansi. Izložene su i analizirane uočene greške u performansama. Predloženi su koraci za poboljšanje performansi modela.

6. LITERATURA

[1] BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension

[2] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[3] Qi, Weizhen, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. "Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training." arXiv preprint arXiv:2001.04063 (2020).

[4] Cachola, Isabel, Kyle Lo, Arman Cohan, and Daniel S. Weld. "TLDR: Extreme summarization of scientific documents." arXiv preprint arXiv:2004.15011 (2020).

[5] Van Noorden, Richard. "Global scientific output doubles every nine years." Nature news blog (2014).

[6] Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. "Exploring the limits of transfer learning with a unified text-to-text transformer." arXiv preprint arXiv:1910.10683 (2019).

[7] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In Advances in neural information processing systems, pp. 5998-6008. 2017.

[8] Rush, Alexander M., Sumit Chopra, and Jason Weston. "A neural attention model for abstractive sentence summarization." arXiv preprint arXiv:1509.00685 (2015).

[9] Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Janvin. "A neural probabilistic language model." The journal of machine learning research 3 (2003): 1137-1155.

[10] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

[11] Fan, James, Raphael Hoffmann, Aditya Kalyanpur, Sebastian Riedel, Fabian Suchanek, and Partha Talukdar. "Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)." In Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX). 2012.

[12] Mani, Inderjeet. "Summarization evaluation: An overview." (2001).

[13] Callison-Burch, Chris, Miles Osborne, and Philipp Koehn. "Re-evaluating the role of BLEU in machine translation research." In 11th Conference of the European Chapter of the Association for Computational Linguistics. 2006.

[14] Zhang, Jingqing, Yao Zhao, Mohammad Saleh, and Peter Liu. "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization." In International Conference on Machine Learning, pp. 11328-11339. PMLR, 2020.

[15] Wu, Jeff, Long Ouyang, Daniel M. Ziegler, Nissan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. "Recursively Summarizing Books with Human Feedback." arXiv preprint arXiv:2109.10862 (2021).

Kratka biografija:

Uroš Ogrizović rođen je 1997. godine. Osnovne akademske studije završio je 2020. godine na Fakultetu tehničkih nauka, na kom brani i master rad 2021. godine iz oblasti Elektrotehnike i računarstva – Softversko inženjerstvo i informacione tehnologije.
Kontakt: uros.ogrizovic@uns.ac.rs

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.38:620.9
DOI: https://doi.org/10.24867/16BE43Korpas

REALIZACIJA FOTONAPONSKOG ROTATORA ZA NAVODNJAVANJE

IMPLEMENTATION OF PHOTOVOLTAIC TRACKER FOR IRRIGATION SYSTEM

Miroslav Korpaš, Zoltan Čorba, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu dat je detaljan opis elemenata sistema za navodnjavanje koji za dobijanje električne energije za napajanje pumpe za navodnjavanje koristi fotonaponske panele montirane na noseću konstrukciju koja prati položaj Sunca na nebu.*

Ključne reči: *Fotonaponski panel, rotator, navodnjavanje*

Abstract – *This paper gives a detailed description of the elements of the irrigation system. The irrigation system use photovoltaic panels mounted on the supporting structure that monitors the position of the Sun in the sky to obtain electricity that power the pump.*

Keywords: *Photovoltaic panel, tracker, irrigation*

1. UVOD

U poljoprivredi se sve ćešće koristiti solarna energija za navodnjavanje. Najveća efikasnost se postiže rotatorskim sistemima koji prate položaj Sunca. Za to je potrebno poznavanje solarne geometrije. Naime, od međusobnog položaja Sunca i Zemlje zavisi solarno zračenje koje dospe na površinu Zemlje.

Noseće konstukcije fotonaponskih (FN) panela mogu biti fiksne ili rotatorske izvedbe. Rotatorski sistemi su tehnički složeniji od fiksnih i za njihovo postavljanje je potrebna veća površina. Ipak, prednost rotatorskih sistema u odnosu na fiksne se ogleda u tome što rotatorski sistemi primaju veću količinu Sunčevog zračenja, a samim tim imaju i veću proizvodnju električne energije.

2. SOLARNA GEOMETRIJA

2.1. Sunčevo zračenje

Sunce, najbliža zvezda našoj planeti, predstavlja fuzioni reaktor koji pretvara vodonik u helijum pri čemu se oslobađa ogromna količina energije. Od ukupno $3{,}8\text{x}10^{26}$W snage koju Sunce zrači u svemir u sekundi, Zemlja primi $1{,}75\text{x}10^{17}$W. To znači da Zemlja dobija više energije od Sunca u toku jednog sata nego što ljudska populacija potroši za godinu dana [1].

U zavisnosti od tačke na kojoj se posmatra Sunčevo zračenje razlikuju se:

1. Direktno Sunčevo zračenje: Zračenje Sunca koje dolazi na samu površinu Zemlje;
2. Ekstraterestično zračenje: Zračenje sa Sunca koje dolazi do vrha atmosfere.

Ekstraterestično zračenje može da se: apsorpuje, reflektuje ili transmituje kroz atmosferu.

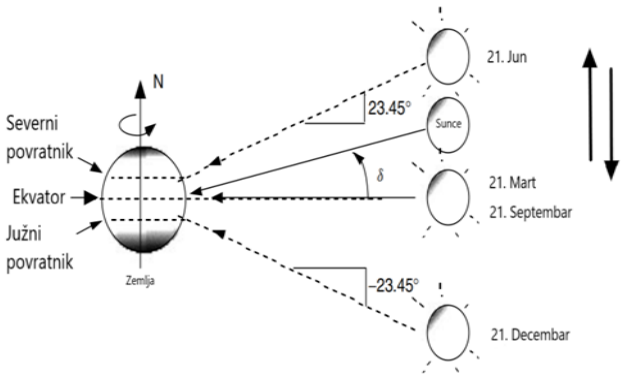
2.2. Kretanje Zemlje

Planeta Zemlja u okviru svog kretanja ima dve komponente:

1. Kretanje Zemlje oko Sunca po eliptičnoj putanji (revolucija) [1];
2. Kretanje Zemlje oko sopstvene ose (rotacija) [1].

Putanja kojom se Zemlja kreće oko Sunca se naziva ekliptika, dok se ravan koja je sadrži naziva ravan ekliptike. Ekvator je zamišljena kružnica koja opisuje Zemlju, a koja se nalazi na podjednakoj udaljenosti od polova. Ravan koja sadrži ekvator naziva se ekvatorijalna ravan.

Osa rotacije Zemlje je normalna na ekvatorijalnu ravan. Ugao koji zaklapa osa rotacije Zemlje i ravan ekliptike je ugao deklinacije (aksijalni ugao) i iznosi 23,45°. Na slici 1 je prikazana promena ugla deklinacije u zavisnosti od dana u godini.



Slika 1. *Promene ugla deklinacije* δ *[2]*

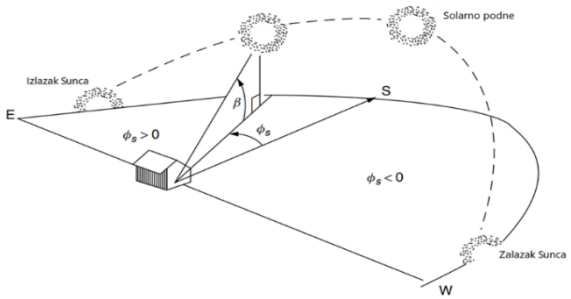
Gledano sa površine Zemlje ugao deklinacije δ se menja u toku godine. Taj ugao se nalazi između ekvatorijalne ravni i prave koja sadrži centar Sunca i centar Zemlje kao što je prikazano na slici 1. Ugao deklinacije računa se na sledeći način:

$$\delta = 23.45 \cdot sin(360 \cdot (n - 81)/365) \qquad (1)$$

gde je n redni dan u godini.

Za određivanje tačnog položaja Sunca na nebu u bilo koje vreme potrebno je znati njegov visinski ugao β i azimutni ugao φ_s (prikazani na slici 2) [2].

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Zoltan Čorba, vanr. prof.



Slika 2. *Prikaz visinskog i azimutnog ugla [2]*

Uglovi β i φ_s zavise od: geografske širine L, dana u godini, doba dana i satnog ugla Sunca H. Računaju se na osnovu sledećih izraza:

$$sin\,\beta = cos\,L \cdot cos\,\delta \cdot cos\,H + sin\,L \cdot sin\,\delta \qquad (2)$$

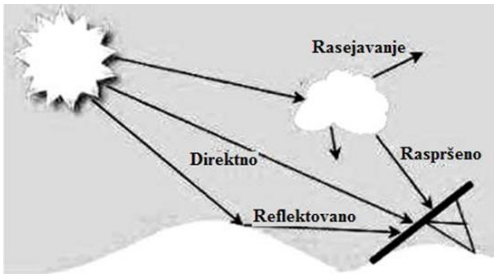
$$sin\,\varphi_s = cos\,\delta \cdot sin\,H / cos\,\beta \qquad (3)$$

Satni ugao Sunca predstavlja broj stepeni koji Zemlja treba da rotira pre nego što se Sunčev meridijan poklopi sa lokalnim Zemljinim meridijanom (to se događa kada je Sunce u zenitu).

2.3. Komponente Sunčevog zračenja koje dolaze na površinu Zemlje

Ukupno zračenje koje dospe do Zemljine površine može se razložiti na sledeće tri komponente koje su prikazane na slici 3:

1. Direktno Sunčevo zračenje - zračenje koje pravolinijski prolazi kroz Zemljinu atmosferu;
2. Raspršeno Sunčevo zračenje - zračenje koje su atomi ili molekuli (koji se nalaze u Zemljinoj atmosferi) rasejali u svim pravcima;
3. Reflektovano Sunčevo zračenje - zračenje koje se odbilo od Zemljine površine ili nekog objekta na njoj.



Slika 3. *Prikaz komponenti Sunčevog zračenja [3]*

Zbog apsorpcije i refleksije Sunčevog zračenja kroz atmosferu do Zemljine površine stigne samo oko 50% ekstraterestičnog zračenja.

Direktna komponenta Sunčevog zračenja koje dolazi do fotonaponskog panela I_{BC} modeluje se na sledeći način:

$$I_{BC} = I_B \cdot cos\,\theta \qquad (4)$$

gde je I_B direktna komponenta Sunčevog zračenja koje dolazi do Zemljine površine, a θ upadni ugao Sunčevog zračenja u odnosu na panel.

Raspršena komponenta Sunčevog zračenja koje dolazi do fotonaponskog panela I_{DC} modeluje se na sledeći način:

$$I_{DC} = I_{DH} \cdot ((1 + cos\,\Sigma)/2) \qquad (5)$$

gde je I_{DH} raspršena komponenta Sunčevog zračenja koje dolazi do Zemljine površine, a Σ ugao nagiba fotonaponskog panela.

Reflektovana komponenta Sunčevog zračenja koje dolazi do fotonaponskog panela I_{RC} modeluje se na sledeći način:

$$I_{RC} = \rho \cdot (I_{BH} + I_{DH}) \cdot ((1 - cos\,\Sigma)/2) \qquad (6)$$

gde je I_{BH} direktna komponenta Sunčevog zračenja na horizontalnoj površini Zemljine, a ρ refleksija tla ispred fotonaponskog panela.

Za određivanje ukupnog Sunčevog zračenja koje dolazi do fotonaponskog panela potrebno je sabrati sve tri komponente Sunčevog zračenja.

$$I_C = I_{BC} + I_{DC} + I_{RC} \qquad (7)$$

3. VRSTE NOSEĆE KONSTRUKCIJE FOTONAPONSKIH PANELA

Fotonaponski paneli se postavljaju na različite sisteme za montažu odnosno na različite noseće konstrukcije. U zavisnosti od mesta montaže razlikuju se:

1. Noseće konstrukcije koje se montiraju na površinu zemlje;
2. Noseće konstrukcije koje su instalirane na građevinskim objektima (zgradama, kućama, halama...);
3. Noseće konstrukcije na vodenim površinama.

3.1. Noseće konstrukcije koje se montiraju na površinu zemlje

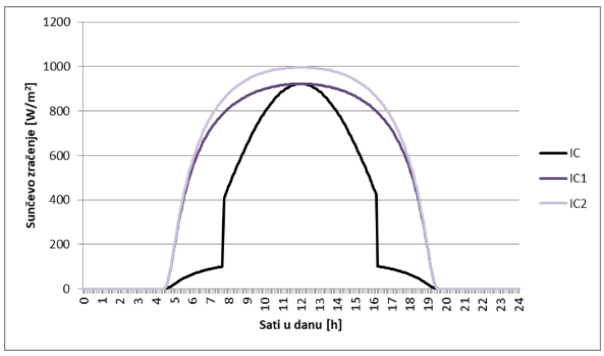
Konstrukcije koje se montiraju na površinu zemlje moraju imati neku vrstu temelja. Razlikuju se fiksne i rotatorske konstrukcije.

Rotatorske konstrukcije su dosta složenije izvedbe u odnosu na fiksne. Što se tiče same noseće konstrukcije rotatora, stubovi se izrađuju od gvožđa koje je zaštićeno od korozije kao i kod fiksnih sistema, međutim elementi na koje se pričvršćuju fotonaponski paneli se izrađuju od aluminijuma. Broj elektromotora koji su potrebni, zavisi od broja osa oko kojih se pokretni deo konstrukcije rotira.

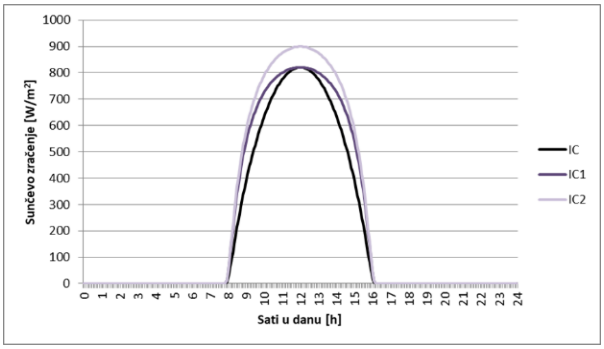
3.2. Prednosti i mane rotatorskih u odnosu na fiksne sisteme

Proizvedena količina električne energije kod fotonaponskih elektrana direktno je proporcionalna ukupnoj količini dolaznog Sunčevog zračenja. Fotonaponske elektrane sa rotatorskim sistemima koje imaju mogućnost praćenja položaja Sunca na nebu po azimutnom ili visinskom uglu ili oba, proizvode veću količinu energije od fiksnih fotonaponskih elektrana.

Na slikama 4 i 5 prikazane su ukupne količine Sunčevog zračenja koje dolazi do površine fotonaponskih panela za fiksne (IC), jednoosne (IC1) i dvoosne (IC2) sisteme za dan letnje dugodnevice i zimske kratkodnevice respektivno.



Slika 4. *Prikaz ukupnog Sunčevog zračenja na dan letnje dugodnevice*



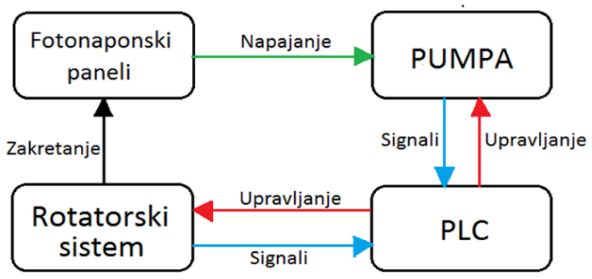
Slika 5. *Prikaz ukupnog Sunčevog zračenja na dan zimske kratkodnevice*

Sa slika 4 i 5 vidi se da je najveća količina zračenja koja dolazi do površine fotonaponskih panela kada je Sunce u zenutu. Kada Sunčevo zračenje ima veći intenzitet, veća je razlika u količini upadne energije na FN panele montiranih na različite vrste noseće konstrukcije.

Potrebna površina za izgradnju fiksne fotonaponske elektrane značajno je manja od površine koju zahteva rotatorska fotonaponska elektrana iste snage. U proseku fotonaponske elektrane snage 1MW sa fiksnim sistemima zauzimaju površinu zemljišta oko 2ha, elektrane sa jednom osom rotacije koje prate položaj Sunca duž ose sever-jug oko 2,5ha, a jednoosne sa vertikalnom osom rotacije i dvoosne rotatorske fotonaponske elektrane od 3ha do 5ha [4].

4. PRIMER PRIMENE ROTATORSKOG SISTEMA ZA NAVODNJAVANJE

Za navodnjavanje površina koje su udaljene od naseljenih mesta, odnosno koje nemaju pristup distributivnoj električnoj mreži, za pokretanje zalivnih pumpi se najčešće koriste dizel agregati ili fotonaponske elektrane malih snaga. Na slici 6 je dat blok dijagram sistema za navodnjavanje FN panelima.



Slika 6. *Blok dijagram FN rotatorskog sistema*

Kao što slika 6 prikazuje sistem za navodnjavanje se sastoji od:

1. Noseće konstrukcije za FN panele,
2. Uplavljačkog sistema za zakretanje FN panela,
3. FN panela,
4. Pumpe i
5. Razvodnog sistema cevi za navodnjavanje.

Na slici 7 je prikazana noseća konstrukcija za 10 fotonaponskih panela.

Slika 7. *Prikaz noseće konstrukcije za 10 FN panela*

Sistem se može napajati iz agregata ili fotonaponskih panela. Na slici 8 je prikazan razvodni orman rotatorskog sistema. Pored sklopne opreme u razvodni orman su ugrađeni PLC i specijalni komunikacioni uređaj za upravljanje pumpom. Sistem može da radi u ručnom ili automatskom režimu.

Slika 8. *Upravljački orman sistema za navodnjavanje*

Za upravljanje elektromotorima, u zakretnom sistemu, namenjeno je 6 digitalnih ulaza PLC-a i 3 digitalna izlaza. PLC-u su potrebne i informacije o tačnoj poziciji sistema i tačnom vremenu za pozicioniranje panela.

Na ulaze PLC-a se povezuju hall senzori sa elektromotora u zakretnom sistemu i krajnji prekidači.

Na izlaze PLC-a se povezuju releji sledeće namene:

1. Relej preko kojeg se kontoliše smer obrtanja elektromotora;
2. Relej preko kojeg se dovodi napon na elektromotor koji zakreće sistem u pravcu istok/zapad;
3. Relej preko kojeg se dovodi napon na elektromotor koji zakreće sistem u pravcu gore/dole.

Fotonaponski paneli su povezani na red, čineći fotonaponski niz. Struja koju daju paneli u tački sa maksimalnom snagom iznosi 8,76A, a napon u tački sa maksimalnom snagom je 6x31,6=189,6V. Ovakav opseg struja i napona može da zadovolji potrebe za napajanjem elektromotora pumpe.

Izbor pumpe vrši se na osnovu izdašnosti bunara i zahtevanog protoka u zalivnom sistemu.

S obzirom da se sistem može napajati sa fotonaponskih panela koji na svom izlazu daju jednosmeran napon ili agregata koji na svom izlazu daje naizmeničan napon, izabrana je Grundoss pumpa iz serije SQFelx koja podržava oba tipa napona. Svaka pumpa iz serije SQFlex se sastoji od tri dela: pretvarač, motor i radna kola.

Pretvarač ima ulogu prilagođenja napona fotonaponskog niza naponu koji je potreban za normalan rad motora u pumpi. Ulazni napon u pretvarač može biti u rasponu od 30VDC do 300VDC ili 1 x 90-240VAC.

Iako sve pumpe iz serije SQFlex koriste isti elektromotor, imaju različite izlazne karakteristike zbog različitog broja radnih kola. Obično veći broj radnih kola obezbeđuje i veći izlazni napor, dok protok zavisi do njihovih dimenzija.

Pumpe imaju u sebi ugrađene sledeće zaštite:

1. Zaštita od rada na suvo;
2. Zaštita od prenapona i podnapona;
3. Zaštita od preopterećenja.

Najčešća primena fotonaponskih panela kao izvora električne energije u navodnjavanju je za zalivanje voćnjaka pomoću sistema kap po kap. Sistem kap po kap se sastoji od jedne ili više magistralne cevi na koje su povezane laterale.

Svaka magistralna cev napaja se vodom preko elektromagnetnog ventila. Laterala je cev određene dužine na kojoj su postavljeni kapljači. Da bi voda isticala kroz svaki kapljač u laterali protokom od oko 2l/h pritisak na početku laterale mora biti oko 1bar.

Na cev koja izlazi iz bunara povezuje se presostat, pomoću njega se ograničava pritisak vode u sistemu. Posle presostata se postavlja filter čija je uloga da sakupi nečistoće koje je pumpa izbacila iz bunara, da ne bi došlo do začepljenja kapljača. Pomoću vodomera se meri količina vode koja ulazi u zalivni sistem, a pomoću elektromagnetnih ventila se određuje koje polje će se zalivati.

Ako sistem za navodnjavanje radi u automatskom režimu, PLC će upravljati sa elektromagnetnim ventilima. Na osnovu informacija sa vodomera i potrebne količine vode u jednom ciklusu PLC određuje koji ventil će biti otvoren i koliki vremenski period. U ručnom režimu rada otvorenost ventila se reguliše pomoću prekidača.

Na slici 9 prikazani su fotonaponski paneli montirani na zakretnu noseću konstrukciju.

Slika 9. *Prikaz FN panela*

5. ZAKLJUČAK

Imajući u vidu da su klimatske promene sve izraženije i da sve više utiču na različite aspekte naših života, potreba za obnovljivim izvorima energije sve više dobija na značaju. Primena fotonaponskih panela za dobijanje električne energije poslednjih godina postaje sve isplativija, s obzirom da cena fotonaponskih panela opada dok sa druge strane cena električne energije dobijena na konvencionalan način raste. Trenutno najisplativiji vid primene fotonaponskih panela u poljoprivredi je za navodnjavanje poljoprivrednih zemljišta koja su udaljena od naseljenih mesta i samim tim nemaju mogućnost dobijanja električne energije iz distributivne mreže. Na taj način se smanjuje nepredvidiv uticaj vremenskih prilika na prinose jer se zemljištu obezbeđuje potrebna količina vode u periodu suše.

6. LITERATURA

[1] https://www.solarnipaneli.org (pristupljeno u junu 2021.)

[2] Masters M. G., Renewable and Effificient Electric Power Systems, Hoboken, Stanford University, (2004)

[3] https:// vtsnis.edu.rs / wp-content /plugins / vts-predmeti / uploads/3%20sunce.pdf (pristupljeno u julu 2021.)

[4] Čorba Zoltan, Fotonaponsko pretvaranje solarne energije i fotonaponske elektrane, Novi Sad, FTN izdavaštvo, (2017)

Kratka biografija:

Miroslav Korpaš rođen je u Novom Sadu 1992. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Energetska elektronika i električne mašine odbranio je 2021.god.

Zoltan Čorba rođen je u Novom Sadu 1962. Doktorirao je na Fakultetu tehničkih nauka 2016. godine. Oblast interesovanja su obnovljivi izvori energije, posebno fotonaponsko pretvaranje energije.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/16BE44Kolosnjaji

GRAFIČKO OKRUŽENJE ZA POMOĆ PRI OTKRIVANJU IDIOMA U PROGRAMSKOM KODU

A GRAPHICAL ENVIRONMENT FOR ASSISTANCE WITH MINING CODE IDIOMS FROM SOURCE CODE

Marijana Kološnjaji, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Generisanje koda u cilju automatizacije razvoja softvera predstavlja najčešći slučaj primene MDSE pristupa. Jedan od alata za generisanje koda je i RoseLib biblioteka, koja nudi API za generisanje C# koda. U ovom radu je predstavljeno rešenje implementirano sa ciljem proširenja RoseLib biblioteke. Implementirano je grafičko okruženje koje omogućava otkrivanje idioma na osnovu skupa idiomatskih softverskih projekata i njihovu integraciju u RoseLib biblioteku. Integracijom pronađenih idioma, API koji nudi RoseLib biblioteka se proširuje na način da omogućava generisanje pronađenih idioma. Time se omogućava lakše generisanje koda koji se ponavlja kroz softverske projekte, čime se ubrzava razvoj softverskog proizvoda.*

Ključne reči: *MDSE, generisanje koda, idiomi*

Abstract – *Code generation with the goal of software development automation is the most common case of applying the MDSE approach. One of the tools for code generation is the RoseLib library, which offers an API for generating C# code. This paper presents a solution implemented with the goal to extend the RoseLib library. A graphical environment which enables mining code idioms based on a set of idiomatic software projects and their integration in the RoseLib library was implemented. By integrating found idioms, the API offered by the RoseLib library is extended to enable generating found idioms. This makes generation of code that recurs over software projects easier, which speeds up the development of a software product.*

Keywords: *MDSE, code generation, code idioms*

1. UVOD

Softversko inženjerstvo upravljano modelima (*Model Driven Software Engineering* – MDSE) predstavlja metodologiju za primenu prednosti modelovanja na aktivnosti softverskog inženjerstva [1]. Automatizacija razvoja softvera generisanjem koda na osnovu modela predstavlja jednu od najčešćih primena MDSE principa. Upotreba generatora koda u procesu razvoja softverskog proizvoda može znatno uticati na smanjenje vremena razvoja i poboljšanje kvaliteta proizvoda. Međutim, generisanje kompletnog softverskog rešenja često nije izvodljivo zbog kompleksnosti njegove implementacije. Tada postoji potreba za dodavanjem određenih delova koda ručno, od strane programera. Iz tog razloga je potrebna upotreba tehnika za integraciju generisanog i ručno pisanog koda.

Postoje razne tehnike za integraciju generisanog i ručno pisanog koda. Međutim, većina njih zahteva određenu strukturu rešenja. U situacijama kada se MDSE tehnike koriste za unapređenje postojećih softverskih proizvoda, promena strukture rešenja često nije moguća. U tom slučaju je potrebno koristiti alate koji omogućavaju integraciju bez potrebe za uvođenjem promena u okviru strukture postojećeg rešenja.

Jedan od takvih alata je i RoseLib biblioteka za generisanje C# koda. RoseLib biblioteka je osmišljena kao poboljšanje Roslyn-a, alata koji nudi mogućnost generisanja koda manipulacijom sintaksnim stablom izvornog koda. RoseLib nudi API (*Application Programming Interface*) koji omogućava generisanje koda upotrebom osnovnih koncepata C# jezika i time nudi intuitivniji način generisanja koda u odnosu na Roslyn.

U ovom radu je predstavljeno rešenje implementirano u cilju proširenja RoseLib biblioteke. Implementirano je grafičko okruženje koje omogućava integraciju idioma pronađenih na osnovu skupa idiomatskih softverskih projekata u RoseLib biblioteku. Pod integracijom idioma u RoseLib se podrazumeva proširenje API-ja biblioteke kako bi se omogućilo generisanje idioma.

2. TEORIJSKE OSNOVE

U ovom poglavlju su opisane tehnologije i alati na kojima se zasniva rad i čije razumevanje je potrebno za razumevanje implementacije rešenja.

2.2. Roslyn

.NET Compiler Platform SDK [2], poznat i pod nazivom Roslyn, predstavlja set API-ja razvijen od strane Microsoft-a. Omogućava pristup modelu koda koji kompajler gradi u toku validacije sintakse i semantike koda. Na taj način pruža mogućnost direktnog pristupa mnoštvu informacija o izvornom kodu koje su dostupne samo kompajleru. Može se primeniti u razvoju alata za analizu, generisanje i transformaciju koda.

Za kontekst rada su iz seta API-ja Roslyn-a najznačniji API kompajlera i API radnog prostora. API kompajlera

---

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bila prof. dr Gordana Milosavljević.

preslikava redosled faza kompajlera. Za svaku fazu kompajlera postoji model objekata koji omogućava pristup informacijama dostupnim u toj fazi. Osnovna struktura podataka koja se koristi iz API-ja kompajlera je sintaksno stablo. Koristi se za analizu strukture izvornog koda, kao i za kreiranje novih i izmenu postojećih delova koda. API radnog prostora spaja informacije o svim projektima u rešenju u jedan model objekata i omogućava im direktan pristup. To doprinosi omogućavanju analize i generisanja koda u okviru čitavih rešenja.

2.3. RoseLib

RoseLib [3] predstavlja biblioteku za generisanje C# koda koja apstrahuje detalje implementacije Roslyn-a. Implementirana je u cilju rešavanja nedostataka Roslyn-a, koji ga čine kompleksnim za upotrebu. Nudi API kojim omogućava generisanje koda upotrebom osnovnih koncepata C# jezika, bez potrebe za fokusiranjem na izgradnju sintaksnih stabala. To čini generisanje koda pomoću RoseLib-a intuitivnijim u poređenju sa generisanjem koda upotrebom Roslyn-a.

RoseLib API se sastoji od dva tipa klasa – selektora i kompozera. Selektori predstavljaju klase koje omogućavaju pretragu sintaksnih stabla. Kompozeri su klase koje omogućavaju izmenu sintaksnih stabla. Organizovani su u vidu hijerarhije koja prati strukturu C# dokumenta.

2.3 Otkrivanje idioma

Idiomi predstavljaju sintaksne fragmente koda koji se ponavljaju kroz softverske projekte i imaju jednoznačnu semantičku ulogu [4]. Mogu sadržati parametre, koji se zovu još i metavarijable. Metavarijable obuhvataju imena identifikatora i sintaksne strukture poput izraza ili blokova koda. Kao jednostavan primer idioma za iteriranje kroz niz u C#-u, može se navesti for petlja. Iako postoji više različitih načina na koji se može iterirati kroz niz u C#-u, for petlja se najčešće koristi od strane programera i zbog toga se smatra idiomom.

RoseLibML [5] predstavlja rešenje za automatsko otkrivanje idioma na osnovu datoteka iz postojećeg skupa softverskih projekata u C# programskom jeziku. Njegova implementacija je zasnovana na Haggis [4] sistemu za automatsko otkrivanje idioma. Korišćene metode su primarno statističke prirode, i kao takve nezavisne od konkretnog jezika. Konkretno, koristi se Markov Chain Monte Carlo (MCMC) metoda za aproksimaciju na osnovu probabilističke kontekstno slobodne gramatike (pCFG) bazirane na dostupnom kodu postojećih softverskih projekata, čime se pronalaze idiomi.

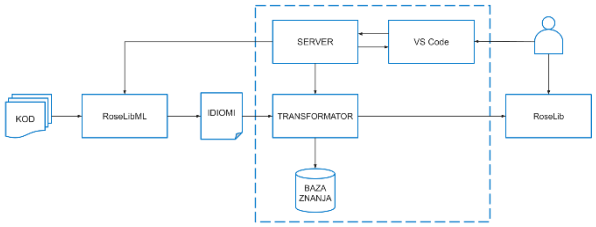
2.4 Language Server Protocol

Language Server Protocol (LSP) [6] definiše standard za komunikaciju između klijenta, koji može biti IDE ili neki od alata za razvoj koda, i jezičkog servera. Kreiran je sa ciljem da olakša obezbeđivanje podrške za različite programske jezike u okviru različitih alata za uređivanje koda.

LSP specifikacija definiše komunikaciju između klijenta i servera preko JSON–RPC protokola. Omogućeno je slanje zahteva, odgovora i obaveštenja. LSP takođe definiše i mogućnost otkazivanja zahteva, kao i mogućnost izveštavanja o napretku i parcijalnim rezultatima u toku obrade zahteva. Definisane su i komande koje omogućavaju izvršavanje proizvoljnih operacija registrovanih od strane servera.

3. IMPLEMENTACIJA

Fokus ovog rada je na implementiranom okruženju za pomoć pri otkrivanju idioma na osnovu skupa softverskih projekata i njihovu integraciju u RoseLib biblioteku. Arhitektura implementiranog rešenja i njegova komunikacija sa ostalim delovima sistema u procesu otkrivanja i integracije idioma su prikazani u okviru dijagrama na slici 3.1. Isprekidanim linijama su obuhvaćeni delovi sistema koji su implementirani u okviru rada.



Slika 3.1 Arhitektura implementiranog rešenja

Rešenje je implementirano u vidu ekstenzije za Visual Studio Code (VS Code) uređivač koda koja preko LSP-a komunicira sa serverom, implementiranim u vidu jezičkog servera. Implementacija servera obuhvata dva glavna zadatka:

1. Otkrivanje idioma na osnovu datoteka iz skupa softverskih projekata
2. Integracija idioma u RoseLib biblioteku

Za otkrivanje idioma, server koristi rešenje implementirano u RoseLibML projektu. Integracija idioma u RoseLib se vrši pomoću posebne komponente koja je nazvana transformator. Za konfiguraciju transformatora se koristi baza znanja.

3.1 Server

Server je implementiran u vidu jezičkog servera, koristeći implementaciju LSP-a napravljenu od strane OmniSharp-a [7]. U najvećoj meri se zasniva na izvršavanju komandi koje su registrovane u okviru servera. Da bi se pokrenulo izvršavanje neke komande, potrebno je sa klijenta poslati *workspace/executeCommand* zahtev iz LSP specifikacije. Kao parametri zahteva se šalju naziv komande i argumenti. Za svaku komandu podržanu od strane servera postoji registrovan rukovalac (eng. *handler*). U toku obrade zahteva, u okviru svakog rukovaoca se prvo vrši validacija argumenata poslatih u zahtevu. Nakon uspešne validacije, nastavlja se sa obradom zahteva i šalje odgovor klijentu.

3.2 Transformator

Transformator je komponenta sistema čiji je primarni zadatak integracija pronađenih idioma u RoseLib biblioteku. Pod integracijom idioma u RoseLib se podrazumeva proširenje biblioteke kako bi omogućila generisanje idioma. To znači da je potrebno proširiti kompozere, komponente RoseLib-a koje omogućavaju generisanje novog koda.

Za svaki idiom pronađen upotrebom RoseLibML projekta, proširuje se odgovarajući kompozer. Informacija o tome koji kompozer se može proširiti nalazi se u bazi znanja. Za proširenje je potrebno u parcijalnu klasu koja predstavlja kompozer generisati novu metodu čija namena je generisanje odabranog idioma. Za generisanje metode koriste se mogućnosti API-ja kompajlera iz Roslyn-a. S obzirom na to da se telo metode razlikuje u zavisnosti od kompozera, za svaki kompozer postoji T4 šablon na osnovu koga se gradi telo metode.

S obzirom na to da je omogućena integracija više idioma u okviru jednog poziva transformera, potrebno ih je najpre grupisati po kompozerima koje proširuju u cilju optimizacije procesa. Ukoliko je datoteka sa parcijalnom klasom koja sadrži odabrane idiome već kreirana, nove metode se integrišu u klasu u toj datoteci. U suprotnom se datoteka sa parcijalnom klasom kreira. Za uključivanje novih datoteka u RoseLib ili integraciju novih metoda u postojeće datoteke, koriste se mogućnosti API-ja radnog prostora iz Roslyn-a.

3.4 Baza znanja

Baza znanja predstavlja komponentu sistema čija uloga je u konfiguraciji transformatora. U implementiranom rešenju, baza znanja je predstavljena u vidu JSON datoteke. U njoj se nalaze podaci o tome u koji kompozer je moguće integrisati koji idiom na osnovu tipa korenskog čvora. Takođe, za svaki kompozer iz RoseLib-a se u okviru baze znanja nalaze pomoćne informacije koje se koriste u toku generisanja koda za njegovo proširivanje. U bazi znanja se takođe nalazi i informacija o putanji na kojoj se nalazi RoseLib biblioteka.

3.5 Klijent

Klijentski deo rešenja je implementiran u vidu ekstenzije za Visual Studio Code uređivač koda. Ekstenzija proširuje korisnički interfejs VS Code-a upotrebom *Webview API*-ja [8]. *Webview API* omogućava kreiranje panela – komponenti koje u okviru postojećeg korisničkog interfejsa mogu prikazati proizvoljne HTML stranice. Za potrebe ekstenzije je kreirano tri panela, od kojih svaki prikazuje jednu fazu u procesu otkrivanja i integracije idioma. Za svaki panel je kreirana HTML stranica u kojoj je definisan izgled, kao i JavaScript datoteka za komunikaciju panela sa ekstenzijom. U svaku stranicu je uključena i CSS datoteka u kojoj je definisan stil izgleda panela. Paneli se prikazuju u obliku kartica u delu za uređivanje koda VS Code alata.

Jedan od zadataka ekstenzije je i komunikacija sa serverom, za koju se koristi *Language Client* modul [9]. Zadužena je za inicijalizaciju i pokretanje servera, koji se dešavaju prilikom aktivacije ekstenzije. Za svaku instancu VS Code-a se pokreće zasebna instanca servera.

Komunikacija ekstenzije sa serverom u toku rada se u najvećoj meri oslanja na slanje zahteva za izvršavanje komandi i obrade odgovora od strane servera. Ekstenzija je takođe zadužena i za zaustavljanje servera, koje se dešava prilikom deaktivacije ekstenzije.

4. PRIKAZ REŠENJA

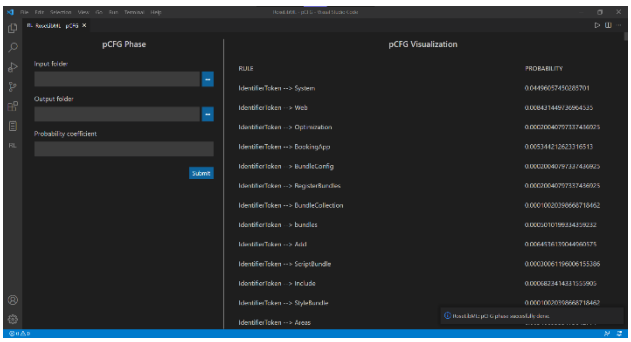
Grafičko okruženje za otkrivanje i integraciju idioma u RoseLib biblioteku, koje je tema rada, implementirano je u vidu ekstenzije za Visual Studio Code uređivač koda. Obuhvata tri glavne funkcionalnosti, koje su ujedno i tri faze u procesu otkrivanja idioma i njihove integracije u RoseLib:

1. Kreiranje pCFG na osnovu skupa datoteka
2. Pokretanje MCMC metode u cilju otkrivanja idioma
3. Podešavanje idioma i njihova integracija u RoseLib biblioteku

Za svaku fazu postoji posebna kartica u okviru VS Code-a, koja se otvara odabirom odgovarajuće opcije iz bočne trake. Predviđeno je da se izvršavaju u redosledu u kome su navedene, s obzirom na to da je izlaz iz jedne faze neophodan za pokretanje sledeće faze.

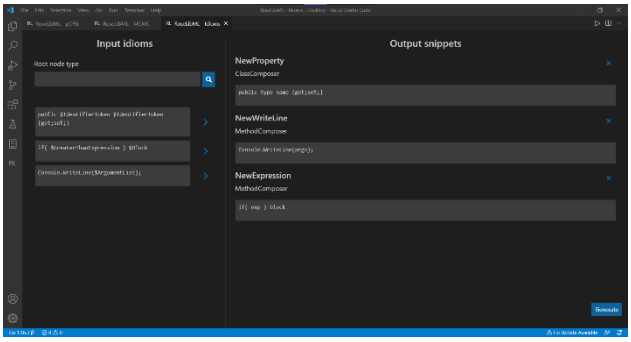
4.1. Upotreba ekstenzije

U okviru kartica za pCFG fazu i MCMC fazu, popunjavaju se forme za unos podataka potrebnih za izvršavanje odgovarajuće komande na jezičkom serveru. U svakoj od njih takođe postoji i mesto za prikaz vizualizacije rezultata. Kao primer se na slici 4.1 može videti prikaz kartice za pCFG fazu.



Slika 4.1 Prikaz izgleda kartice za pCFG

U okviru kartice za podešavanje idioma i njihovu integraciju u RoseLib, prikazuje se lista idioma. Odabirom nekog idioma iz liste, prelazi se na prikaz forme za njegovo podešavanje. U okviru forme se bira jedan od kompozera u koji se idiom može integrisati, unosi se naziv idioma, kao i nazivi za sve metavarijable u okviru idioma. Podešeni idiomi se prikazuju u posebnom delu kartice rezervisanom za njihov prikaz. Prikaz ekstenzije za VS Code sa otvorenom karticom u okviru koje je podešeno tri idioma se može videti na slici 4.2.



Slika 4.2 Prikaz kartice sa podešenim idiomima

Nakon završenog podešavanja, pokreće se opcija generisanja koda koja, na osnovu odabranih idioma i informacija popunjenih u procesu podešavanja, proširuje kompozere iz RoseLib biblioteke.

4.2 Proširenje i upotreba RoseLib biblioteke

Nakon uspešno završene faze integracije, API RoseLib biblioteke je proširen metodama koje omogućavaju generisanje koda sa odabranim idiomima. Za svaki idiom je, u datoteku koja odgovara odabranom kompozeru, dodata metoda čiji zadatak je generisanje tog idioma. Na slici 4.3 se može videti sadržaj generisane datoteke sa parcijalnom klasom za proširivanje *ClassComposer*-a.

```
1  // ------------------------------------------------------------------------------
2  //     This file was generated on 10/13/2021 01:14:28.
3  //
4  //     Changes to this file may cause incorrect behavior
5  //     and will be lost if the code is regenerated.
6  // ------------------------------------------------------------------------------
7  using Microsoft.CodeAnalysis.CSharp;
8  using Microsoft.CodeAnalysis.CSharp.Syntax;
9  using System;
10 using System.Linq;
11
12 namespace RoseLibApp.RoseLib.Composers
13 {
14     public partial class ClassComposer
15     {
16         public ClassComposer AddNewProperty(string type, string name)
17         {
18             if (!IsAtRoot())
19             {
20                 throw new Exception("A class must be selected (which is also a root to
                       the composer)");
21             }
22
23             string fragment = $"public {type} {name} {{get;set;}}";
24             var compilationUnit = CSharpSyntaxTree.ParseText(fragment).GetRoot();
25             var members = (compilationUnit as CompilationUnitSyntax).Members.ToArray();
26             var newNode = (CurrentNode as ClassDeclarationSyntax).AddMembers(members);
27             Replace(CurrentNode, newNode, null);
28             return this;
29         }
30     }
31 }
```

Slika 4.3 Generisana datoteka za *ClassComposer*

Generisana metoda koja se integriše u *ClassComposer* sadrži naziv idioma (*NewProperty*) u svom nazivu i metavarijable idioma (*type* i *name)* kao parametre. U okviru metode se metavarijable iz idioma zamenjuju vrednostima prosleđenim putem argumenata, što se može videti u liniji 23. Zatim se na osnovu idioma pravi sintaksno podstablo koje se dodaje na odabrano mesto u okviru sintaksnog stabla izvornog koda koji se menja.

Na slici 4.4 može se videti primer upotrebe RoseLib biblioteke, nakon integracije idioma, za kreiranje nove C# datoteke. U liniji 17 se vidi upotreba nove generisane metode *AddNewProperty*, čija je definicija prethodno prikazana u kodu na slici 4.3. U ovom primeru se, u liniji 29, takođe koristi i metoda *AddNewWriteLine* koja služi za generisanje još jednog od idioma prethodno prikazanih na slici 4.2.

```
1  CompilationUnitComposer cuComposer = new CompilationUnitComposer();
2  cuComposer.AddUsing("System").AddNamespace("RoseLibTest");
3
4  cuComposer.SelectNamespace();
5  var nsComposer = cuComposer.ToNamespaceComposer();
6
7  nsComposer.AddClass(new ClassOptions()
8      {
9          ClassName = "TestClass",
10         AccessModifier = RoseLib.Enums.AccessModifierTypes.PUBLIC
11     }
12 );
13
14 nsComposer.SelectClassDeclaration("TestClass");
15 var clComposer = nsComposer.ToClassComposer();
16
17 clComposer.AddNewProperty("string", "TestProperty");
18 clComposer.AddMethod(new MethodOptions()
19     {
20         MethodName = "TestMethod",
21         ReturnType = "void",
22         Parameters = new List<RLParameter>()
23     }
24 );
25
26 clComposer.SelectMethodDeclaration("TestMethod");
27 var mComposer = clComposer.ToMethodComposer();
28
29 mComposer.AddNewWriteLine("TestProperty");
```

Slika 4.4 Upotreba RoseLib biblioteke

5. ZAKLJUČAK

U radu je opisano rešenje implementirano u cilju proširenja RoseLib biblioteke. Implementirano rešenje predstavlja grafičko okruženje koje omogućava otkrivanje idioma na osnovu skupa idiomatskih softverskih projekata i njihovu integraciju u RoseLib.

Kao rezultat upotrebe implementiranog okruženja, RoseLib API se proširuje i sadrži nove metode od kojih je svaka namenjena generisanju određenog idioma. Na taj način se korisniku biblioteke pruža mogućnost lakšeg generisanja koda koji se ponavlja kroz softverske projekte. To može znatno da utiče na ubrzanje razvoja softvera upotrebom MDSE principa.

Dalji razvoj okruženja bi se mogao usmeriti na unapređenje interpretacije idioma koji se dobiju upotrebom RoseLibML projekta. Pronađeni idiomi bi se mogli grupisati na osnovu njihove namene, čime bi se umesto proširivanja postojećih, mogli praviti namenski kompozeri. Time bi se smanjla kompleksnost postojećih kompozera koja može biti rezultat integracije velikog broja idioma. Takođe, mogla bi se obezbediti i funkcionalnost automatskog predlaganja naziva idioma i njegovih metavarijabli na osnovu konteksta. Time bi se uklonila potreba za popunjavanjem tih informacija od strane korisnika, čime bi se ubrzao sam proces integracije.

6. LITERATURA

[1] M. Brambilla, J. Cabot and M. Wimmer, *Model-Driven Software Engineering in Practice: Second Edition*. Morgan & Claypool, 2017.

[2] The .NET Compiler Platform SDK, dostupno na: https://docs.microsoft.com/en-us/dotnet/csharp/roslyn-sdk/ (pristupano u oktobru 2021.)

[3] N. Todorović, A. Lukić, B. Zoranović, R. Vaderna, Ž. Vuković and S. Stoja, „RoseLib: A Library for Simplifying .NET Compiler Platform Usage“. in: *ICIST 2018 Proceedings*. 2018, pp. 216–221

[4] M. Allamanis and C. Sutton, „Mining idioms from source code“, in: *Proceedings of the 22nd acm sigsoft international symposium on foundations of software engineering*. 2014, pp. 472–483

[5] N. Todorović and A. Lukić, RoseLibML, dostupno na: https://github.com/lukic-aleksandar/RoseLibML, (pristupano u oktobru 2021.)

[6] Language Server Protocol, dostupno na: https://microsoft.github.io/language-server-protocol/ (pristupano u oktobru 2021.)

[7] OmniSharp Language Server Protocol Implementation, dostupno na: https://github.com/OmniSharp/csharp-language-server-protocol (pristupano u oktobru 2021.)

[8] Webview API, dostupno na: https://code.visualstudio.com/api/extension-guides/webview (pristupano u oktobru 2021.)

[9] VSCode Language Client – Server Module, dostupno na: https://www.npmjs.com/package/vscode-languageclient (pristupano u oktobru 2021.)

Kratka biografija:

Marijana Kološnjaji rođena je u Vrbasu 1996. godine. Osnovne akademske studije na Fakultetu tehničkih nauka u Novom Sadu, smer Računarstvo i automatika, upisala je 2015. godine. Diplomirala je 2019. godine i iste godine je upisala Master akademske studije na Fakultetu tehničkih nauka u Novom Sadu, smer Računarstvo i automatika.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/16BE45Petrovic

IMPLEMENTACIJA GENERATORA KODA ZA TROSLOJNE POSLOVNE APLIKACIJE

IMPLEMENTATION OF A CODE GENERATOR FOR THREE-TIER BUSINESS APPLICATIONS

Nikolina Petrović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – SOFTVERSKO INŽENJERSTVO I INFORMACIONE TEHNOLOGIJE

Kratak sadržaj – *U radu je opisan razvoj generatora koda za troslojne poslovne aplikacije. Detaljno je opisana specifikacija i implementacija generatora.*

Ključne reči: *generator koda, poslovne aplikacije, crud operacije, model driven development.*

Abstract – *The paper describes the development of a code generator for three-layer business applications. The specification and implementation of the generator is described in detail.*

Keywords: *code generation, business applications, CRUD operations, model driven development.*

1. UVOD

Napredovanje tehnologije, kao i globalna dostupnost kako računara tako i interneta dovela je do širenja tržišta i povećanog broja krajnjih korisnika. Savremeni marketing iziskuje posedovanje aplikacija dostupnih na internetu, koje omogućavaju proširenje biznisa i veći broj kupaca. U online svetu vlada velika konkurencija u svim sferama poslovanja, i s obzirom na količinu sadržaja koji se svakodnevno plasiraju, vrlo je teško i zahtevno biti jedinstven i drugačiji. Kada smislimo aplikaciju koja je jedinstvena, veoma je bitno da na vreme reagujemo.

Takođe, iz godine u godinu na tržištu se javlja sve veći broj firmi koje se bave razvojem softvera. Performanse su samo jedan aspekt ukupnog kvaliteta softvera i obično nisu i najvažniji. Bitan aspekt izrade svake aplikacije predstavlja vreme izrade softvera, kao i cena koja zavisi od vremena uloženog za izradu istog. Postavlja se pitanje kako povećati produktivnost i skratiti vreme izrade softvera.

U ranim fazama izrada aplikacija, često pišemo kod koji se ponavlja. Vreme, koje predstavlja veoma bitan faktor u izradi aplikacija, koristimo za pisanje takozvanih *CRUD* (create, retrieve, update, delete) operacija koje za svaki entitet izgledaju isto. *CRUD* operacije su operacije koje služe za kreiranje, dobavljanje, izmenu i brisanje entiteta iz softverskog rešenja.

Entiteti su u ranim fazama razvoja aplikacije veoma jednostavni, a radi smanjenja grešaka u kasnijem razvoju, kao i lakšeg održavanja softvera, teži se da entiteti takvi i ostanu. Automatizacijom ranih faza izrada softvera možemo drastično smanjiti vreme izrade istog. Takođe, dobro osmišljen i napravljen model koji će se koristiti za automatizaciju ranih faza može dovesti do izrade celog softvera u rekordnom vremenu. Automazitizaciju razvoja softverskih aplikacija možemo postići razvojem automatskih generatora aplikacija ili generatora određenih delova aplikacija (klasa, kontrolera, repozitorijuma..). Takvi generatori će smanjiti količinu rada programera, povećati njegovu produktivnost i smanjiti vreme izrade samog softverskog rešenja.

Tema rada prestavlja razvoj generatora koda koji će omogućiti korisnicima (programerima) automatizovanje ranih faza razvoja softvera i samim tim povećati brzinu izrade programskog rešenja. Generator će omogućiti generisanje *CRUD* operacija i šablona za prikaz podataka. Uz dobro izmodelovan ulaz u generator koda, korisnik će imati mogućnost da bez bilo kakvih izmena dobije potpuno funkcionalno programersko rešenja. .

2. KORIŠĆENE TEHNIKE I TEHNOLOGIJE

Generator koda predstavlja aplikaciju koja služi za automatizovano kreiranje drugih softverskih komponenti. Svaki generator koristi određeni ulaz i na osnovu definisanih pravila generiše izlaz iz generatora.

2.1. Model generatora koda

Generator, koji se još naziva i procesor šablona, koristi dati model podataka ili šablonske stranice, integriše ih na taj način što odgovarajuće podatke modela postavlja u za to označena mesta na šablon stranicama i proizvodi odgovarajuć izlaz.

Generator koda koji će biti implementiran kao ulaz koristi *UML* dijagram. Izlaz iz generatora koda će biti *SpringBoot* [1] modeli, kontroleri i repozitorijumi, kao i *Freemarker* [2] šabloni za grafički prikaz entiteta.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bila prof. dr Gordana Milosavljević.

Inženjerstvo vođeno modelima predstavlja format za pisanje i implementaciju softvera brzo, efikasno i uz minimalne troškove. Ovaj pristup se fokusira na konstrukciju softverskog modela. Model je dijagram koji određuje kako bi softverski sistem trebalo da radi pre generisanja koda.

Paradigma modeliranja smatra se efikasnom ako njeni modeli imaju smisla sa gledišta korisnika koji je upoznat sa domenom i ako mogu poslužiti kao osnova za implementaciju sistema.

Modeli su razvijeni kroz opsežnu komunikaciju između menadžera proizvoda, dizajnera, programera i korisnika domenskih aplikacija. Kako se modeli približavaju završetku, omogućuju razvoj softvera i sistema.

Ovaj princip daje prednosti u produktivnosti u odnosu na druge razvojne metode jer model pojednostavljuje inženjerski proces.

Cela logika je napisana u *UML* dijagramima koji ne zavise od platforme na kojoj će aplikacija biti korištena, te je ovaj način pisanja softvera za višekratnu upotrebu. Takođe, prednost ovakvog pristupa jeste to što omogućava trenutno validaciju korisničkih zahteva i samim tim smanjuje mogućnost greške u ranim fazama. Radikalno smanjuje razvoj, vreme i troškove poslovnih aplikacija.

3. SPECIFIKACIJA

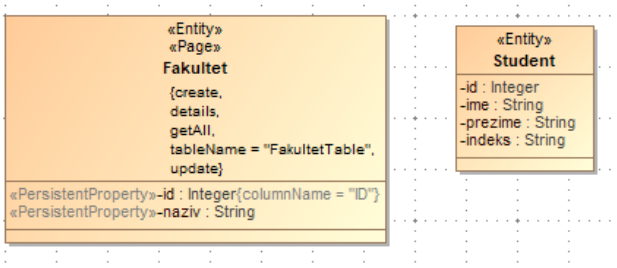
Generator koda koji će biti implementiran predstavlja *CRUD* generator *Java Spring* aplikacije, uz podršku i kreiranje *FreeMarker* šablona. Specifikacija entiteta, kao i međusobne relacije između istih, se zadaje u vidu *UML* dijagrama.

Semantika ovog dijagrama će biti objašnjena u nastavku ovog poglavlja. Takođe, pored specifikacije ulaznog dela generatora, biće specificiran i izlaz generatora.

Slika 3.1. *Model generatora koda*

Ulaz u generator koda, kao što je ranije rečeno, predstavlja *UML* dijagrtam. Kako bi generator koda ispunio svoj maksimalni kapacitet, i količina generisanog koda bila dosta veća u odnosu na količinu koda napisanu od strane programera, posebnu pažnju je potrebno obratiti na prvi korak – kreiranje dijagrama. Dobro struktuirani ulazni podaci će rezultirati kvalitetnim i upotrebljivim rezultatima na izlazu.

Na slici 3.2. prikazan je pojednostavljen ulaz u generator koda. *UML Stereotypes* [3] su mehanizmi koji služe za proširivanje vokabulara *UML*-a, kako bi omogućili kreiranje novih elemenata modela. U praksi, primena odgovarajućih stereotipa modele može učiniti razumljivim i čitljivijim.



Slika 3.2. *Pojednostavljen ulaz u generator koda*

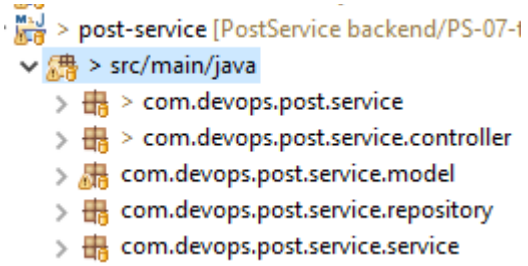
Za potrebe generatora koda opisanog u radu korišćeni su stereotipi *Entity* i *Page*.

Entity stereotip – generatoru koda govori da klasa koja ga sadrži u datom modelu predstavlja entitet. Za dati entitet biće kreirana odgovarajuća *Java* klasa – model.

Page stereotip – generatoru koda govori da je u pitanju klasa koja će predstavljati stranicu u generisanoj aplikaciji. Za klase koje sadrže ovaj stereotip će, pored modela (koji će biti kreiran kod *entity* stereotipa) biti kreirani i odgovarajući kontroler, repozitorijum i servis kako bi se omogućila komunikacija sa bazom podataka. Takođe, biće kreiran i *FreeMarker* šablon za grafički prikaz svih operacija. Očekivani izlaz iz generatora koda predstavlja *Java* izvorni kod i *FreeMarker* šabloni.

Nakon modelovanja *UML* dijagrama koji predstavljaju ulaz u generator koda, korisnik generatora bira putanju na kojoj želi da kod bude generisan, kao i naziv paketa u kom želi da klase budu generisane.

Na ovaj način, korisnik ima potpunu slobodu i generator koda može da koristi u već postojećim projektima, kao i u već implementiranim paketima. Kako bi se generisane klase razlikovale od postojećih, dodat je sufiks *Gen* (skraćeno od generisan), kao i propratni komentar u svakoj od klasa.



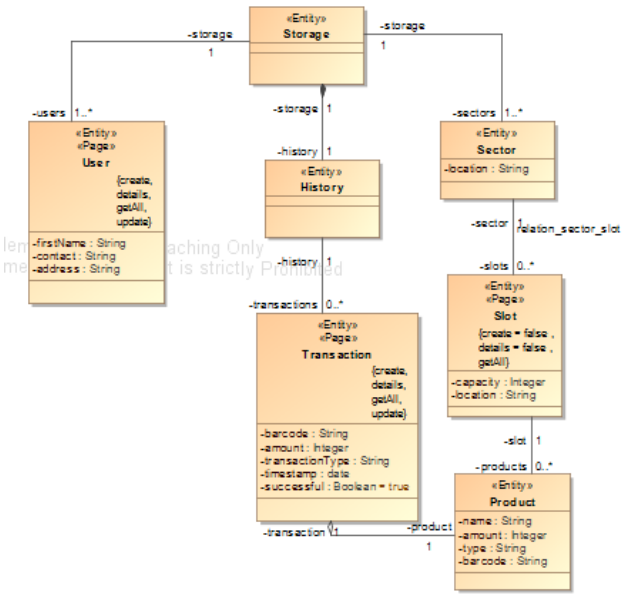
Slika 3.2. Prikaz strukture paketa

4. IMPLEMENTACIJA

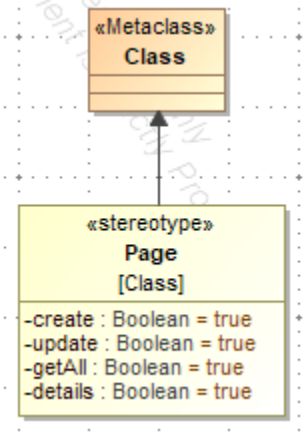
Ulaz u generator koda predstavlja UML dijagram. Na slici 4.1. prikazan je potpuno funkcionalan dijagram, koji je korišten kao primer za potrebe generatora, a čija će primena biti detaljnije opisana u narednim poglavljima.

Za opis klasa su korišteni *UML* stereotipi. Kao što je ranije rečeno, glavna dva tipa stereotipa neophonda za izradu i razumevanje generatora su *Entity* i *Page*. Sami stereotipi predstavljaju proširenje postojeće metaklase.

Stereotip *Page* nam govori da klasa na kojoj se on nalazi treba da bude stranica, odnosno da je neophodno kreirati odgovarajući repozitorijum, servis i kontroler, kao i šablone.



Slika 4.1. Primer ulaza u generator koda



Slika 4.2. Stereotip Page

Na slici 4.2. prikazan je stereotip Page, sa svojim označenim vrednostima. Objašnjenje označenih vrednosti:

1. *create* – ukoliko je vrednost *true*, kreiraće se sve neophodne metode za kreiranje entiteta opisanog datom klasom
2. *update* – ukoliko je vrednost taga *true*, kreiraće se sve neophodne metode za izmenu entiteta opisanog klasom
3. *getAll* – ukoliko je vrednost tačna, kreiraće se sve neophone metode za vraćanje liste svih objekata tipa datog entiteta
4. *details* – ukoliko je vrednost *true* biće kreirana

Generator koda generiše kontrolere, servise, modele i repozitorijume koji su definisani ulazom u generator. Kreiraju se neophodne klase, kao i neophodne zavisnosti među istim.
Metode koje su generisane unutar kontrolera i servisa su sledeće:

1. *findById* – pronalazak entiteta pomoću jedinstvenog identifikatora
2. *add* – dodavanje novog entiteta
3. *update* – izmena postojećeg entiteta
4. *delete* – brisanje entiteta
5. *findAll* – pronalazak svih entiteta

.
Takođe, generator koda kreira *FreeMarker* šablone koji će služiti za grafički prikaz aplikacije. U zavisnoti od vrednosti stereotipa, biće kreirani šabloni za dodavanje, prikaz svih, detaljni prikaz jednog entiteta, kao i izmenu entiteta sa unapred popunjenim podacima entiteta kojeg želimo da izmenimo.

5. PRIMENA GENERATORA KODA

U poglavlju koje sledi biće opisana dva projekta u čijim procesima implementacije je iskorišten generator koda. Statistički prikaz će pokazati koja količina koda je generisana i koliko je proces izrade ubrzan.

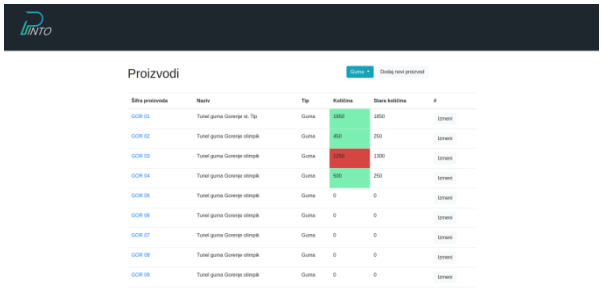
Aplikacija za skladište - U specifikaciji aplikacije dato je da je neophodno kreirati skladište za proizvode. U pitanju je firma čiji se asortiman deli na proizvode od plastike i proizvode od gume. Od operacija bilo je potrebno kreirati dodavanje novog proizvoda, prikaz svih proizvoda (kao i sortiranje po tipu), izmenu proizvoda (akcenat je stavljen na količinu) i prikaz detalja odabranog proizvoda. Zahtev korisnika je da u tabeli, ukoliko je prethodna količina bila manja od nove unete količine to polje bude crvene boje, kako bi mu signaliziralo da je potrebno da dopuni skladište.

Ukoliko su dodati novi proizvodi, te je nova količina veća od prethodne, potrebno je polje obojiti u zelenu boju. Takođe, na zahtev korisnika u tabeli je dodato polje Izmeni koje ga direktno vodi do stranice za izmenu proizvoda kako bi se proces izmene dodatno ubrzao.

	Model	Repozitorijum	Servis	Kontroler
Broj ručno pisanih metoda	0	1	1	1
Broj generisanih metoda	0	0	3	4
Broj generisanih linija koda (ukupno za sve entitete)	44	14	29	54
Broj ručno pisanih linija koda (ukupno za sve entitete)	0	1	3	4
Procenat generisanog koda	100%	93%	90%	93%

Slika 5.1. Statistički prikaz generisanog koda

Na slici 5.2. prikazan je izgled aplikacije za čiju implementaciju je korišten generator koda.



Slika 5.2. Prikaz aplikacije za skladište

Platforma za deljenje fotografija na internetu - Tema zadatka jeste bila razvijanje platforme (društvene mreže) za deljenje fotografija na internetu. Arhitekturu projekta je bilo potrebno razviti u vidu mikroservisa.
Mikroservisna arhitektura je princip razvoja aplikacija u obliku malih, izdvojenih i nezavisnih servisa koji komuniciraju putem jednostavnih mehanizama kao što je HTTP API.

	Model	Repozitorijum	Servis	Kontroler
Broj ručno pisanih metoda	0	2	4	4
Broj generisanih metoda	0	0	3	4
Broj generisanih linija koda (ukupno za sve entitete)	124	13	31	54
Broj ručno pisanih linija koda (ukupno za sve entitete)	0	2	37	18
Procenat generisanog koda	100%	86%	46%	75%

Slika 5.3. Statistički prikaz generisanog koda *Post* servisa

	Model	Repozitorijum	Servis	Kontroler
Broj ručno pisanih metoda	0	2	4	8
Broj generisanih metoda	0	0	3	4
Broj generisanih linija koda (ukupno za sve entitete)	75	13	31	54
Broj ručno pisanih linija koda (ukupno za sve entitete)	0	2	76	53
Procenat generisanog koda	100%	86%	29%	50.4%

Slika 5.4. Statistički prikaz generisanog koda *Account* servisa

	Model	Repozitorijum	Servis	Kontroler
Broj ručno pisanih metoda	0	6	5	5
Broj generisanih metoda	0	0	6	8
Broj generisanih linija koda (ukupno za sve entitete)	122	27	58	108
Broj ručno pisanih linija koda (ukupno za sve entitete)	0	6	51	28
Procenat generisanog koda	100%	81%	53%	73%

Slika 5.5. Statistički prikaz generisanog koda *Account* servisa

Campaign service, *Post service* i *Account service* servisi predstavljaju ključni deo aplikacije i deo aplikacije na kome je korišten generator koda.

Kako je komunikacija između servisa bila napravljena u vidu mikroservisa, relacije između entiteta bile su ručno čuvane u vidu jedinstvenih identifikatora i bile pozivane kada je to bilo neophodno.

6. ZAKLJUČAK

Osnovni motiv za razvoj generatora koda predstavlja povećavanje produktivnosti vremena utrošenog od strane programera za izradu projekta automatizacijom ranih faza izrade. Automatizacija svakodnevnih zadataka i poslova u velikoj meri može povećati efikasnost kako pojedinca, tako i firme. Generator koda opisan u radu u velikoj meri automatizuje rane faze izrade svakog web rešenja napisanog u pomenutim tehnologijama. Pored toga, zbog ulaza u sistem koji predstavlja UML dijagram programeru omogućava vizualni pregled klasa i lakše uviđanje potencijalnih grešaka ili mogućih poboljšanja samog modela.
Kreiranjem modela, kontrolera, servisa, repozitorijuma i FreeMarker šablona, generator koda nakon generisanja programeru daje potpuno funkcionalno programsko rešenje.
Generator koda je moguće proširiti dodavanjem metoda koje se, pored CRUD operacija, veoma često koriste u programskim rešenjima – metode za filtriranje i sortiranje sadržaja. Idealno rešenje bi bilo da se korisniku omogući odabir metoda koje su mu neophodne, kao što je urađeno za CRUD operacije pomoću označenih vrednosti. Pored metoda za filtriranje i sortiranje, generator bi bilo poželjno proširiti i metodama za pretragu po određenim parametrima ili boolean vrednostima (prikaži sve aktivne korisnike,..).

7. LITERATURA

[1] *Spring* Boot, https://spring.io/projects/spring-framework, preuzeto septembra 2021.
[2] *FreeMarker*, https://freemarker.apache.org//, preuzeto septembra 2021.
[3] UML Stereotypes, https://www.visual-paradigm.com/tutorials/how-to-create-stereotyped-model-element.jsp, preuzeto septembra 2021

Kratka biografija:

Nikolina Petrović rođena je 03. marta 1997. godine u Beogradu. U Inđiji je završila osnovnu školu „Dušan Jerković“. Nakon toga upisuje računarsku gimnaziju „Smart“ u Novom Sadu i završava je 2016. godine. Upisuje Fakultet tehničkih nauka, smer Softversko inženjerstvo i informacione tehnologije. Studije završava u roku, 2020. godine. Iste godine upisuje master akademske studije, smer Softversko inženjerstvo i informacione tehnologije.

U realizaciji Zbornika radova Fakulteta tehničkih nauka u toku 2021. godine učestvovali su sledeći recenzenti:

Aco Antić
Aleksandar Erdeljan
Aleksandar Kovačević
Aleksandar Kupusinac
Aleksandar Ristić
Bato Kamberović
Biljana Njegovan
Bogdan Kuzmanović
Bojan Batinić
Bojan Lalić
Bojan Tepavčević
Bojana Beronja
Branislav Atlagić
Branislav Nerandžić
Branka Nakomčić
Branko Milosavljević
Branko Škorić
Damir Đaković
Danijela Ćirić
Danijela Gračanin
Danijela Lalić
Darko Čapko
Darko Marčetić
Darko Reba
Dejan Ecet
Dejan Jerkan
Dejan Ubavin
Dejana Nedučin
Dragan Ivanović
Dragan Jovanović
Dragan Ivetić
Dragan Jovanović
Dragan Kukolj
Dragan Pejić
Dragan Šešlija
Dragana Bajić
Dragana Konstantinović
Dragana Šarac
Dragana Štrbac
Dragoljub Šević
Dubravka Bojanić
Dušan Dobromirov
Dušan Gvozdenac
Dušan Kovačević
Dušan Uzelac
Duško Bekut
Đorđe Ćosić
Đorđe Lađinović
Đorđe Obradović
Đorđe Vukelić
Đula Fabian
Đura Oros
Đurđica Stojanović
Filip Kulić
Goran Sladić
Goran Švenda
Gordana Milosavljević
Gordana Ostojić
Igor Budak
Igor Dejanović
Igor Karlović
Igor Peško
Ivan Beker
Igor Maraš
Ivan Mezei
Ivan Todorović
Ivana Katić
Ivana Kovačić
Ivana Maraš
Ivana Miškeljin
Jasmina Dražić
Jelena Atanacković Jeličić
Jelena Borocki
Jelena Demko Rihter
Jelena Radonić
Jelena Slivka
Jelena Spajić
Jovan Petrović
Lazar Kovačević
Leposava Grubić Nešić
Livija Cvetičanin
Ljiljana Vukajlov
Ljiljana Cvetković
Ljubica Duđak
Maja Turk Sekulić
Marinko Maslarić
Marko Marković
Marko Todorov
Marko Vekić
Maša Bukurov
Matija Stipić
Milan Čeliković
Milan Mirković
Milan Rapajić
Milan Segedinac
Milan Simeunović
Milan Trifković
Milan Trivunić
Milan Vidaković
Milena Krklješ
Milica Kostreš
Milica Miličić
Mijodrag Milošević
Milovan Lazarević
Miodrag Hadžistević
Miodrag Zuković
Mirjana Damnjanović
Mirjana Malešev
Miroslava Radeka
Mirko Borisov
Miro Govedarica
Miroslav Hajduković
Miroslav Kljajić
Miroslav Popović
Miroslav Zarić
Mitar Jocanović
Mitar Đogo
Mladen Kovačević
Mladen Tomić
Mladen Radišić
Nebojša Brkljač
Neda Milić Keresteš
Nemanja Stanisavljević
Nemanja Sremčev
Nikola Đurić
Nikola Jorgovanović
Nikola Radaković
Ninoslav Zuber
Ognjen Lužanin
Peđa Atanasković
Petar Malešev
Platon Sovilj
Radivoje Dinulović
Radomir Kojić
Radovan Štulić
Relja Strezoski
Slavica Mitrović
Slavko Đurić
Slobodan Dudić
Slobodan Krnjetin
Slobodan Morača
Sonja Ristić
Srđan Kolaković
Srđan Popov
Srđan Vukmirović
Staniša Dautović
Stevan Gostojić
Stevan Milisavljević
Stevan Stankovski
Strahil Gušavac
Svetlana Bačkalić
Svetlana Nikoličić
Tanja Kočetov
Tatjana Lončar - Turukalo
Uroš Nedeljković
Valentina Basarić
Velimir Čongradec
Veran Vasić
Veselin Perović
Višnja Žugić
Vladimir Katić
Vladimir Mučenski
Vladimir Strezoski
Vlado Delić
Vlastimir Radonjanin
Vojin Ilić
Vuk Bogdanović
Zdravko Tešić
Zoran Anišić
Zoran Brujić
Zoran Čepić
Zoran Jeličić
Zoran Mitrović
Zoran Papić
Željen Trpovski
Željko Jakšić