



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



ЗБОРНИК РАДОВА ФАКУЛТЕТА ТЕХНИЧКИХ НАУКА

Едиција: Техничке науке - зборници

Година: XXXVI

Број: 7/2021

Нови Сад

Едиција: „Техничке науке – Зборници“

Година: XXXVI

Свеска: 7

Издавач: Факултет техничких наука Нови Сад

Главни и одговорни уредник: проф. др Раде Дорословачки, декан Факултета техничких
Наука у Новом Саду

Уредништво:

Проф. др Раде Дорословачки

Проф. др Александар Купусинац

Проф. др Срђан Колаковић

Проф. др Борис Думнић

Проф. др Дарко Стефановић

Проф. др Себастиан Балаш

Проф. др Драган Ружић

Проф. др Мирослав Кљајић

Проф. др Дубравко Ђулибрк

Проф. др Дејан Убавин

Проф. др Миодраг Ђукић

Проф. др Мирјана Дамњановић

Проф. др Јелена Атанацковић Јеличић

Проф. др Властимир Радоњанин

Проф. др Драган Јовановић

Проф. др Мила Стојаковић

Проф. др Ливија Цветићанин

Проф. др Драгољуб Новаковић

Проф. др Теодор Атанацковић

Редакција:

Проф. др Александар Купусинац, главни
уредник

Проф. др Жељен Трповски, технички
уредник

Проф. др Дарко Стефановић

Проф. др Драгољуб Новаковић

Доц. др Иван Пинђер

Бисерка Милетић

Језичка редакција:

Бисерка Милетић, лектор

Софија Рацков, коректор

Мр Марина Катић, преводилац

Савет за библиотечку и издавачку делатност ФТН,
проф. др Милан Мартинов, председник.

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

CIP-Каталогизација у публикацији
Библиотека Матице српске, Нови Сад

378.9(497.113)(082)

62

ЗБОРНИК радова Факултета техничких наука / главни и одговорни уредник

Раде Дорословачки. – 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 седма овогодишња свеска часописа „Зборник радова Факултета техничких наука“.

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

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

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

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

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

У овом броју штампани су радови студената мастер студија, сада већ мастера, који су радове бранили у периоду од 30.10.2020. до 26.04.2021. год., а који се промовишу 18.05.2021. 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 7. објављени су радови из области:

- електротехнике и рачунарства.

У свесци са редним бројем 6. објављени су радови из области:

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

У свесци са редним бројем 8. објављени су радови из области:

- инжењерског менаџмента,
- инжењерства заштите на раду и заштите животне средине,
- мехатронике,
- математике у техници,
- геодезије и геоматике,
- регионалне политике и развоја,
- управљања ризиком од катастрофалних догађаја и пожара и
- инжењерства информационих система.

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

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

„Високо место у друштву најбољих“

Уредништво

SADRŽAJ

STRANA

Radovi iz oblasti: Elektrotehnika i računarstvo

1. Mihailo Mandić, Milan Vidaković, UPOTREBA GRAPHQL TEHNOLOGIJE ZA RAZVOJ VEB APLIKACIJA	1163-1166
2. Filip Savić, INTEGRACIJA MAINFLUX PLATFORME U SISTEM ZA UPRAVLJANJE ENERGETSKOM POTROŠNJOM U PAMETNOJ KUĆI	1167-1170
3. Siniša Arslanović, UPOREDNA ANALIZA METODA PROJEKTOVANJA ASINHRONIH MOTORA	1171-1174
4. Aleksandar Kahriman, UTICAJ RAZVOJA VOĐENOG TESTOVIMA NA DIZAJN SOFTVERA	1175-1178
5. Ognjen Vlajić, VEB PLATFORMA ZA TESTIRANJE ZNANJA SA PRAĆENJEM POKRETA OKA	1179-1182
6. Vladimir Olear, Dejan Jerkan, SNAGE U SLOŽENOPERIODIČNIM REŽIMIMA ELEKTROENERGETSKOG SISTEMA	1183-1186
7. Aleksandra Milivojević, TESTIRANJE GRAPHQL APLIKACIJA	1187-1190
8. Ivana Marković, SOLID RADNI OKVIR	1191-1194
9. Немања Лазих, Владо Поробић, ПРИМЕРИ ПРИМЕНЕ SCADA СИСТЕМА У ИНДУСТРИЈИ НАФТЕ И ГАСА	1195-1198
10. Sreten Petrović, STRATEGIJE ZA RAD SA NEDOSTAJUĆIM VREDNOSTIMA	1199-1202
11. Зоран Вуковић, Владимир Катић, ПРИМЕНА СОФТВЕРСКИХ АЛАТА ЗА ПРОРАЧУН НАПАЈАЊА СОЛАРНОМ ЕНЕРГИЈОМ КУЋЕ У ДВОРОВИМА	1203-1206
12. Marko Živković, Dejan Jerkan, PREDLOG REŠENJA ZA MONITORING STANJA ASINHRONOG MOTORA	1207-1210
13. Rade Jerkić, Dragan Milićević, POZICIONA KONTROLA U ELEKTROMOTORNIM POGONIMA - ANALIZA TIPIČNOG INDUSTRIJSKOG REŠENJA	1211-1215
14. Olga Savić, IZAZOVI U RAZVOJU iOS APLIKACIJA VISOKE UPOTREBLJIVOSTI	1216-1219
15. Sandra Rajanović, PREDIKCIJA BILLBOARD HITOVA NA OSNOVU AUDIO I TEKSTUALNIH OBELEŽJA PESAMA .	1220-1223
16. Ivana Santrač, Boris Dumnić, RAZVOJ MODELA UPRAVLJANJA PMSM ZA PRIMENU U POGONU ELEKTRIČNIH VOZILA U dSPACE/MICROLABBOX OKRUŽENJU	1224-1227

	STRANA
17. Dimitrije Mijić, ISPITIVANJE RELEJNE ZAŠTITE SREDNJENAPONSKOG IZVODA	1228-1231
18. Stefan Švenderman, PRIMENA IMPEDANSNE SPEKTROSKOPIJE ZA ANALIZU UTICAJA ETARSKIH ULJA NA PROVODNE STRUKTURE FABRIKOVANE NA PAPIRNOJ OSNOVI	1232-1235
19. Dario Kanjo, Neven Kovački, REKONFIGURACIJA DISTRIBUTIVNIH MREŽA PRIMENOM MATEMATIČKIH I HEURISTIČKIH OPTIMIZACIONIH METODA	1236-1239
20. Vuk Bošković, SERVIS ZA IZRADU PODLOGE ZA AKADEMSKO PRIZNAVANJE PERIODA MOBILNOSTI	1240-1243
21. Ilinka Kovačević, Milan Vidaković, INTEGRACIJA FIREBASE CLOUD MESSAGING SISTEMA U VEB I MOBILNE APLIKACIJE	1244-1247
22. David Vuletaš, PROVERA IDENTITETA KORISNIKA KORIŠĆENJEM QR KODA	1248-1251
23. Dejan Bešić, MIKROSERVIS ZA EKSTRAKCIJU TEKSTA IZ WORD I PDF DOKUMENATA	1252-1255
24. Tanja Inđić, MIKROSERVIS ZA PRETRAGU REČI U PROJEKTU REČNIKA SRPSKOG JEZIKA	1256-1258
25. Bogdan Kovačev, RAZVOJ APLIKACIJE ZA AŽURIRANJE MODELA ELEKTROENERGETSKE MREŽE U OKVIRU CLOUD OKRUŽENJA	1259-1262
26. Ђорђе Тошић, РАЗВОЈ И ИНТЕГРАЦИЈА СИЕМ СОФТВЕРСКОГ РЕШЕЊА У SCADA СИСТЕМЕ	1263-1266
27. Dragan Škiljević, PREDIKCIJA POZICIJE FUDBALSKOG IGRAČA UPOTREBOM ALGORITAMA MAŠINSKOG UČENJA	1267-1270
28. Невена Крунић, Зоран Стојановић, УТИЦАЈ СКЛАДИШТА ЕЛЕКТРИЧНЕ ЕНЕРГИЈЕ НА КРАТКЕ СПОЈЕБЕ У НАПОЈНОЈ МРЕЖИ	1271-1274
29. Miloš Nišić, PREDIKCIJA BROJA INDEKSNIH POENA IGRAČA U ABA LIGI SA FOKUSOM NA PRIKUPLJANJU I EKSPLOATIVNOJ ANALIZI PODATAKA	1275-1278
30. Filip Filipović, SISTEM ZA UPRAVLJANJE DISTRIBUIRANIM IZVORIMA ENERGIJE OD STRANE KRAJNJIH POTROŠAČA	1279-1282
31. Marina Mitrović, UTICAJ REGULACIJE NAPONA NA RAD DISTANTNE ZAŠTITE	1283-1286
32. Vuk Vidačić, Vladimir Katić, UTICAJ OBNOVLJIVIH IZVORA ENERGIJE NA PROPADE NAPONA U DISTRIBUTIVNIM MREŽAMA	1287-1290
33. Stefan Milanović, EVALUACIJA UPOTREBLJIVOSTI NOVINSKIH APLIKACIJA ZA ANDROID UREĐAJE	1291-1294
34. Ninoslav Subotić, KONFIGURISANJE I TESTIRANJE MIKROPROCESORSKOG RELEJA ZA ZAŠTITU ENERGETSKOG TRANSFORMATORA	1295-1298
35. Илија Гелић, Дејан Јеркан, PLC ПРОГРАМ ЗА УПРАВЉАЊЕ ЕЛЕКТРОМОТОРНИМ ПОГОНИМА ВЕЛИКЕ ЧАГЕ	1299-1302
36. Marinela Mikša, Željko Trpovski, RAZVOJ 5G MREŽE, STAVOVI I OČEKIVANJA	1303-1306
37. Sofija Pantović, DETEKCIJA OBJEKATA U SAOBRAĆAJU UPOREBOM KONVOLUCIONIХ NEURONSKIH MREŽA	1307-1310

UPOTREBA GRAPHQL TEHNOLOGIJE ZA RAZVOJ VEB APLIKACIJA USE OF GRAPHQL TECHNOLOGY FOR DEVELOPING WEB APPLICATIONS

Mihailo Mandić, Milan Vidaković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Ovaj rad opisuje implementaciju GraphQL tehnologije, u obliku veb aplikacije za prodaju polovnih automobila. U radu su date osnove GraphQL-a, specifikacija rešenja koja je predstavljena UML dijagramima kao i najvažniji implementacioni delovi sa klijentske i serverske strane. Klijentski deo aplikacije je rađen uz pomoć Angular okruženja i Apollo klijenta. Serverski deo je pisan u programskom jeziku Java i korišćen je Spring framework.

Ključne reči: GraphQL, Java, Spring, Veb aplikacija.

Abstract – The main goal of this paper is to present the implementation of GraphQL in a web application for selling used cars. The paper covers the basics of GraphQL. The specification of the application is presented with UML diagrams. Paper also contains the most important parts of implementation logic that covers both client and server-side. Client-side is written using the Angular framework and Apollo client. Server-side is written in Java programming language with the Spring framework.

Keywords: GraphQL, Java, Spring, Web application.

1. UVOD

Najvažniji deo veb aplikacija jeste komunikacija između klijenta i servera. Ukoliko je komunikacija spora i/ili neefikasna bilo kakav napor u optimizaciji aplikacionog koda neće rešiti taj problem. Postoji više načina kako klijent može da razmenjuje informacije sa serverom. Jedan takav način je pomoću SOAP-a (*Simple Object Access Protocol*). Drugi način je pomoću REST-a (*Representational State Transfer*). Resursi su podaci i funkcionalnosti sistema koji se razmenjuju između klijenta i servera putem odabranog stateless protokola. GraphQL je razvio Facebook 2012. godine. Prošlo je 3 godine otkako je postao dostupan za sve, pre toga je bio samo interna tehnologija Facebooka [1]. Kako su mobilne aplikacije Facebooka postajale sve kompleksnije, bilo je potrebno pronaći novi način kako izbeći loše performanse i popraviti stabilnost aplikacija. Količina koda koja treba da se napiše za pripremu podataka, kako na serverskoj strani, tako i na klijentskoj strani, nije bila uopšte mala. Zbog toga su inženjeri Facebooka odlučili da sva postojeća rešenja nisu dovoljno dobra i zbog toga je nastao GraphQL.

NAPOMENA:

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

2. TEORIJSKE OSNOVE GRAPHQL-A

Da bismo koristili GraphQL servis, potrebno je podesiti odabranog klijenta i server. Razmena podataka se obavlja tako što klijent šalje zahtev ka serveru i on odgovara sa željenim podacima. Recimo da imamo klasu *User* koja u sebi ima atribut - *username*, *age*, *location* i atribut objekat *address*. Klasa *Address* sadrži *street*, *city* i *zip* atribut. Klase *User* i *Address* se nalaze na našem serveru i definisane su u GraphQL šemi. U listingu 1. možemo videti primer upita sa klijentske strane gde želimo da obuhvatimo samo neke atribute.

```
query getUser {  
  user {  
    username  
    age  
    address {  
      street  
    }  
  }  
}
```

Listing 1. Primer GraphQL upita.

Kao odgovor sa serverske strane dobijamo JSON (Javascript Object Notation) prikazan u listingu 2.

```
{  
  "data": {  
    "user": {  
      "username": "Ricky"  
      "age": 19,  
      "address": {  
        "street": "Liberty Boulevard"  
      }  
    }  
  }  
}
```

Listing 2. Odgovor od strane servera.

Implementacije GraphQLa variraju u zavisnosti od odabranih klijenata i servera. Osnovni koncepti su šema, upiti, mutacije i ulazna tačka. Ulazna tačka (*endpoint*), je tačka gde su upućeni svi zahtevi prema GraphQL serverskoj strani. GraphQL upit je zahtev poslat od strane klijenta ka serveru radi dobavljanja podataka. Može se posmatrati kao GET zahtev u REST-u. Postoji više načina kako možemo da napišemo upit. Upit se sastoji od operacijskog tipa, operacijskog imena i tela upita ili samo tela upita. Primer koji sadrži sve elemente upita je u listingu 1. *Query* je operacijski tip, *getUser* je operacijsko ime a *user* i ostatak je telo upita. Postoji još dva

operacijska tipa - *mutation* i *subscription*. Isti upit se može napisati i bez operacijskog imena i tipa. Takav upit se naziva kratki upit. Postoje ograničenja kako se šta piše. Operacijski tip je obavezan osim ako se piše kratki upit. U slučaju kratkog upita, operacijsko ime kao i definicije varijabli ne mogu da se navedu. Praksa pokazuje da je izuzetno poželjno koristiti kompletne upite (tip i ime operacije) zbog debugovanja i beleženja serverskih logova.

GraphQL podržava prosleđivanje argumenata. U svako polje možemo proslediti jedan ili više argumenata. Kao argument dozvoljeno je da se proslede različiti tipovi kao na primer skalari, enumi ili kompleksni input object tipovi. U većini aplikacija koje koriste GraphQL, prosleđivanje argumenata u samom upitu nije najbolje rešenje. Argumenti koji se prosleđuju u aplikacijama su (uglavnom) dinamični a ne statični. Ukoliko je potrebno da se proslede dinamički argumenti moraju se koristiti varijable GraphQL-a. Na taj način u samom upitu se nalaze varijable koje dobijaju vrednost tako što se prosledi zasebni rečnik (*dictionary* - par ili skup parova ime:vrednost). Prednost korišćenja varijabli umesto statičkog prosleđivanja argumenata jeste da ne moramo svaki put pisati novi upit ukoliko je potrebno da prosledimo drugu varijablu. Definicija varijable je data u formatu (*\$variableName: type*). Varijabla se označava sa prefiksom \$ i iza nje sledi njen tip. Definicije varijabli podsećaju na definicije argumenata u funkcijama raznih programskih jezika. Sve deklarisanе varijable moraju biti ili skalarnog ili enum ili input object tipa. Ukoliko polje upita prima kompleksan objekat potrebno je znati koji je odgovarajući input tip definisan u šemi koja se nalazi na serveru. Definicije varijabli mogu biti opcione ili obavezne u zavisnosti od toga kako je navedeno u šemi. Obavezne definicije varijabli razlikujemo od opcionih uz pomoć uzvičnika (!) koji stoji iza tipa varijable - (*\$variableName: type!*).

Kao što je ranije napomenuto, upiti služe da klijent dobavi potrebne podatke sa servera po nekom kriterijumu. Ako je potrebno da se izvrši neka manipulacija podacima (bilo da je to unos novih podataka, brisanje ili promena postojećih) koji su smešteni u bazi podataka koja je povezana sa našim serverom, GraphQL nudi mutacije (*mutation*) kao rešenje. Sledeći listing pokazuje jedan primer mutacije sa propratnim rečnikom.

```
mutation loginUser($username: String!, $password: String!) {
  userLogin (username: $username, password: $password) {
    id
    age
  }
}
```

```
{
  "username": "Ricky",
  "password": "25454sda78931"
}
```

Listing 3. Primer mutacije sa propratnim rečnikom varijabli.

Mutacija iz prethodno navedenog listinga prima parametre *username* i *password* tipa *String* koji su obavezni jer je prisutan uzvičnik. Mutacije kao i upiti mogu da vrate polja. Ta polja moraju da se poklapaju sa

definicijom tipa objekta na koji se mutacija odnosi u šemi. Kao povratni odgovor servera dobija se *id* i *age* polje koji se mogu iskoristiti na klijentskoj strani. Mutacije ne moraju nužno da primaju bilo kakve argumente. Takođe, mutacije mogu da primaju kompleksne objekte a ne samo skalare. Ako se prosleđuju kompleksni objekti, potrebno je koristiti input object tip. To je specijalna vrsta object tipa.

Svaki GraphQL servis u sebi sadrži šemu. Da bi se zadovoljio neki upit ili mutacija potrebno je taj zahtev validirati i izvršiti u odnosu na šemu. Šema služi da se u potpunosti opiše API servera što omogućava klijentima da znaju koje operacije se mogu izvršiti na serveru. To znači da su u šemi definisani svi tipovi objekata i njihova polja koji reprezentuju podatke. Šema je definisana pomoću GraphQL šematskog jezika (*GraphQL schema language*). Taj jezik se naziva i SDL (*Schema Definition Language*) [2].

Tip koji se najčešće sreće je object tip. Object tip reprezentuje objekat koji se dobavlja iz servisa i u sebi sadrži sva odgovarajuća polja. Primer object tipa je dat u narednom listingu.

```
type User {
  username: String!
  password: String
  age: Int
  location: String
  buddies: [User]
}
```

Listing 4. Definicija object tipa u šemi.

GraphQL polja odgovaraju atributima klase u izabranom programskom jeziku. Samo polja koja su definisana u nekom tipu mogu da se pojave u nekom upitu ili mutaciji upućenom ka serveru. Uzvičnik (!) pored nekog polja označava da je vrednost tog polja uvek pristupa (nije null) ako se potražuje.

Šema u sebi pored definicija svih tipova koje podržava sadrži i *root operation* tipove koje podržava u odeljku *schema*. Ako su korenske operacije (*root operation*) *query*, *mutation* i *subscription* definisane pomoću tipa koji se zove *Query*, *Mutation* i *Subscription* respektivno nije potrebno definisati *schema* sekciju. Svaka šema mora da podrži *query* root operation tip i on mora biti object tip [3]. Pored korenskih operacionih tipova i object tipova postoje još i skalarni tipovi. Skalarni tipovi u GraphQL-u služe za konkretnu definiciju polja pored ostalih tipova. Na primer u programskom jeziku Java takvi tipovi se nazivaju primitivni tipovi. Postoje predefinisani skalari kao i mogućnost da mi sami definišemo naše skalare. Predefinisani skalari su *Int*, *Float*, *Boolean*, *String* i *ID*. Posebni tip skalara su i enumeracije. Enumeracije su ograničene na određeni set vrednosti.

GraphQL u sebi podržava interfejsе. Interfejs je apstraktni tip koji u sebi sadrži određen broj polja. Kada se interfejs nasledi, sva njegova polja moraju biti sadržana u tipu koja ga nasleđuje. Pored interfejsa, GraphQL definiše i unije. Razlika između unije i interfejsa jeste što unije ne specificiraju zajednička polja.

U prethodnim pasusima smo pomenuli input object tip. On služi za plasiranje kompleksnih objekata u upite i mutacije. Input object definiše set ulaznih polja. Umesto ključne reči *type*, koja se koristi pri definiciji object tipa,

za input object tip je potrebno napisati *input*. Važno je napomenuti da input object tip i object tip imaju različite namene iako možda na prvi pogled izgledaju vrlo slično po definiciji u šemi. Neretko se dešava da object tip, iako deli neka ili možda čak i većinu polja sa input object tipom, sadrži druge object tipove i tako se stvara jedna vrlo kompleksna struktura podataka. Object tip može u sebi da sadrži reference prema interfejsima i unijama a njegova polja mogu da definišu argumente u sebi. Interfejsi i unije su apstraktni tipovi za koje nema smisla da se šalju sa klijentske strane.

2.1. REST i GraphQL

REST (*Representational State Transfer*) je stil arhitekture sistema koji smo koristi u izradi našeg projekta. Za pristupanje nekom resursu koji se krije iza REST API-ja, potrebno je da klijent zna putanju do tog resursa kao i da specificira metodu šta želi da uradi s tim resursom. Metode koje klijent može da iskoristi su GET, POST, PUT, DELETE i ostale. U REST API arhitekturi, server definiše koji će podaci biti vraćeni dok kod GraphQL servera, server samo definiše dostupne podatke a klijent specificira koje podatke želi da budu vraćeni. Kada je u pitanju rukovanje greškama, REST se oslanja na podrazumevane status kodove HTTP-a (404, 401, 500, itd.). Kod GraphQL-a, statusni kod je uvek 200 OK bez obzira na odgovor [4]. U slučaju greške klijentu se šalje odgovor sa tekstom greške. U implementaciji GraphQL-a, postoji mogućnost definisanja greški i poruka koje server može da vrati.

Pored određenih prednosti naspram REST-a, GraphQL donosi i određen broj mana. Na primer, . Ako je potrebno koristiti keširanje podataka u GraphQL-u mora se koristiti eksterni mehanizam na klijentskoj strani dok REST koristi HTTP koji su sebi ima ugrađeno keširanje. Još jedna mana GraphQL-a je format odgovora. Kao odgovor samo podržava JSON format dok REST podržava JSON, XML, HTML i druge [5].

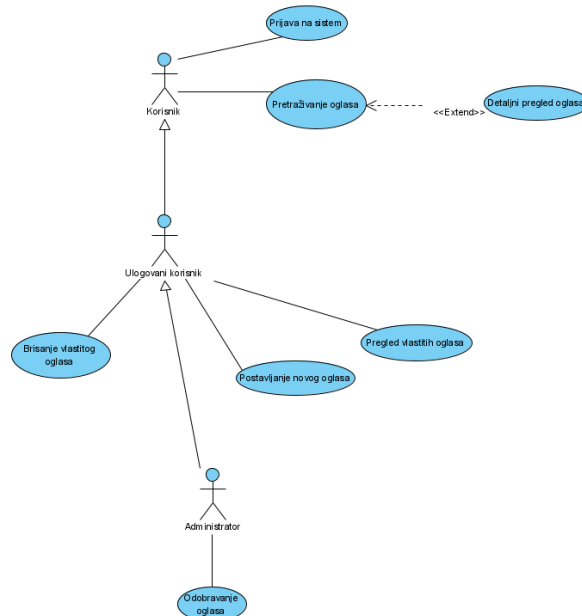
2.2. SOAP i GraphQL

U poređenju sa SOAP-om, GraphQL deli neke elemente poput jedne ulazne tačke, endpoint-a, za pristup podacima. U obe tehnologije potrebno je deklarirati tipove podataka. Takođe, obe tehnologije dele i istu manu a to je nedostatak ugrađenih mehanizama za keširanje. Iako SOAP može da koristi HTTP kao i REST, ograničen je samo na upotrebu POST metoda koja ne kešira. Prednost GraphQL-a u odnosu na SOAP su bolje performanse dok SOAP ima prednost za održavanje starijih sistema koji su bili aktuelni po pojavljivanju SOAP-a.

3. SPECIFIKACIJA ZADATKA

Kao pomoć u opisu specifikacije zadatka sačinjeni su UML dijagram korišćenja i dijagram sekvence. UML (*Unified Modeling Language*) je jezik za modelovanje u softverskom inženjerstvu. Cilj UML-a je da pruži standardan način predstavljanja dizajna nekog softverskog sistema. Dijagram korišćenja potpada pod dijagrame ponašanja i opisuje neke od veza između aktera, slučajeva korišćenja i sistema. U navedenom dijagramu korišćenja, koji je dat na narednoj slici, postoje tri tipa korisnika. Neprijavljen korisnik (ili samo korisnik) ima opcije da se prijavi na sistem i da pretražuje oglase. Kada nađe

odgovarajuću listu oglasa, omogućeno je da pojedinačan oglas pogleda detaljnije. Kada korisnik izvrši prijavu na sistem, postaje prijavljen korisnik. Neprijavljen korisnik može da vidi svoju listu postavljenih oglasa koji su odobreni od strane administratora. Takođe postoji mogućnost da prijavljen korisnik obriše svoj već postavljeni oglas. Da bi se postavio novi oglas, potrebno je da korisnik bude prijavljen. Administrator sistema ima dodatnu funkciju da odobrava oglase pre nego što postanu vidljivi.



Slika 1. Dijagram slučajeva korišćenja.

Nakon prijave na sistem, korisnik vidi pregled sopstvenih oglasa i opciju da postavi novi oglas. Nakon popunjavanja svih polja oglasa, rezultat te operacije se šalje aplikaciji radi obrade. Aplikacija kontaktira bazu za upis oglasa i oglas se upisuje kao neodobren. Oglasi koji su označeni kao neodobreni dospevaju na administratorski panel. Administrator ima mogućnost da odbije ili odobri neki oglas. U ovom slučaju administrator odobrava oglas. Posle odobravanja oglasa, aplikacija šalje poruku bazi podataka da se status oglasa promeni u odobren. Kao takav, odobren oglas se pojavljuje u listi vlastitih oglasa korisnika kao i u svim pretragama bilo kog korisnika aplikacije.

4. OPIS IMPLEMENTACIJE

Kao što je ranije navedeno, aplikacija se sastoji od klijentskog i serverskog dela.

4.1. Serverski deo

Paketi koji se obično koriste pri izradi serverskog dela su *graphql-spring-boot-starter* i *graphql-java-tools*. Prvi je zadužen da pretvori backend deo aplikacije u GraphQL server a drugi služi za izgradnju GraphQL šeme kao i da olakša rad i implementaciju GraphQL-a. GraphQL server se nalazi na putanji *http://localhost:8080/graphql*. Umesto *graphql*, putanja može da se promeni u proizvoljnu, u *application.properties* fajlu. Šema projekta podržava upite i mutacije kao što je definisano u *schema* odeljku. Sastoji se iz object tipova i input object tipova.

Object tipovi su *Query*, *Mutation*, *Ads*, *Image*, *User*, *CarModel*, *Extras*, *Safety*, *Characteristics* i *Condition*.

Input object tipovi su *AdsInput*, *CarModelInput*, *UserInput*, *ExtrasInput*, *ConditionInput*, *CharacteristicsInput* i *SafetyInput*. Input object tipovi sadrže reč *Input* u svom nazivu. Svi nazivi polja svih object tipova izuzev *Query* i *Mutation* odgovaraju nazivima atributa klasa koja su definisane u projektu. To omogućava automatsko mapiranje GraphQL object tipova sa klasama koje su definisane u Javi. Isto važi i za input object tipove. Uz pomoć GraphQL Java alata [6] koji smo uključili u projekat, omogućeno je mapiranje GraphQL objekata na metode i attribute Java objekata. Za većinu polja koja sadrže skalarne tipove dovoljno je napraviti POJO (*Plain Old Java Object*) sa istim poljima, getter-ima i setter-ima. Kompleksnija polja poput ugrađenih objekata zahtevaju u nekim slučajevima pisanje posebnih Resolver-a. Resolveri su funkcije koje daju vrednost za polje u šemi ili obavljaju neke kompleksne kalkulacije ili nešto treće.

U nekim slučajevima potrebno je implementirati *GraphQLResolver* interfejs. Pored Resolver-a za pojedinačna polja, postoje Resolveri za korenske operacije.

U šemi projekta prisutne su mutacije i upiti kao operacije pa samim tim potrebno je implementirati dva korenska resolvera - *GraphQLQueryResolver* i *GraphQLMutationResolver*.

Uslov za implementiranje *GraphQLQueryResolver* interfejsa je da svako polje iz šeme *Query* objekta ima svoju metodu sa istim imenom [7]. Takođe, svako polje *Mutation* objekta u šemi treba da ima svoju metodu sa istim imenom. Jedna od prednosti GraphQL-a jeste definisanje sopstvenih grešaka i poruka. Ukoliko je to potrebno uraditi, mora se implementirati *GraphQLError* interfejs. Kako GraphQL ima nedostataka demonstrirano je upotrebom REST-a u zadatku. Rad sa fajlovima nije jača strana GraphQL-a i ako je moguće taj deo treba izbeći koristeći neku drugu tehnologiju. S obzirom da je ovo aplikacija za prodaju polovnih automobila bilo je potrebno omogućiti postavljanje slike u oglasu. Kada se slika pošalje sa klijentske strane, prvo dolazi do REST kontrolera. Putanja za upload slike je <http://localhost:8080/image/upload> i potrebno je koristiti HTTP POST metodu.

4.2. Klijentski deo

Za izradu klijentskog dela korišćen je Angular i Apollo klijent. Za korišćenje Apollo klijenta potrebno ga je instalirati i podesiti putanju do GraphQL servera. Za pisanje bilo kakvog GraphQL upita ili mutacije potrebno je da tekst bude obmotan *gql* funkcijom koja se nalazi u paketu *apollo-angular*. U konkretnom slučaju, npr. pretrage oglasa, nakon pritiskanja dugmeta za submit svih podataka za pretragu, podaci se beleže i spremaju se za slanje ka GraphQL serveru. Važno je napomenuti da sva imena varijabli na frontendu moraju odgovarati svim poljima definisanim u šemi i klasama na serveru radi tačnog mapiranja. Poziva se apollo *query* metoda koja izvršava zahtev ka serveru i čeka se njen odgovor radi popunjavanja tabele sa rezultatima. Zapravo šta se ovde dešava jeste da Apollo šalje HTTP POST zahtev serveru sa podacima u JSON formatu. Nakon obrade podataka server šalje podatke takođe u JSON formatu. Kada je u pitanju slanje slike na server, koristi se http klijent. Po

odabiru željene slike se šalje HTTP POST zahtev prema REST kontroleru koji očekuje informacije o slici na <http://localhost:8080/image/upload> putanji. Nakon uspešnog postavljanja slike na server, kao odgovor se uzima jedinstveni identifikator slike da bi bio uključen sa ostalom podacima koji će biti poslani putem Apollo klijenta na server.

5. ZAKLJUČAK

U ovom radu je dato jedno rešenje kako implementirati GraphQL u programskom jeziku Java. Serverski deo je potpomognut Spring okruženjem, dok klijentski je sačinjen od Angulara, Apollo klijenta i programskog jezika Typescript.

Aplikacija omogućava pregled svih oglasa na sistemu kao i postavke vlastitih oglasa za prodaju polovnih automobila. Takođe sadrži sistem za prijavu/registaciju korisnika u sistemu i administratorski deo za sprečavanje zloupotrebe postavljanjem neprimerenih oglasa.

Prikazano rešenje predstavlja samo studiju slučaja, te nije pogodno za komercijalnu upotrebu u trenutnoj formi. Za komercijalnu upotrebu neophodno bi bilo doraditi makar sledeće funkcionalnosti: zaštita korisnika i cele aplikacije od neovlašćenih upada u sistem, kao i podrška komunikaciji između prodavca i kupca.

6. LITERATURA

- [1] Why GraphQL? <https://engineering.fb.com/core-data/graphql-a-data-query-language/>, Oktobar 2020.
- [2] Schema definition language <https://graphql.org/learn/schema/#type-language>, Oktobar 2020.
- [3] Root operation types <https://spec.graphql.org/June2018/#sec-Root-Operation-Types>, Oktobar 2020.
- [4] GraphQL error handling, <https://medium.com/better-practices/rest-soap-graphql-gesundheit-6544053f65cf>, Oktobar 2020.
- [5] REST formats, <http://help.arcgis.com/en/businessanalyst/apis/rest/referen ce/outputFormats.html>, Oktobar 2020.
- [6] GraphQL Java Kickstart, <https://github.com/graphql-java-kickstart/graphql-java-tools>, <https://www.graphql-java-kickstart.com/tools/>, Oktobar 2020.
- [7] Naming conventions, <https://www.baeldung.com/spring-graphql#3-root-query-resolver>, Oktobar 2020.

Kratka biografija

Mihailo Mandić je rođen 1995. godine u Beogradu. Završio je gimnaziju „Isidora Sekulić” u Novom Sadu, društveno-jezički smer. Fakultet tehničkih nauka, smer Računarstvo i automatika, upisao je 2014. godine. Nakon završenih osnovnih studija, upisao je master studije na istom fakultetu.

Milan Vidaković je rođen 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.

**INTEGRACIJA MAINFLUX PLATFORME U SISTEM ZA UPRAVLJANJE
ENERGETSKOM POTROŠNJOM U PAMETNOJ KUĆI****INTEGRATION OF MAINFLUX PLATFORM INTO THE SMART HOUSE ENERGY
CONSUMPTION SYSTEM**

Filip Savić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – VAŠ ODSEK

Kratak sadržaj – Zadatak rada jeste integracija Mainflux platforme u sistem za upravljanje energetskom potrošnjom u pametnoj kući. Mainflux platforma je korišćena zajedno sa Bash scripting-om da obezbijedi dostupnost podataka potrebnih za generisanje realne potrošnje električne energije. Simulacija je rađena pomoću Typhoon HIL softvera. Python programski jezik je korišćen za dobavljanje podataka sa Mainflux-a, stavljanje podataka u Typhoon HIL simulaciju, te upis realne energetske potrošnje u CSV datoteku.

Ključne reči: Mainflux, Typhoon HIL, pametna kuća, Python, IoT

Abstract – This paper deals with the integration of the Mainflux platform into a smart house energy consumption system. Mainflux platform was used together with Bash scripting to provide the necessary data for more real energy consumption simulation. The simulation is done using Typhoon HIL software. Python programming language has been used for software support, as a main integration component. It fetches data from the Mainflux platform, puts it into the Typhoon HIL simulation, and at the end writes the generated energy estimation in a CSV file.

Keywords: Mainflux, Typhoon HIL, smart home, IoT

1. UVOD

Zadatak rada predstavlja integracija Mainflux platforme sa TyphoonHIL simulatorom u cilju ubacivanja realnih podataka o temperaturi i solarnoj iradijaciji u Typhoon HIL simulaciju pametne kuće (eng. „smart home“). Ovaj rad predstavlja dio sistema koji softverskim agentima povezanim na realne pametne uređaje ima za cilj da smanji energetske potrošnje u pametnoj kući. Prvenstveno je bilo potrebno pribaviti i instalirati neophodan softver za korišćenje Mainflux [1, 4] platforme i Typhoon HIL Control Center-a [2].

2. OPIS KORIŠĆENIH TEHNOLOGIJA

Konačno softversko rešenje kreirano je uz pomoć Typhoon HIL Control Center-a i Typhoon HIL API-ja, Mainflux platforme čije su komponente pokretane pomoću Docker [3] softverskog alata, te upotrebom Bash

scripting-a i programskog jezika Python. Podaci o temperaturi i iradijaciji preuzeti su sa sajta australijske kompanije Solcast [5].

Skripta za pribavljanje podataka sa Mainflux platforme i upravljanje Typhoon HIL simulacijom pisane su u programskom jeziku Python. Typhoon-HIL-API [6] je Python modul koji podržava upravljanje simulacijom.

2.1 Mainflux platforma

Mainflux predstavlja skalabilnu, bezbjednu, patent-free i open-source platformu za Internet of Things, napisanu u programskom jeziku Go, koja se distribuira putem softverskih paketa („kontejnera“) Docker tehnologije.

Glavne karakteristike Mainflux-a su:

- brza i skalabilna mikro-servisna arhitektura,
- dobro formiran API (HTTP, MQTT, WebSocket),
- interakcija sa uređajima kroz različite protokole i
- lako pokretanje i distribucija putem Docker alata.

Ove karakteristike omogućavaju brz i jednostavan razvoj pametnih IoT rješenja.

U daljem tekstu će se, zbog jednostavnosti, za Mainflux platformu navoditi samo Mainflux.

Mainflux predstavlja skup servisa od kojih svaki ima važnu ulogu:

- Users – upravljanje korisnicima i autentifikacijom,
 - Things – registrovanje novih uređaja i aplikacija, pravljenje komunikacionih kanala između njih, te kontrola komunikacije,
 - Protocol adapters – (HTTP, MQTT, WebSocket) adapteri koji obezbjeđuju interfejs za pristup komunikacionim kanalima,
 - Normalizer – normalizacija SenML [7] poruka i slanje istih ka tokovima za dalju obradu podataka i
 - Custom consumers – svi servisi koji vrše upis u skladišta podataka (eng. „data store“) i čitanje normalizovanih poruka kroz HTTP API.
- Tri glavna entiteta kojima Mainflux upravlja su:
1. Thing – stvar – predstavlja sve entitete koji komuniciraju kroz Mainflux,
 2. Channel – kanal – predstavlja komunikacioni kanal. Služi kao tema razmjene poruka (eng. „message topic“)
 3. User – korisnik – pravi korisnik sistema (čovjek).
- Predstavljen je svojom adresom e-pošte i lozinkom.

Messaging – razmjena poruka

Mainflux koristi NATS [8] kao glavni stub komunikacije, zbog svojih performansi i malog zauzeća resursa.

NAPOMENA:

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

Mainflux nudi više protokola za razmjenu poruka, od kojih su u realizaciji sistema korišćeni HTTP i MQTT. O MQTT-u i SenML-u će biti nekoliko riječi, a HTTP neće biti opisivan zbog svoje rasprostranjenosti.

SenML

SenML (Sensor Markup Language) specifikacija definiše standardizovan format poruka za opisivanje senzora i njihovih mjerenja. Poruke mogu biti u mnogim formatima, od kojih je u projektu, zbog konciznosti i rasprostranjenosti, korišćen JSON (JavaScript Object Notation). Primjer SenML poruke dat je u Listingu 1.

```
[{"n": "temperature", "u": "Cel", "v": 23.1}]
```

Listing 1. Primjer SenML poruke

Vrijednosti „n“, „u“ i „v“ predstavljaju naziv uređaja, jedinicu, te vrijednost izmjerene veličine, redom. Svako mjerenje se, makar bilo i samo jedno, mora nalaziti unutar liste, koja se u JSON notaciji označava uglastim zagradama - [].

MQTT

MQTT (Message Queuing Telemetry Transport) predstavlja M2M (Machine to Machine) komunikacioni protokol. Dizajniran je za podršku izuzetno lakog (eng. „lightweight“) transporta poruka po modelu „publish/subscribe“. Neki od primjera upotrebe su kod senzora koji komuniciraju sa brokerom putem satelitske veze ili u raznim scenarijima automatizacije domaćinstava i malih uređaja.

Docker

Docker je tehnologija za kreiranje softverskih paketa, odnosno „kontejnera“ za samostalne aplikacije. U njima se nalaze biblioteke i sve ostalo potrebno za pokretanje i izvršavanje aplikacije.

Docker kontejneri omogućavaju „pakovanje“ aplikacije i svega što joj je potrebno za rad u „sliku“ kontejnera (eng. „container image“).

„Slika kontejnera“ je šablon za izvršavanje kontejnera. Nakon što se kontejner kreira, njegovim izvršavanjem upravlja „Docker Engine“, poznat i pod nazivom „Docker Daemon“. Interakcija sa Docker Engine-om omogućena je putem komande „docker“, odnosno interfejsa komandne linije (eng. „command line interface“, CLI).

Instalacija i pokretanje Mainflux-a

Za postupak instalacije koji je opisan u nastavku, potreban je softver cURL i git CLI. Da bi instalirali platformu, prvo je potrebno preuzeti i instalirati Docker i docker compose [9], koji služi za pokretanje aplikacija sastavljenih iz više Docker kontejnera.

Kada su Docker i docker-compose instalirani, moguće je preuzeti i pokrenuti Mainflux platformu komandama sadržanim u listingu 2.

```
> git clone https://github.com/mainflux/mainflux.git
> cd mainflux
> docker-compose -f docker/docker-compose.yml -f
docker/addons/influxdb-reader/docker-compose.yml -f
docker/addons/influxdb-writer/docker-compose.yml up -d
```

Listing 2. Komande za instalaciju Mainflux platforme

Mainflux CLI

Mainflux CLI [10] predstavlja alat za upravljanje korisnicima, stvarima, kanalima i porukama. Alat je moguće preuzeti zasebno sa github stranice izdanja (eng. „releases“) preuzeti tar.gz datoteku te je otpakovati u željenom direktorijumu. Listing 3 prikazuje set komandi za ovaj tip instalacije.

```
> wget
https://github.com/mainflux/mainflux/releases/download/v0.11.0/mainflux-cli_v0.11.0_linux-amd64.tar.gz
> tar xvf mainflux-cli_v0.11.0_linux-amd64.tar.gz
```

Listing 3. Komande za ručno preuzimanje i instalaciju Mainflux CLI alata

Nakon instalacije moguće je koristiti Mainflux CLI ako je pokrenut glavni Mainflux servis. Komande korišćene u ovom projektu su messages za poruke, things za upravljanje stvarima te users za upravljanje korisnicima.

2.2. Bash scripting

Bash scripting [11] je korišćen za skriptu koja kreira čitač CSV datoteke (Mainflux-ov CSV Reader). Skripta se koristi za pravljenje Mainflux entiteta koji su potrebni za pokretanje reader-a. Kreiraju se korisnik, kanal i stvar koja predstavlja sam čitač.

Bash skripta predstavlja datoteku koja sadrži set komandi koje se izvršavaju u komandnoj liniji (eng. „command line“). Bilo koja komanda koja se stavi u skriptu radi potpuno istu stvar kao i kada se samostalno pokrene u komandnoj liniji. Neke od osnovnih komandi su echo za ispis u konzolu, if za kontrolu toka programa te for za programske petlje. Svaka bash skripta u kao prvu liniju shebang (#!) koji označava da slijedi putanja do interpretera ili programa koji će izvršiti skriptu. Putanja do interpretera je obično /bin/bash.

2.3. Typhoon HIL Control Center

Typhoon HIL Control Center [12] predstavlja softverski alat kompanije Typhoon HIL za inženjering sistema zasnovanih na modelu (eng. „Model-Based System Engineering“). Sastoji se od četiri modula: Schematic Editor, HIL SCADA, Typhoon Test IDE i Test Suite. U ovom zadatku Control Center je korišćen za simulaciju pametne kuće i energetske potrošnje u njoj.

2.4. Python i upravljanje simulacijom

Za upravljanje Typhoon HIL simulacijom korišćena je Python biblioteka Typhoon-HIL-API. Pruža razne funkcije koje su podijeljene u četiri grupe: za kontrolu inicijalizacije, za mijenjanje stanja, za preuzimanje informacija iz simulacije te razne Utility funkcije. Listing 4 prikazuje Python skriptu koja uvozi biblioteku, učitava model, pokreće simulaciju te postavlja neke vrijednosti unutar simulacije.

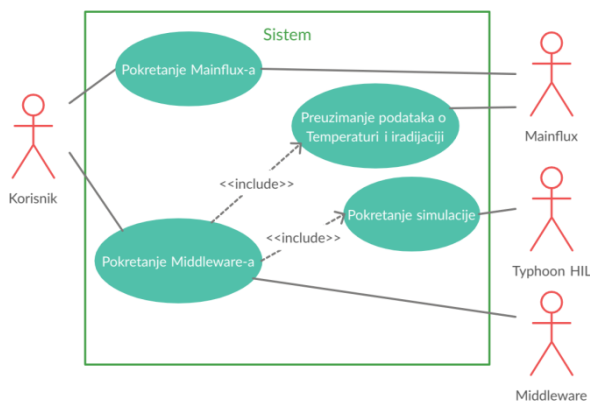
```
import typhoon.api.hil as hil
hil.load_model(file=r'/home/filip/master/typhoon_hil/Target files/model.cpd')
hil.start_simulation()
hil.set_analog_output(5, 'V(Va)', scaling=150.0, offset=5.0)
```

Listing 4. Osnovne komande Typhoon HIL API-ja

3. SPECIFIKACIJA APLIKACIJE

Zadatak predstavlja integraciju Mainflux-a sa Typhoon HIL simulatorom. Cilj ovog rada jeste nadzor, mjerenje i upravljanje potrošnjom električne energije u pametnoj kući. Pošto su trenutno vrijednosti temperature i iradijacije u simulaciji generisane, učitavanje realnih podataka omogućava bolju simulaciju energetske potrošnje.

Podaci o temperaturi i solarnoj iradijaciji ubačeni su u Mainflux iz CSV datoteke, a učitavanje tih podataka u simulaciju realizovano je korišćenjem Python skripte. Slika 1 prikazuje Use case dijagram sistema.



Slika 1. Use case dijagram

4. OPIS IMPLEMENTACIJE

Za potrebe zadatka korišćeni su Mainflux platforma, programski jezik Python te Typhoon HIL API. Mainflux je preuzet sa github stranice Mainflux projekta, a pokretan je docker i docker compose alatima. Integracija je urađena koristeći Python. Typhoon HIL softver za Linux operativni sistem dostavljen je od strane zaposlenih kompanije Typhoon HIL dok je biblioteka Typhoon HIL API preuzeta je sa PyPI repozitorijuma.

4.1. Odabir podataka

Podaci o temperaturi i iradijaciji preuzeti su sa sajta Solcast kompanije. Sa stranice „Historical & TMY“ (eng. „Typical meteorological year“) web sajta Solcast kompanije moguće je preuzeti podatke o temperaturi i solarnoj iradijaciji. Preuzimanje podataka vrši se kreiranjem zahtjeva. Potrebno je unijeti lokaciju za koju nas podaci interesuju te označiti period u godinama ili od jednog do drugog konkretnog datuma. Zatim se biraju parametri kao što su horizontalna iradijacija, azimut, temperatura vazduha, brzina vjetrova itd. Podatke je moguće preuzeti u intervalima od po 5, 10, 15, 30 i 60 minuta. Za svaki skup podataka preuzima se odvojena CSV datoteka.

4.2. Uloga Mainflux-a

Mainflux je u ovom zadatku služio kao repozitorijum podataka te lako proširivo rješenje u slučaju prelaska na realan sistem pametne kuće. Čuva podatke o temperaturi i iradijaciji i čini ih dostupnim za dobavljanje preko interneta. Glavni doprinos Mainflux-a za ovaj zadatak jeste CSV čitač (eng. „reader“). CSV reader se kreira kao Mainflux stvar prikazana na Mainflux kanal. Da bi se kreirao bilo koji Mainflux entitet, potrebno je da postoji korisnik. Za potrebe testiranja rada kreirano je mnogo

korisnika, kanala i reader-a, te je proces automatizovan u jednu bash skriptu koja zbog svoje veličine neće biti prikazana u ovom radu. Skripta kreira korisnika, uzima njegov token, te ga prosljeđuje prilikom kreiranja kanala. Zatim korisnikov token i ID kreiranog kanala šalje u JSON za kreiranje CSV reader-a kao Mainflux stvari. JSON osim toga sadrži i putanju do CSV datoteke te kolone koje se čitaju iz nje.

Da bi se kreirani CSV reader pokrenuo, potrebno je preuzeti kôd za CSV reader preuzima se sa github naloga Darka Draškovića [13]. Zatim je potrebno pokrenuti Mainflux core (glavni servis) i Mongo writer koji predstavlja Mainflux dodatak za skladištenje (eng. „storage“) podataka. Nakon što se pokrenu ovi servisi, može se pokrenuti i sam reader. U terminalu se treba navigirati do csv-dbreader direktorijuma u Mainflux-u sa github-a Darka Draškovića, te pokrenuti komande prikazane na listingu 5.

```
export MF_CSV_ADAPTER_CONFIG_FILE =  
/home/filip/Faks/Master/readers.cfg  
go run main.go
```

Listing 5. Pokretanje CSV reader-a

Prva komanda postavlja putanju do konfiguracione datoteke za reader-e „readers.cfg“, a druga pokreće reader pišan u jeziku Go. Nakon ovoga u terminalu u kome je pokrenut Mongo writer biće prikazano mnoštvo linija koje su indikator da je sve iz CSV datoteke upisano u bazu. Nakon ovoga treba samo pokrenuti Mongo reader koji će omogućiti preuzimanje podataka iz baze preko API-ja.

4.3. Uloga Typhoon HIL-a

U ovom zadatku korišćen je Typhoon HIL Control Center te Typhoon HIL API. Typhoon HIL se koristio za simuliranje pametne kuće i energetske potrošnje unutar nje, ali su podaci o temperaturi i iradijaciji bili generisani. Bilo je potrebno izmijeniti inicijalni model pametne kuće dobijen od strane zaposlenih kompanije Typhoon HIL da bi se u njega mogli ubaciti podaci o temperaturi i iradijaciji. Ovo je urađeno u Schematic Editor-u korišćenjem SCADA Input komponente koja ima mogućnost da joj se programski mijenja vrijednost, čak i tokom trajanja simulacije.

4.4. Kreiranje Middleware-a

Nakon završene pripreme Mainflux i Typhoon HIL strana za komunikaciju, sve što je potrebno za ubacivanje podataka jeste GET zahtjev na Mainflux-ov Mongo Reader za dobavljanje podataka, te ubacivanje dobijenih podataka u simulaciju preko SCADA Input-a. Slijedi listing 6 koji prikazuje kôd middleware-a.

```
import requests, json  
import typhoon.api.hil as hil  
TEMPERATURE_FIELD = 'AirTemp'  
IRRADIATION_FIELD = 'Ghi'  
CHANNEL_ID = 'e9880be5-44e5-4739-848a-5aee83c99b12'  
THING_KEY = 'ae3fc247-e790-483b-b588-e3abd003fb22'  
MSG_PER_CSV_ROW = 6  
compiled_model_path =  
'/home/filip/Faks/Master/TyphoonHIL/modeli/house'  
Target_files/house.cpd'  
model_loaded = hil.load_model(compiled_model_path,  
vhl_device=True)  
message_row_number = 0  
while True:
```

```

OFFSET = message_row_number * MSGS_PER_CSV_ROW

url =
'http://localhost:8904/channels/'+CHANNEL_ID+'/messages?limit='+str(MSGS_PER_CSV_ROW)+'&offset='+str(OFFSET)
)
headers = {'Authorization': THING_KEY}
req = requests.get(url, headers=headers)
json_response = json.loads(req.text)
if json_response["messages"] == []:
    break

for message in json_response["messages"]:
    if message["subtopic"]==TEMPERATURE_FIELD:
        temperature = message["value"]
    if message["subtopic"]==IRRADIATION_FIELD:
        irradiation = message["value"]

hil.set_scada_input_value("TemperatureInput",
value = temperature)
hil.set_scada_input_value("IrradiationInput",
value = irradiation)
hil.start_simulation(var_sim_define_step =
hil.get_sim_step() * 15)
P_grid_net = hil.read_analog_signal( name =
"P_grid_net")
message_row_number += 1

```

Listing 6. Programski kôd Middleware-a

Prve linije listinga prikazuju komande učitavanje biblioteka za slanje zahtjeva, rad sa JSON-om te Typhoon HIL API-ja. Nakon toga slijede definicije konstanti koje predstavljaju nazive kolona u CSV datoteci, tj. podtemu (eng. „subtopic“) na Mainflux kanalu pod kojom su temperatura i iradijacija upisane u bazu podataka. Konstanta MSGS_PER_CSV_ROW predstavlja broj poruka koje su upisane u bazu za svaki red CSV datoteke, tj. za svaki period od 15 minuta. Sljedeće dvije linije listinga predstavljaju putanju do kompajliranog modela te programsko učitavanje modela. Lokalna promjenljiva `message_row_number` predstavlja redni broj reda CSV datoteke koji čitamo.

Nakon toga slijedi while petlja u kojoj se učitavaju podaci, postavljaju vrijednosti u simulaciju, te preuzima energetska potrošnja. Konstanta OFFSET utiče na to od koje poruke u bazi će početi čitanje, dok se prethodno definisana konstanta MSGS_PER_CSV_ROW koristi kao ograničenje koliko poruka se preuzima sa Mainflux-a u jednoj iteraciji petlje, što je u ovom slučaju 6.

Slijedi definisanje URL-a te formiranje zaglavlja postavljanjem ključa stvari, samog CSV reader-a, za autorizaciju zahtjeva ka Mongo Reader-u. Promjenljiva `req` predstavlja rezultat poziva `get` metode biblioteke `requests` kojoj se proslijede prethodno definisani URL te zaglavlje za autorizaciju.

Iz odgovora se uzima tekstualna reprezentacija i pretvara u JSON pomoću metode `loads`. Nakon linije za uslov prekida while petlje, a to je da smo došli do posljednjeg upisa u bazi, slijedi uzimanje vrijednosti temperature i iradijacije iz poruka pomoću ključa `value`.

Nakon ovoga se, pomoću Typhoon HIL API-ja, dobijene vrijednosti postavljaju u simulaciju koja se pokreće te vraća energetska potrošnja.

5. ZAKLJUČAK

Ovaj rad predstavio je integraciju Mainflux platforme sa Typhoon HIL simulatorom u cilju ubacivanja realnih podataka o temperaturi i solarnoj iradijaciji u simulaciju radi tačnijeg generisanja potrošnje električne energije. Rad doprinosi rješenju šireg opsega koje se bavi pravljenjem i upotrebom softverskih agenata koji se trenutno povezuju na simulirane pametne uređaje, a cilj je da rade i sa realnim, te upravljaju njihovim ponašanjem.

Kako je rješenje predstavljeno u ovom radu dio većeg rješenja, unapređenje bi bilo stvaranje jednostavnog API-ja za pokretanje Middleware-a preko interneta.

Takođe, unapređenje unutar samog sistema bilo bi automatizacija pokretanja svih komponenata. Mogle bi se napraviti dvije Bash skripte, jedna za pokretanje Mainflux-a za komunikaciju sa Middleware-om, druga za pokretanje svega potrebnog za kreiranje CSV reader-a.

6. LITERATURA

- [1] Mainflux platforma, <https://www.mainflux.com/> (pristupljeno u oktobru 2020.)
- [2] Typhoon HIL Control Center, <https://www.typhoon-hil.com/products/hil-software/> (pristupljeno u oktobru 2020.)
- [3] Docker, <https://www.docker.com/> (pristupljeno u oktobru 2020.)
- [4] Mainflux github, <https://github.com/mainflux/mainflux> (pristupljeno u oktobru 2020.)
- [5] Solcast, <https://solcast.com/> (pristupljeno u oktobru 2020.)
- [6] Typhoon HIL API, https://www.typhoon-hil.com/documentation/typhoon-hil-api-documentation/hil_api.html (pristupljeno u oktobru 2020.)
- [7] SenML, <https://tools.ietf.org/html/draft-ietf-core-senml-13> (pristupljeno u oktobru 2020.)
- [8] NATS, <https://nats.io/documentation/> (pristupljeno u oktobru 2020.)
- [9] Docker compose, <https://docs.docker.com/compose/> (pristupljeno u oktobru 2020.)
- [10] Mainflux CLI, <https://mainflux.readthedocs.io/en/latest/cli/> (pristupljeno u oktobru 2020.)
- [11] Bash scripting, <https://help.ubuntu.com/community/Beginners/BashScripting> (pristupljeno u oktobru 2020.)
- [12] Typhoon HIL Control Center, <https://www.typhoon-hil.com/products/hil-software/> (pristupljeno u oktobru 2020.)
- [13] Github Darka Draškovića, <https://github.com/darkodraskovic/mainflux/tree/dbreader/cmd/csv-dbreader> (pristupljeno u oktobru 2020.)

Kratka biografija:

Filip Savić rođen je u Šapcu 1995. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Softversko inženjerstvo i informacione tehnologije odbranio je 2020. god.

kontakt: filipsavic1995@yahoo.com

UPOREDNA ANALIZA METODA PROJEKTOVANJA ASINHRONIH MOTORA COMPARATIVE ANALYSIS OF ASYNCHRONOUS MOTOR DESIGN METHODS

Siniša Arslanović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA

Kratak sadržaj – Asinhroni motori važe za najčešće primenjivane električne mašine industriji. Razvojem industrije javila se potreba za sve preciznijim i izazovnijim tehničkim zahtevima. Cilj ovog istraživanja je poređenje dve metode projektovanja asinhronih motora, analitičke i metode konačnih elemenata. Za poređenje je korišćen osmopolni motor, osne visine 500 mm, snage 630 kW. Za potrebe proračuna korišćeni su softveri KONPRO i ANSYS Maxwell. Na kraju rada prikazano je poređenje karakteristika dobijenih proračunima i ispitnog protokola pomenutog motora.

Ključne reči: Asinhroni motor, metoda konačnih elemenata, uporedna analiza, FEM

Abstract – Asynchronous motors are the most commonly used electrical machines in the industry. With the development of the industry, there was a need for more precise and challenging technical requirements. The aim of this research is to compare two methods of designing asynchronous motors, analytical and finite element method. An eight-pole motor with frame size 500 and a power of 630 kW was used for comparison. KONPRO and ANSYS Maxwell software were used for calculation purposes. At the end of the paper, a comparison of the characteristics obtained by calculations and the test protocol of the mentioned motor is presented.

Keywords: Asynchronous motor, finite element method, comparison, FEM

1. UVOD

Dizajn, izbor materijala, tehnološki postupci i testiranje performansi predstavljaju osnovne korake za proizvodnju motora. Asinhroni motori su mašine koje se najčešće koriste za pogon pumpi, dizalica, ventilatora i sličnih mehanizama [1]. Pored gore navedenih procesa, najbitnija stavka jeste elektromagnetni proračun mašine.

U sklopu elektromagnetnog proračuna dimenzioniše se magnetno kolo mašine, proračunavaju se razni parametri kao što su naponi, struje, fluksevi, gustine flukseva u raznim delovima magnetnog kola i još mnoge druge elektromagnetne veličine koje utiču na rad jednog motora [2]. U ovom radu, elektromagnetnom proračunu se pristupilo na dva načina, analitički i metodom konačnih elemenata.

Analitički pristup podrazumeva proračun karakteristika motora putem analitičkih izraza, polazeći od pretpostavljene geometrije magnetnog kola, nominalne snage motora, nominalnog napona motora i nominalne frekvencije [3].

Metoda konačnih elemenata zasniva se na rešavanju Maksvelovih jednačina elektromagnetnog polja u svakoj tački posmatranog domena. Za potrebe proračuna ovom metodom, formira se model sa pretpostavljenom geometrijom. Za odabrani model formira se mreža konačnih elemenata, uglavnom trougaonog oblika (Slika 2.5). Što je ta mreža gušća, rezultati proračuna su tačniji [4].

2. PRORAČUN KONSTRUKCIJE MOTORA

2.1. Opšti podaci

Tabela 2.1 – Opšti podaci motora

Tip	RZKIET 500 Lk-8
Snaga	630 kW
Napon	Δ 690 V
Broj polova	2p=8
Frekvencija	50 Hz
Tip hlađenja	IC 411
Način ugradnje	IM 1001-B3
Masa	5400 kg
Zaštita	IP 55
Radna mašina	Ventilator

2.2. Stator

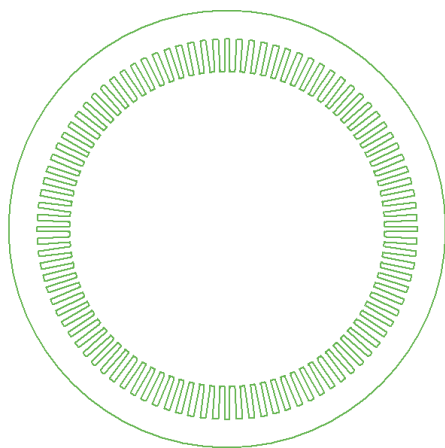
Tabela 2.2 – Podaci statora

Spoljašnji prečnik	900 mm
Unutrašnji prečnik	650 mm
Dužina limpaketa	750 mm
Broj utora	96
Broj navoja	10
Dimenzija žice	9.5x2.5 mm
Korak namotaja	1-10
Br.paralelnih grana	4

Poprečni presek geometrije statora prikazan je na Slici 2.1.

NAPOMENA:

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



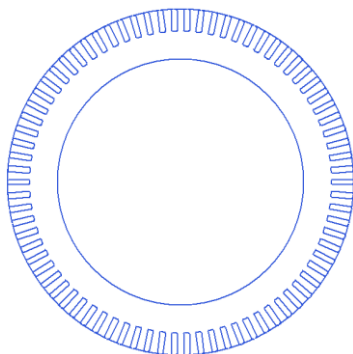
Slika 2.1 – Geometrija statora

2.3. Rotor

Tabela 2.3 – Podaci rotora

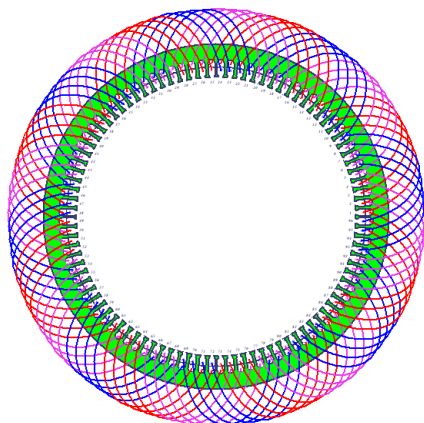
Spoljašnji prečnik	646 mm
Unutrašnji prečnik	460 mm
Dužina limpaketa	750 mm
Broj utora	78
Dimenzija štapa	10x40 mm

Poprečni presek geometrije rotora prikazan je na Slici 2.2.



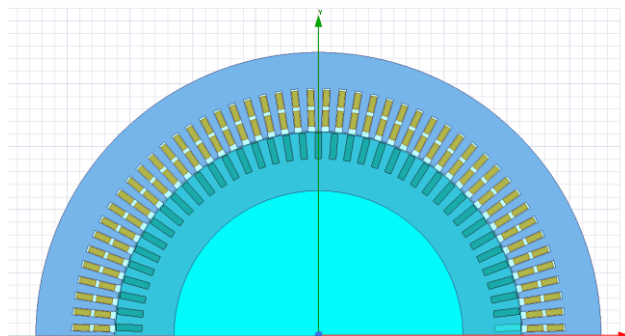
Slika 2.2 – Geometrija rotora

Namotaj statora je dvoslojni uslagani (petljasti), izveden od profilne žice dimenzije 9,5x2,5mm, sa korakom 1-10. Kružna šema statorskog namotaja prikazana je na Slici 2.3.



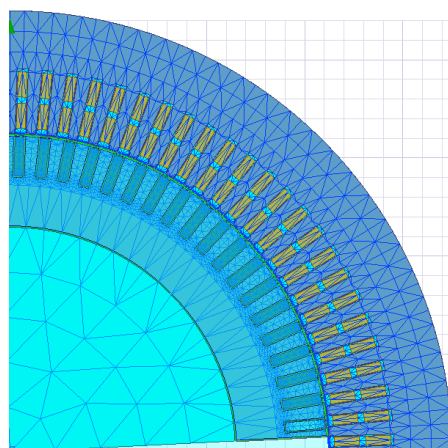
Slika 2.3 – Kružna šema statorskog namotaja

Prethodne slike predstavljaju model motora napravljen u RMXprt, sastavnom delu ANSYS Maxwell softvera. Pomoću njega, pravi se 2D model mašine koji je prikazan na sledećoj slici.



Slika 2.4 – 2-D model posmatranog modela

Nakon uspostavljenog modela i definisanja graničnih vrednosti i izvora polja, formira se mreža konačnih elemenata za koju se u svakoj tački proračunavaju Maksvelove jednačine elektromagnetnog polja.



Slika 2.5 – Mreža konačnih elemenata

2.4 Poređenje rezultata

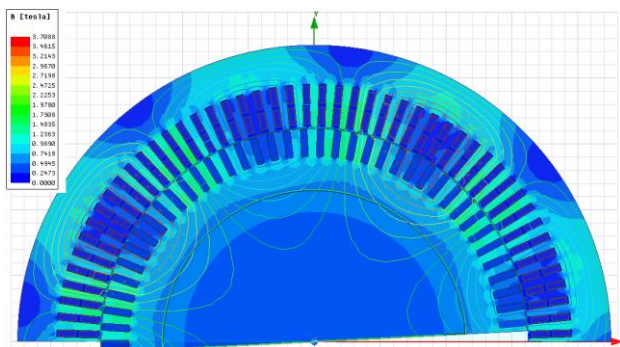
Najpre je urađen proračun analitičkom metodom, nakon koga je usledio precizniji proračun metodom konačnih elemenata.

Nakon uspostavljenog modela i definisanja parametara simulacije dolazi se do traženih karakteristika motora. Do finalnog dizajna mašine dolazi se preko više iteracija koje podrazumevaju prilagođavanje početne geometrije magnetnog kola, parametara namotaja statora kao i ostalih aktivnih delova mašine.

Nakon proračuna sprovedenih putem obe metode projektovanja, prikazuje se poređenje proračuna sa realnom mašinom i njenim ispitnim protokolom. Jasno se primećuju određena odstupanja koja su rezultat kako tehnološkog procesa proizvodnje tako i same konstrukcije motora.

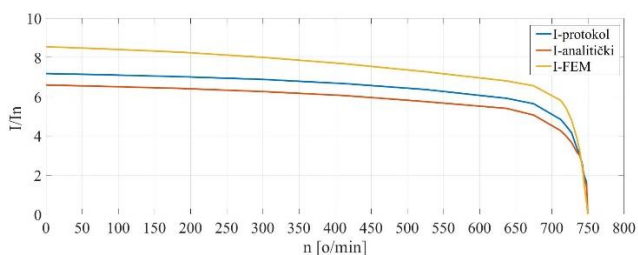
Tabela 2.4 – Poređenje rezultata

	Analitička metoda	FEM	Ispitni protokol
Indukcija u vazдушnom zazoru [T]	0.698978	0.688653	-
Indukcija u zubima statora [T]	~ 1.590	1.30900	-
Indukcija u zubima rotora [T]	~1.469	1.402	-
Indukcija u jarmu statora [T]	1.088	1.002	-
Indukcija u jarmu rotora [T]	1.212	0.211	-
Struja magnećenja [A]	147.9	151.4	-
Reaktansa magnećenja [Ω]	4.52	4.37	-
Otpornost statora [Ω]	0.01055	0.01228	0.01
Otpornost rotora [Ω]	0.01156	0.00744	-
Rasipna induktivnost statora [Ω]	0.17311	0.13221	-
Rasipna induktivnost rotora [Ω]	0.15517	0.16081	-
Gustina struje statora [A/mm ²]	2	2.19	-
Gustina struje u štapu rotora [A/mm ²]	2.22	2.01	-
Gustina struje u kratkospojnom prstenu [A/mm ²]	1.84	1.83	-
Struja praznog hoda [A]	148.01	153.3	147.7
Gubici u gvožđu [W]	7458	4823	16082
Gubici u praznom hodu [W]	8150	5668	20366
Gubici u bakru statora [W]	4365	5077	5239
Gubici u bakru rotora [W]	3712	2355	4294
Gubici na trenje i ventilaciju [W]	3741	3752	5371
Dodatni gubici [W]	3150	6300	3024
Ukupni gubici [W]	22426	22188	34475
Ulazna snaga [W]	652494	658598	655500
Stepen korisnog dejstva [%]	96.56	96.63	94.81
Nominalna struja [A]	643.2	642.8	651.5
Svedena rotorska struja po fazi [A]	327.1	324.8	-
Faktor snage	0.848	0.849	0.853
Brzina obrtanja [o/min]	745	747	744.8
Nominalni moment [Nm]	8087.7	8132.8	8077.7
Maksimalni moment [r.j.]	3.11	3.58	2.83
Polazni moment [r.j.]	1.06	1.26	1.21
Polazna struja [r.j.]	6.68	8.66	7.17

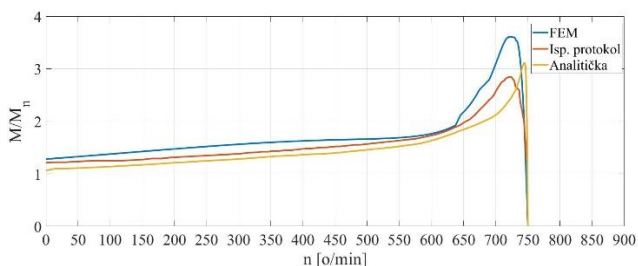


Slika 2.6 – Magnetna slika mašine

Na sledećim slikama prikazano je grafičko poređenje rezultata za struje i momentne karakteristike dobijene na različite načine.



Slika 2.7 – Poređenje struja



Slika 2.8 – Poređenje momentnih karakteristika

3. ZAKLJUČAK

U radu su detaljno opisane i izvedene dve metode projektovanja elektromotora. Analitička metoda, koja je praćena interno razvijenim softverom KONPRO u kompaniji WOLONG SEVER i metoda konačnih elemenata praćena ANSYS Maxwell-om. Obe metode pokazale su veoma dobre i verodostojne rezultate. Kada se uporede, vidimo da metoda konačnih elemenata daje mnogo preciznije rezultate kao i mnogo više mogućnosti za uspostavljanje raznih uslova simulacije i praćenja rezultata.

Poređenjem rezultata ove dve metode sa rezultatima ispitnog protokola može se videti da finalni proizvod, osim same konstrukcije i dizajna zavisi i od tehnoloških procesa, kvaliteta materijala sa kojim se radi, stručnosti radne snage kao i iskustva projektanta. Značajnije odstupanje simulacije i proračuna javlja se u gubicima u gvožđu.

Njihov iznos je kod realne mašine veći od računatih vrednosti. Ovo ukazuje na to da pored simulacije i proračuna uvek treba voditi računa i o ostalim faktorima. Daljeg značajnog odstupanja nema, što potvrđuje da je i jedna i druga metoda uspešno implementirana.

Ovako konstruisan i ispitan motor uspešno je isporučen krajnjem kupcu, uz dodatna poboljšanja i rekonstrukcije. Motor se koristi za pogon ventilatora u cevi.

4. LITERATURA

- [1] Berislav Jurković: „Elektromotorni pogoni“ školska knjiga-Zagreb, 1978.
- [2] Ion Boldea, Syed A. Nasar : „The induction machines design handbook, second edition“, ISBN 978-1-4200-6668-5, CRC Press, Taylor & Francis grup, New York, 2010.
- [3] Jožef Varga : “ Analitička metoda za sračunavanje karakteristika trofaznih asinhronih motora” D.K.: 621.313.33.025.3==861
- [4] S. J. Salon, “Introduction to Finite Elements,” in *Finite Element Analysis of Electrical Machines*, Springer US, 1995, pp. 1–16.

Kratka biografija:



Siniša Arslanović rođen je u Subotici, 1993. god. Diplomski rad iz oblasti Električnih mašina odbranio je 2018. godine u Novom Sadu na Fakultetu tehničkih nauka. Iste godine upisao je Master akademske studije. Zaposlen je kao projektant električnih mašina u kompaniji WOLONG SEVER. kontakt: sinisaarslanovic@gmail.com

UTICAJ RAZVOJA VOĐENOG TESTOVIMA NA DIZAJN SOFTVERA IMPACT OF TEST DRIVEN DEVELOPMENT ON SOFTWARE DESIGN

Aleksandar Kahriman, *Fakultet tehničkih nauka, Novi Sad*

Oblast – SOFTVERSKO INŽENJERSTVO

Kratak sadržaj - U ovom radu su predloženi principi razvoja vođenog testovima čija primena vodi do fluidnijeg razvoja. Dodatno je ustanovljena veza između fluidnosti razvoja i potencijala za evoluiranje aplikativnog dizajna softvera. Proaktivnim pristupom evoluiranja modula se iterativnim postupkom dolazi do boljeg aplikativnog dizajna.

Ključne reči: razvoj vođen testovima, TDD, dizajn softvera, arhitektura softvera

Abstract – This paper offers principals of test driven development, applying which leads to more fluid development. Additionally, a link between fluid development and the potential to evolve application design is established. Proactive approach to evolving modules in iterations results in better application design

Keywords: test driven development, TDD, software architecture, software design

1. UVOD

Verovatno većim ubrzanjem nego bilo koja druga, softverska industrija evoluirala metodologije i alate pomoću kojih teži da zadovolji apetite čovečanstva za revolucionarnom tehnologijom. Softverska infrastruktura, koja se postepeno razvijala, je danas najuticajniji katalizator rasta industrije. Ona omogućava razvoj izuzetno složenih rešenja za frakciju vremena koje je bilo potrebno u prošlosti. Mnoge kompanije postaju lideri na tržištima u kojima posluju upravo zahvaljujući ekonomičnoj optimizaciji i automatizaciji procesa koju softverska infrastruktura omogućava. Upravo ova duboka integrisanost u svakodnevni rad mnogih industrija i živote individua stvara veliki pritisak na softversku industriju da isporučuje kvalitetan softver u što kraćim mogućim vremenskim rokovima.

Značajan doprinos softverskoj infrastrukturi čine metodologije čije praktikovanje dovodi do većeg stepena optimizacije razvoja softvera. Jedna takva metodologija je razvoj vođen testovima (engl. *Test Driven Development*, TDD), koja se pojavila u industriji početkom 21. veka. TDD je ciklična metodologija razvoja softvera u kojoj svaki ciklus započinje pisanjem automatskog testa. Dodatno, tokom svakog ciklusa vrši se neophodna implementacija kako bi se test uspešno izvršio. Ciklus se završava restrukturiranjem koda u cilju povećanja kvaliteta. Proaktivnim pristupom, TDD metodologija smanjuje broj grešaka u razvoju i pruža sigurnost tokom evoluiranja

softvera [1-3]. Uspešno praktikovanje TDD metodologije je veoma izazovno iz razloga što zahteva pragmatičnost, kao i dugogodišnje iskustvo inženjera u aplikativnom dizajnu. Stoga, neophodno je osmisлити dizajn strategije kako bi razvoj softvera bio efektivniji. U radu su predloženi principi, koji su oformljeni unapređivanjem ovih strategija. Ovi principi omogućavaju fluidnije TDD cikluse, koji dovode do boljeg aplikativnog dizajna i umanjuju negativne posledice praktikovanja metodologije.

2. TEORIJSKI OKVIR

Softverska industrija je davno prepoznala implikacije grešaka u razvoju i cenu ispravki istih u različitim periodima životnog ciklusa softvera. U cilju amortizacije ovog problema uvedene su mere verifikacije ispravnog ponašanja softvera. Inicijalna strategija se oslanjala isključivo na manuelnu verifikaciju, ispoljavajući negativne karakteristike pri skaliranju. Naglim razvojem softverske infrastrukture automatsko testiranje doprinosi skalabilnosti verifikacije u razvoju softvera. Izvršavanje velikog broja automatskih testova koji verodostojno analiziraju integritet softvera za relativno kratak vremenski period uliva samopouzdanje u proces razvoja koje se direktno preslikava u rast kvaliteta.

2.1. Razvoj vođen testovima

Postepenim uvođenjem preporučenih praksi za pisanje kvalitetnih automatskih testova ispraćeno evolucijom tehnologije i pratećih alata postalo je moguće izvršavati veliki broj testova za relativno kratko vreme. Ovaj napredak u automatskom testiranju je doveo do novih metodologija razvoja softvera, među kojima je TDD. Najvažnija novina koju TDD potencira je pisanje automatskih testova neposredno uoči aplikativnog koda čije ponašanje verifikuju. Automatski test postaje forma specifikacije za mali deo funkcionalnosti koji je potrebno implementirati. Pošto aplikativni kod u trenutku pisanja testa još ne postoji, test se inicijalno neuspešno izvršava, označavajući crvenu fazu TDD ciklusa. Dalje se prelazi u zelenu fazu ciklusa, u kojoj se piše minimalni aplikativni kod kako bi se test uspešno izvršio. Posle zelene faze nastupa opcionalna faza restrukturiranja u kojoj se aplikativni kod evoluirala pod sigurnosnom mrežom automatskih testova. Kada inženjer iskustveno odluči da dalje poboljšanje aplikativnog koda nije pragmatično, i kada se svi testovi uspešno izvršavaju, prelazi se u novi TDD ciklus, tj. u crvenu fazu.

Prilikom TDD ciklusa konstantnim pokretanjem automatskih testova se dobijaju blagovremene povratne informacije o integritetu softvera. Na ovaj način, održavanje kvaliteta aplikativnog koda i smanjenje tehničkog duga prelaze iz reaktivnih u proaktivne radnje rezultujući

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Goran Savić.

predvidivim razvojem, manjim brojem grešaka, usporavanjem degradacije kvaliteta softvera vremenom i smanjenjem stresa razvojnog tima [1-6].

2.2. Dizajn softvera

Dizajn (arhitektura) softvera predstavlja skup značajnih odluka koje oblikuju softverski sistem, gde se značaj meri cenom promene. Ove odluke se konstantno donose na raznim nivoima granularnosti u opsegu od jedne funkcionalnosti do celog sistema. Cilj softverskog dizajna je optimizacija količine resursa koji su potrebni da se razvije i održava softverski sistem [7].

Rastom kompleksnosti sistema, do te mere da je standardno da se prosečan softver sastoji iz stotina hiljada linija aplikativnog koda, došlo je do ozbiljnih izazova, posebno u segmentima ispravnosti i proširivosti. Naravno, industrija je vremenom učila iz svojih poteškoća, uočavajući apstraktne šablone koji vode do smanjenja entropije i povećanja kvaliteta u razvoju softvera. Prethodno spomenuti šabloni su diskutovani aktivno u okviru zajednice softverskih inženjera tokom poslednjih par decenija. Dobra sinteza ovih napora je skup principa poznat kao SOLID akronim [7]. Ovi principi postavljaju apstraktne ciljeve ka kojima treba težiti kako bi se došlo do boljeg aplikativnog dizajna. Jedini način za poboljšanje dizajna je analiza svake odluke u kontekstu SOLID i drugih principa dizajniranja softvera. TDD navodi na raspoređivanje problema na manje delove. Prolazeći kroz TDD cikluse na većem nivou granularnosti inženjer dobija blagovremene detaljne povratne informacije o kompleksnosti dizajna i ispravnosti rešenja. Dakle, usporavajući tempo razvoja i povećavajući granularnost analize problema TDD na proaktivan način povećava verovatnoću donošenja boljih dizajn odluka.

3. PREDLOG PRINCIPA EFEKTIVNOG PRAKTIKOVANJA TDD METODOLOGIJE

Aplikativni dizajn i TDD imaju konstraintuitivan odnos. Naime, dobar dizajn vodi do fluidnijih TDD ciklusa koji omogućavaju bolji uvidaj u celokupnu sliku tehničkih i domenskih suptilnosti problema. Blagovremeno prepoznavanje ovih suptilnosti vodi do boljeg dizajna. Aktivan rad na poboljšanju aplikativnog dizajna i praktikovanje TDD metodologije čine sinergiju koja ima potencijal povećanja kvaliteta softvera. Ukoliko TDD ciklusi nisu fluidni, gube svoju potentnost u produkovanju dobrog dizajna. Značajna činjenica je da TDD ne garantuje dobar dizajn, već forsira blagovremenu evaluaciju dizajna, koja vodi do fluidnosti pri razvoju. U daljem radu su identifikovani i predloženi principi koji omogućavajući fluidnije TDD cikluse vode do boljeg aplikativnog dizajna i obrnuto.

3.1. Princip konstrukcije grafa zavisnosti

Upotrebom klasične metode instanciranja objekata u testovima dolazi do konflikta između težnje ka evoluciji modula i zadržavanja osećaja sigurnosti ispravnog ponašanja postojećih funkcionalnosti. Uvođenjem novih klasa konstruktori koji se koriste za instanciranje objekata bivaju narušeni pa samim time testovi koji ih koriste prijavljuju greške tokom kompajliranja. Da bi testovi mogli da se kompajliraju potrebno je napraviti izmene u njima. Ako se nakon tih izmena neki testovi neuspešno izvršavaju, postavlja se pitanje da li se neuspešno izvrša-

vaju zato što je napravljena greška prilikom faze restrukturiranja ili prilikom izmena pri instanciranju objekata u testovima. Svaki put prilikom ovakve evolucije inženjer mora da prilagodi instanciranje objekata u potencijalno velikom broju testova. Neophodan napor i mogućnost nastanka dileme obeshrabruje pokušaje evoluiranja modula. Ukoliko modul ne evoluiira, dolazi do gubitka jasnoće, jednostavnosti i narušavanja principa dobrog dizajna (SOLID i drugih). Moguće je umanjiti ovaj problem delegiranjem instanciranja objekata specijalizovanom modulu koji se naziva kontejner za injekciju zavisnosti. DI kontejner (engl. *dependency injection (DI) container*) rekurzivno konstruiše objekte zajedno sa objektima od kojih zavise i omogućava konfigurabilnost implementacionih detalja.

Delegiranjem kreiranja objekta klase preko čijeg javnog interfejsa se vrši testiranje modula se drastično smanjuje mogućnost neuspešnog izvršavanja testova usled izmena u grafu zavisnosti. Ukoliko se pri inicijalizaciji testa iskoriste iste konfiguracije DI kontejnera kao u aplikaciji verodostojnost istog raste jer bolje opisuje realnu upotrebu modula. TDD je izuzetno tesno uvezan sa osećajem samopouzdanja do mere da je ono apsolutno neophodno za fluidan razvoj. Iz potrebe smanjenja pojava lažno neuspešno izvršavajućih testova koji degradiraju samopouzdanje tokom razvoja proističe sledeći princip:

Korišćenje isključivo DI kontejnera za konstruisanje objekata pri testiranju vodi do smanjenja lažno neuspešno izvršenih testova, pojačavajući osećaj sigurnosti koji testovi omogućavaju tokom TDD ciklusa.

3.2. Princip kontinualne evolucije kolekcije testova

Inicijalni testovi koji se napišu tokom implementacije funkcionalnosti ne moraju biti konačni. Oni često nisu ni ispravni dugoročno posmatrano, a i nije nužno da budu. Njihov zadatak je da obezbede referentne tačke sigurnosti u toku iterativne evolucije aplikativnog koda. Samim time, postoji momenat kada oni bivaju uklonjeni ili evoluirani u testove većeg opsega ili striktnijih kriterijuma verifikacije.

Striktno praćenje pravila TDD metodologije dovodi praktikante do konstatnih zastoja, jer impliciraju da neki deo ponašanja mora biti unapred poznat u potpunosti pre pisanja testa u crvenoj fazi. Ovo često za posledicu ima pojavu koja je poznata pod nazivom paraliza analizom (engl. *analysis paralysis*). Inženjer troši dragoceno vreme da osmisli sve detalje, a za uzvrat često ne dolazi do dugoročno tačnih informacija. Kako ne bi došlo do ovoga, prvo treba prihvatiti da nijedan kvalitetno napisan test nije beznačajan, bez obzira na dugoročnu vrednost uslova verifikacije. Ideja je početi od onoga što je trenutno poznato. Provera da rezultat neke metode koja enkriptuje ulazni tekst nije prazan tekst je dobar primer. Nakon par TDD ciklusa ovaj i još par sličnih testova će se uspešno izvršiti pri čemu ih treba prepraviti uvođenjem striktnijih uslova verifikacije ili ih ukloniti. Nakon nekog broja ciklusa deo funkcionalnosti biva implementiran do te mere da pouzdano radi. Ovo i dalje nije poželjno stanje kolekcije testova. Neminovno postoji dosta testova nad manjim opsegom koji potvrđuju ista ponašanja kao i testovi nad većim opsegom, samo sa verodostojnijim kriterijumom verifikacije. Stoga, deo testova treba evoluirati u testove većeg opsega, a deo ukloniti, kako bi se

omogućilo samopouzdanje i manervabilnost za evoluiranje dizajna bez ograničenja koje uvode testovi nad manjim opsegom. Odnosno, pojedinačne testove treba tretirati kao večno privremene u cilju evolucije aplikativnog dizajna i kvaliteta cele kolekcije testova. U razvijanju softvera je jedina konstanta promena, pa odatle proizilazi sledeći princip:

Kontinualnom evolucijom kolekcije automatskih testova prilikom TDD ciklusa, vodeći se kriterijumom težnje ka verifikaciji nad većim opsegom, se omogućava evolucija aplikativnog dizajna nad većim opsegom.

3.3. Princip poreza na nasleđivanje

Objektno orijentisana paradigma definiše četiri glavna principa koji služe kao gradivni elementi kvalitetnog aplikativnog dizajna: apstrakcija, enkapsulacija, nasleđivanje i polimorfizam. Od ovih principa nasleđivanje je verovatno u najširoj upotrebi i ujedno najmanje shvaćeno [8]. U velikom broju slučajeva inženjeri koriste nasleđivanje u svrhu ponovne upotrebe koda. Pri tome podklase vremenom moraju da obezbede funkcionalnosti koje zahtevaju promene u ponašanju implementiranom u nadklasi. Promene u nadklasi se veoma često reflektuju na ostale podklase uzrokujući nepredvidive posledice. Pored toga nadklasa postaje kompleksnija vremenom, i samim time, skuplja za održavanje. Zloupotreba nasleđivanja na ovaj način narušava princip otvorenosti-zatvorenosti, kao i princip Liskov substitucije [7]. Na manje apstraktnom nivou važi da od 11 dizajn šablona ponašanja opšte poznatih u softverskoj industriji samo jedan (*template method*) koristi nasleđivanje kao glavni gradivni element postizanja željenog cilja [9]. Posmatrajući sve tipove veza u aplikativnom kodu nasleđivanje je ujedno najjača i najrigidnija.

Ovu grešku TDD adresira na proaktivan način. Naime, za svaku podklasu je potrebno napisati testove koji prolaze i kroz ponašanje nadklase. Više koda u nadklasi vodi do veće replikacije testova, što vodi do veće potrošnje resursa radi postizanja cilja visoke pokrivenosti. TDD na ovaj način efektivno, po svom dizajnu, uvodi porez na zloupotrebu nasleđivanja koji se oseti odmah. Ovo je velika prednost nad kasnijim pisanjem testova, jer podstiče tim da evaluiira dizajn odluke pre nego što šteta postane velika. Posledice grešaka u dizajnu softvera se obično otkrivaju kada prođe znatno vreme od momenta donošenja istih, pa je ovaj blagovremeni ishod izuzetno poželjan i iz njega proizilazi sledeći princip:

Visok ciljani stepen pokrivenosti aplikativnog koda automatskim testovima u kontekstu TDD-a vodi do smanjenja štetne upotrebe nasleđivanja uvođenjem pragmatične potrebe za blagovremenom evaluacijom kvaliteta odluke upotrebe istog.

3.4. Princip određivanja opsega verifikacije jediničnih testova

Jedinični testovi verifikuju male segmente koda. Ne postoji opšte pravilo koje može da dovede do izbora opsega verifikacije koji neće ispoljiti nijednu negativnu posledicu. Odabir opsega verifikacije jediničnog testa je proces određivanja najoptimalnijeg za dati aplikativni kod u kontekstu potreba zainteresovanih strana. Intuitivno je da bi grupa klasa povezana kompozicijom predstavljala dobar opseg verifikacije jediničnog testa, dok bi klase povezane agregacijom bile dobri kandidati za imitiranje

pri verifikaciji. Klase povezane kompozicijom trebaju po dizajnu da rešavaju konkretan problem na istom nivou apstrakcije, da nemaju veliku kompleksnost, da se sastoje iz relativno malo koda i da se ne oslanjaju na nestabilne konkretizacije. U TDD-u testovi prednjače dizajnu i implementaciji na ovom nivou granularnosti. Stoga, problem se svodi na identifikaciju vrste asocijacije u toku TDD ciklusa. Ukoliko međusobnom asocijacijom dve klase narušavaju princip jedne odgovornosti onda je ta asocijacija agregacija. Primenjujući princip jedne odgovornosti kao kriterijum za blagovremenu eliminaciju klasa povezanih agregacijom oblikuje se opseg verifikacije. Primenom ove strategije aktivno tokom TDD ciklusa smanjuje se vreme donošenja odluka kao što su gde smestiti i kako uvezati novo ponašanje sa ishodom poboljšanog dizajna, iz čega proističe sledeći princip:

Primenom principa jedne odgovornosti aktivno tokom TDD razvoja moguće je blagovremeno identifikovati segmente koda koji ne pripadaju opsegu verifikacije jediničnog testa, vodeći do jasno definisanog opsega verifikacije i boljeg dizajna.

3.5. Princip balansa verodostojnosti i uloženog napora

Dosadašnja istraživanja su pokazala da je razvoj praktikovanjem TDD-a čak do 35% sporiji [10, 11]. Pitanje se postavlja šta praktikant TDD metodologije može da uradi kako bi smanjio dodatan napor pri razvoju softvera. Strategija za optimizaciju uloženog napora sledi iz razumevanja doktrina dveju TDD škola. Starija je Detroit škola, čija je doktrina poznata po pristupu iz srži problema ka spoljašnosti. Primenom ove doktrine bi praktikant krenuo da razvija softver od internih do integracionih modula, od domenskih do ulazno izlaznih. Novija je London škola, čija je doktrina poznata po pristupu sa periferija problema ka srži. Primenom ove doktrine bi praktikant krenuo da razvija softver od integracionih do internih modula, od ulazno izlaznih do domenskih.

Dobra početna strategija je krenuti sa periferijom ukoliko je u pitanju razvoj složenog softvera. Na ovaj način je inicijalna razlika napora pri praktikovanju TDD-a relativno mala. Ovo važi zato što se inicijalno pišu testovi koji verifikuju funkcionalnost dosta oskudno i čija je primarna svrha da otkriju da li je ideja za rešavanje problema obećavajuća ili ne. Doktrina London škole uvođenjem imitacija na više nivoa modulaziracije omogućava upravo to jer se korišćenjem savremenih alata imitacije mogu brzo implementirati.

Ukoliko je ideja obećavajuća, može se preći na sledeći nivo (sledeći modul) gde opet treba evaluirati opštu ideju bez trošenja previše vremena. Ovaj proces se ponavlja sve do inicijalne implementacije funkcionalnosti sa imitacijama na određenim mestima. Potom treba implementirati detalje koji su ostali imitirani. Značajno je napomenuti da u ovom momentu većina slučajeva izvršavanja toka funkcionalnosti nisu pokriveni ovim testovima, tj. da je verodostojnost testova veoma niska.

U daljoj implementaciji je potrebno pooštriti kriterijume verifikacije i evoluirati kolekciju testova nastalu u inicijalnoj fazi. Ukoliko su ti detalji deo integracionih modula, preferabilno je nastaviti implementaciju primenom doktrine London škole. Ukoliko su pak deo internih modula koji su sami po sebi složeni i u idealnom

slučaju ne zavise od drugih modula, preferabilno je primeniti doktrinu Detroit škole.

Primena TDD-a sa visokim ciljanim stepenom verodostojnosti je preferabilna ukoliko je poznato da je problem koji se rešava složen. Naravno, nisu svi problemi složeni i nekada je teško unapred proceniti koji jesu. Iz potrebe da se optimizuje napor uložen pri praktikovanju TDD metodologija proističe sledeći princip:

Pri razvoju vođenom testovima treba dovoljno često evaluirati složenost problema koji se rešava kako bi se donela odluka koja doktrina TDD-a je preferabilnija za rešavanje istog.

4. ZAKLJUČAK

Uticaj razvoja vođenog testovima na dizajn softvera je neosporiv samom činjenicom da podstiče inženjera da blagovremeno evaluira dizajn odluke. Dizajn softvera se idejno osmišljava uoči razvoja i iterativno poboljšava tokom životnog ciklusa softvera. Dakle, realizacija boljeg dizajna je moguća isključivo kroz mogućnost i ekonomičnost konstantne evolucije istog.

Principi predloženi u ovom radu predstavljaju dobre smernice koje teže da omoguće i olakšaju ekonomičnu evoluciju dizajna softvera, nadograđujući se na proces i beneficije TDD metodologije.

5. LITERATURA

- [1] K. Beck, *Test-driven development: by example*. Boston: Addison-Wesley, 2003.
- [2] L. Koskela, *Test driven: TDD and Acceptance TDD for Java developers*. Greenwich, CT: Manning, 2008.
- [3] R. C. Martin, *The clean coder: a code of conduct for professional programmers*. Upper Saddle River, N.J.: Prentice Hall, 2011.
- [4] B. George and L. Williams, "An initial investigation of test driven development in industry," in *Proceedings of the 2003 ACM symposium on Applied computing - SAC '03*, Melbourne, Florida, 2003, p. 1135, doi: 10.1145/952532.952753.
- [5] N. Nagappan, E. M. Maximilien, T. Bhat, and L. Williams, "Realizing quality improvement through test driven development: results and experiences of four industrial teams," *Empir. Softw. Eng.*, vol. 13, no. 3, pp. 289–302, Jun. 2008, doi: 10.1007/s10664-008-9062-z.
- [6] E. M. Maximilien and L. Williams, "Assessing test-driven development at IBM," in *25th International Conference on Software Engineering, 2003. Proceedings.*, Portland, OR, USA, 2003, pp. 564–569, doi: 10.1109/ICSE.2003.1201238.
- [7] R. C. Martin, *Clean architecture: a craftsman's guide to software structure and design*. 2018.
- [8] E. Tempero, J. Noble, and H. Melton, "How Do Java Programs Use Inheritance? An Empirical Study of Inheritance in Java Software," in *ECOOP 2008 – Object-Oriented Programming*, vol. 5142, J. Vitek, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 667–691.
- [9] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design patterns: Elements of reusable object-oriented software*. England: Pearson education Limited, 1995.
- [10] T. Bhat and N. Nagappan, "Evaluating the efficacy of test-driven development: industrial case studies," in *Proceedings of the 2006 ACM/IEEE international symposium on International symposium on empirical software engineering - ISESE '06*, Rio de Janeiro, Brazil, 2006, p. 356, doi: 10.1145/1159733.1159787.
- [11] L. Huang and M. Holcombe, "Empirical investigation towards the effectiveness of Test First programming," *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 182–194, Jan. 2009, doi: 10.1016/j.infsof.2008.03.007.

Kratka biografija:



Aleksandar Kahrman rođen je u Novom Sadu 1994. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Računarstvo i automatika – Softversko inženjerstvo odbranio je 2017. god. Interesuju ga pragmatične strategije postizanja većeg nivoa kvaliteta softverskih sistema. Kontakt: aleksandar.kahrman@gmail.com

VEB PLATFORMA ZA TESTIRANJE ZNANJA SA PRAĆENJEM POKRETA OKA WEB PLATFORM FOR KNOWLEDGE TESTING WITH EYE-TRACKING

Ognjen Vlajić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je opisana veb platforma za online testiranje čiji glavni zadatak jeste da uz pomoć uređaja za praćenje pokreta oka prikupi podatke o ispitaniku. Predstavljena je tehnologija za praćenje pokreta oka. Data je specifikacija i objašnjenje dela implementacije platforme.

Ključne reči: Praćenje pokreta oka, online testiranje, metrike, tačke pogleda, fiksacija, regija od interesa

Abstract – This paper describes a web platform for online testing whose main purpose is collecting eye-tracking data about students. The eye-tracking technology is presented. The specification and implementation of the platform are given.

Keywords: Eye-tracking, online testing, metrics, gaze point, fixation, area of interest

1. UVOD

Razvoj tehnologije i revolucija Interneta utiču na sve sfere života, pa tako i na obrazovanje i dovode do toga da se tradicionalno testiranje na papiru (pen-and-pencil testing, paper-based testing - PBT) polako zamenjuje testiranjem na računaru (computer-based testing - CBT).

Testiranje na računaru započeto je pre skoro jednog veka kada je izvršeno testiranje na IBM računaru [1] i od tada se značajno promenilo. Prvo je vršeno testiranje na samo jednom računaru, potom testiranju u okviru lokalne računarske mreže, da bi se danas došlo do online testiranja (web-based testing - WBT).

Online testiranje ima mnoge prednosti u odnosu na testiranje na papiru, a neke od njih su: upotreba svih vrsta multimedijalnog sadržaja u okviru testa, uklanjanje prostornog ograničenja i mogućnost da se test radi sa bilo kog mesta gde ima Interneta, mogućnost da se automatski dobiju rezultati testa.

Ipak, pored svih ovih prednosti online testiranje ima i jednu manu, a to je da veliki broj profesora ima problema sa sastavljanjem testa.

Kako bi im se ovaj posao donekle olakšao u ovom radu će biti predstavljena tehnologija za praćenje pokreta oka i platforma za online testiranje bazirana na njoj. Glavni zadatak ove platforme će biti da profesorima uz odgovarajuću interpretaciju da podatke o tome kako studenti razmišljaju i kako se ponašaju u toku testa.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Goran Savić, vanr.prof.

2. TEHNOLOGIJA ZA PRAĆENJE POKRETA OKA

Praćenje pokreta oka (eye-tracking, gaze interaction) je tehnologija koja se koristi kako bi se procenilo gde i koliko dugo neko gleda na stimulans dok obavlja neki zadatak [2]. Tehnologija za praćenje pokreta oka je našla svoju primenu u mnogim oblastima kao što su: medicina, marketing, industrija. Pored korišćenja u istraživačke svrhe tehnologija se može iskoristiti i kao pomoćni alat kod osoba sa invaliditetom kako bi im se omogućilo upravljanje sadržajem na ekranu.

2.1. Tipovi uređaja za praćenje pokreta oka

Postoje četiri osnovna tipa uređaja za praćenje pokreta oka:

- screen-based (remote, desktop)
- glasses (mobile)
- uređaj ugrađen unutar VR (virtual reality) opreme
- webcam-based

Poslednji navedeni tip (webcam-based uređaj) tehnološki zaostaje za ostalima iz razloga što ne koristi infrared tehnologiju.

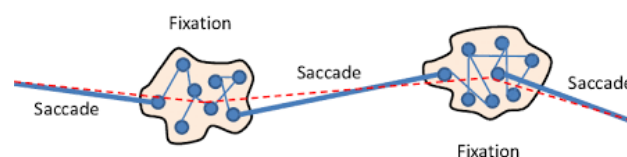
2.2. Osnovni pojmovi

U ovom delu rada će biti predstavljeni osnovni pojmovi vezani za pokrete oka i izlaze koji se dobijaju sa uređaja.

Tačka pogleda (gaze point, point of gaze, point of regard) je jedan od osnovnih izlaza koji se dobijaju sa uređaja i predstavlja tačku na ekranu u koju se gleda. Broj tačaka pogleda koji se može dobiti u jednoj sekundi od uređaja zavisi od frekvencije uzrokovanja uređaja, pa tako ukoliko uređaj ima frekvenciju od 60 Hz dobijamo 60 tačaka pogleda u sekundi.

Fiksacija (fixation) predstavlja uspostavljanje i zadržavanje pogleda na vizuelnom cilju tj. fokusiranje pogleda na neki objekat u određenom vremenskom periodu (200-300 ms). Sa stanovišta uređaja fiksacija predstavlja klaster koji formiraju tačke pogleda koje su veoma blizu u vremenu i prostoru kao što se može videti na slici 1. Fiksacija je idealna mera za vizuelnu pažnju.

Saccade predstavlja brzo pomeranje pogleda sa jedne fiksacije na drugu (40-50 ms).

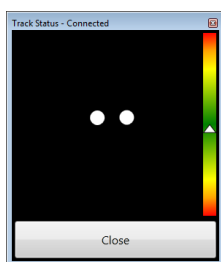


Slika 1. Tačke pogleda, fiksacije (klaster tačaka) i saccade

2.3 Priprema uređaja za upotrebu

Od uređaja za praćenje pokreta oka želimo da dobijemo što preciznije podatke. Kako bi se ovo postiglo pre prikupljanja podataka sa uređaja potrebno je proći kroz dva koraka:

1) Pozicioniranje uređaja na optimalnu distancu i ugao u odnosu na lice korisnika - Ovo se postiže uz pomoć softvera koji dolaze sa uređajima. Softver ukazuje na to da li je uređaj dobro pozicioniran pomoću *track status*-a. *Track status* se sastoji od linije sa indikatorom pozicije koji mora da bude što bliže sredini linije. Primer *track status*-a se može videti na slici 2.



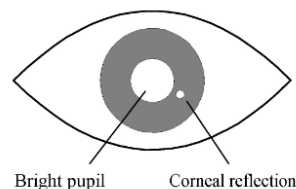
Slika 2. *Track status*

2) Prolaženje kroz proces kalibracije - Kalibracija je proces u toku kog uređaj meri kako oči reflektuju svetlost odnosno procenjuje gde se nalazi tačka posmatranja. Ona se vrši tako što se od korisnika traži da prati tačke ili druge grafičke elemente na ekranu kao što je prikazano na slici 3. Dok korisnik prati grafičke elemente uređaj uz pomoć LED kamera šalje infracrvene zrake ka oku kako bi stvorio jaku refleksiju koja će omogućiti lakše praćenje oka. Refleksija se stvara tako što infracrveni zrak prolazi kroz mrežnjaču i veliki deo se reflektuje nazad omogućavajući da se zenica vidi kao svetla tačka (bright pupil effect), nešto slično poput efekta crvene zenice prilikom slikanja foto aparatom.

Pored ovoga infracrveni zrak takođe stvara refleksiju na rožnjači koja se pojavljuje kao mali odsjaj kao što se može videti na slici 4. Kamera uređaja prikuplja informacije o refleksijama oka, one se kombinuju sa 3D modelom oka koji uređaj poseduje i uz pomoć određenih kalkulacija i filtriranja pronalazi se tačka posmatranja. Prikupljanje opisanih refleksija je u nauci poznato kao "corneal-reflection/pupil-centre" metoda [3]. Određeni uređaji imaju podršku i za efekat tamne zenice (dark pupil effect) pošto se ona kod nekih korisnika pokazala kao bolje rešenje.



Slika 3. Prikaz grafičkih elemenata koji se koriste prilikom kalibracije (od korisnika se očekuje da zadrži pogled unutar krugova)



Slika 4. Efekat svetle zenice i refleksija na rožnjači

2.4 Metrike

Metrika predstavlja alat koji se koristi nad određenom grupom podataka (izlaza) koji su dobijeni od uređaja za praćenje pokreta oka sa ciljem da se da uvid u to kako se korisnik ponašao i kako je razmišljao u određenim situacijama. Postoje različite metrike, neke u literaturi imaju i više imena [4], a ovde će biti navedene samo neke:

1) *Area of interest* (region of interest) omogućava da se označe regije stimulansa koji se prikazuje korisniku. Mogu biti različitih oblika, ali su najčešće u obliku pravougaonika ili elipse. Koristi se za izvođenje drugih metrika.

2) *Time to first fixation* ukazuje na to koliko ukupno vremena treba ispitaniku da pogleda u određenu regiju od interesa od početka prikaza stimulansa. Ukoliko postoji više ispitanika ova metrika može da podrazumeva prosečno vreme koje je potrebno svim ispitanicima da pogledaju regiju.

3) *Time spent* (fixation time, dwell time) predstavlja ukupno vreme koje je ispitanik proveo gledajući u određenu regiju od interesa.

4) *Ratio* metrika pruža informaciju o tome koliko ispitanika je zaista pogledalo u specifičnu regiju. Može se koristiti isključivo kada postoji veći broj ispitanika.

5) *Fixation sequences* se bazira na prostornim i vremenskim informacijama tj. gde i kada je ispitanik gledao. Sekvenca se može sastojati samo od fiksacija, a može i od fiksacija i *saccade* (u ovom slučaju se u literature može naći i pod nazivom *scanpath*).

6) *Revisits* predstavlja informaciju o tome koliko puta je ispitanik vratio pogled na određenu regiju od interesa.

7) *First fixation duration* pruža informaciju o tome koliko dugo je trajala prva fiksacija u okviru neke regije.

8) *Average fixation duration* predstavlja informaciju o tome koliko se u proseku ispitanik zadržavao na nekoj regiji od interesa. Ova informacija se može izvući za pojedinca ili grupu.

9) *Heatmap* predstavlja prikaz različitih gradijenata boja nad slikom. U slučaju praćenja pokreta oka *heatmap* predstavlja vizualizaciju distribucije tačaka pogleda na stimulansu. Uglavnom se koriste tri boje (crvena, zuta i zelena) za prikaz količine tačaka pogleda koje su usmerene ka određenom delu stimulansa.

10) *Opacity map* predstavlja vizuelnu reprezentaciju podataka uređaja za praćenje pogleda. Ova mapa je suprotnost *heatmap*-i. Na ovoj mapi delovi stimulansa kojima je ispitanik posvetio najviše pažnje su potpuno transparentni, onima kojima je manje su delimično transparentni, a onima kojima uopšte nije su zatamnjeni.

3. VEB PLATFORMA ZA TESTIRANJE ZNANJA

U ovoj sekciji će biti prikazani specifikacija i delovi implementacije platforme za testiranje znanja.

3.1 Specifikacija platforme

Platforma treba da podrži dve vrste korisnika: profesora i studenta. Profesor kreira test tako što definiše pitanja, odgovore i regije od interesa u okviru svakog pitanja. Student rešava test putem platforme pri čemu se vrši snimanje njegovog pogleda pomoću uređaja za praćenje pokreta oka. Nakon završenog testiranja student dobija izveštaj o rezultatima na osnovu tačnosti odgovora, dok profesor dobija izveštaj o rezultatima i o tome gde je student gledao prilikom rešavanja testa.

3.2 Tehnologije

Za implementaciju platforme su korišćeni Java sa Spring Boot-om na serverskoj strani i React na klijentskoj strani. Za bazu podataka je korišćena SQL baza MySQL.

3.3 Implementacija regija od interesa

Regije od interesa su uz fiksacije najvažnije za analizu podataka, pa je iz tog razloga nemoguće zamisliti jednu ovakvu platformu bez njih. Implementacija regija je izvršena crtanjem po html stranici u okviru <div> taga. Da bi se ovo omogućilo iskorišćeno je nekoliko stvari:

- kombinacija *Javascript Mouse Event*-ova *mousedown* i *mouseup*, čime se dobila mogućnost da se prevlačenjem po ekranu nacrtaju regije od interesa
- apsolutno pozicioniranje html elemenata uz pomoć *CSS property-a position*
- preklapanje <div> elementa koji sadrži elemente pitanja i <div> elementa koji služi kao *canvas* uz pomoć *CSS property-a z-index*
- korišćenje *viewport CSS* jedinica za širinu i visinu (vh i vw), kao i korišćenje dimenzija ekrana i *browser-a* uz pomoć *Javascript window* objekta

Profesor za vreme definisanja pitanja kreira regije od interesa prevlačenjem miša. Tom prilikom u bazu se smeštaju 4 koordinate, po 2 za gornju levu tačku i donju desnu tačku regije. Ove koordinate su u opsegu od 0 do 1 u zavisnosti od *viewport-a*. Formule za računanje ovih koordinata su sledeće:

$$x = \text{Math.round}\left(\frac{\text{event.clientX} + \text{window.scrollX}}{\text{window.innerWidth} * 10000}\right) * 10000$$

$$y = \text{Math.round}\left(\frac{\text{event.clientY} + \text{window.scrollY}}{\text{window.innerHeight} * 10000}\right) * 10000$$

U toku testiranja 4 koordinate koje su izračunate za vreme definisanja pitanja se koriste prilikom učitavanja svakog pitanja. Ove koordinate se prvo prebacuju u piksele, a zatim u vrednosti u opsegu od 0 do 1, ali ovaj put u zavisnosti od ekrana monitora pošto te vrednosti koriste i uređaji za praćenje pokreta oka. Izračunate vrednosti se čuvaju u bazi zajedno sa vremenom kada je pitanje promenjeno i kasnije se koriste za generisanje izveštaja.

3.4 Implementacija komunikacije sa uređajem za praćenje pokreta oka

Implementacija komunikacije između servera i uređaja za praćenje pokreta oka je ostvarena preko TCP/IP *socket-a*. Ovakav tip komunikacije je zgodan pošto se zasniva na otvorenim standardima TCP/IP protokolu i XML ili JSON

formatu, kao i to što se može koristiti za dobijanje podataka od uređaja u realnom vremenu.

Da bi se postigao ovakav tip komunikacije potrebno je implementirati *ClientSocket* klasu pošto je u ovom slučaju server klijent, a uređaj server. Po pravilu komunikaciju preko *socket-a* započinje klijent, pa je to slučaj i u ovoj platformi. Veza sa uređajem se uspostavlja pre nego što se započne testiranje. Od studenta se traži da preko forme unese ip adresu (ukoliko nije tačna ona koja je automatski popunjena), port i tip uređaja, a zatim se kreira instanca klase *ClientSocket* i poziva njena metoda *connect()*, sa listinga 1. koja startuje dve nove niti. Dve nove niti koje su startovane služe za slanje i primanje poruka ka/od uređaja. Nit zadužena za primanje poruka, sve poruke upisuje u fajl zajedno sa vremenom kad su primljene, a onda se one kasnije parsiraju i koriste za generisanje izveštaja.

```
public boolean connect() {
    try {
        InetAddress addr =
            InetAddress.getByName(address);
        Socket sock = new Socket(addr, port);
        BufferedReader in = new BufferedReader(new
            InputStreamReader(sock.getInputStream()));
        PrintWriter out = new
            PrintWriter(newBufferedWriter(new
                OutputStreamWriter(sock.getOutputStream()),
                true));
        filename = UUID.randomUUID() + ".txt";
        BufferedWriter fileWriter = new
            BufferedWriter(new FileWriter(filename));

        SocketData sd = new SocketData();
        data.put(id, sd);
        new ReaderThread(sock, in, fileWriter);
        new WriterThread(sock, out, sd);
    } catch (Exception ex) {
        ex.printStackTrace();
        return false;
    }
    return true;
}
```

Listing 1. Prikaz *connect()* metode *ClientSocket* klase

Uspešan proces testiranja obuhvata slanje četiri poruke sa servera ka uređaju: za uspostavljanje konekcije, za kalibraciju, za prijem podataka sa uređaja i za raskidanje konekcije.

3.5 Generisanje i prikaz izveštaja

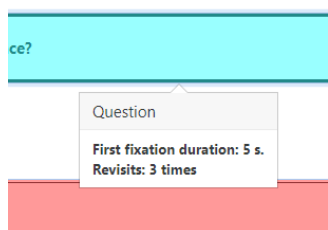
Pre generisanja izveštaja trebalo je doneti odluku koji podaci se očekuju od uređaja. Kako bi se ova odluka donela iskorišćena je dokumentacija *Open Gaze API*-ja [5], *open-source* rešenja za komunikaciju sa uređajem. Sledeći podaci su proglašeni za bitne:

- Counter - brojač pristiglih zapisa sa uređaja. Pomaže da se proverí da li neki zapis fali
- Timer - vreme koje je prošlo od poslednje inicijalizacije sistema
- Fixation POG - sadrži koordinate fiksacije, vreme početka, trajanje, identifikator i indikator da li je fiksacija validna
- Best POG - sadrži prosečnu vrednost tačaka pogleda levog i desnog oka (ukoliko je vrednost jednog oka nedostupna uzima se vrednost tačke pogleda samo onog oka koje je dostupno) i indikator da li je tačka validna

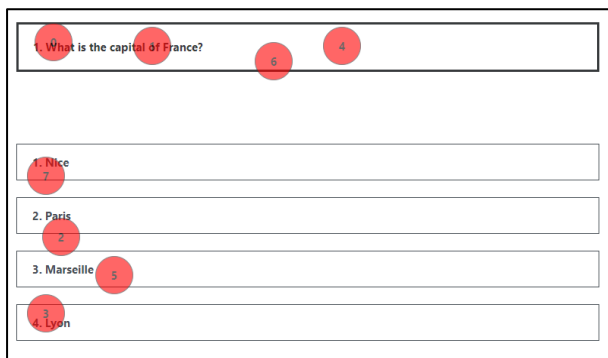
Generisanje izveštaja se vrši automatski. Kada se pokrene generisanje, svi ovi podaci bi trebali da se dobiju parsiranjem fajla koji je napravljen za vreme testiranja. Nakon parsiranja podaci se čuvaju u bazi, zajedno sa vremenom kad su pristigli. Ovo vreme se upoređuje sa vremenom kad je student prelazio sa pitanja na pitanje i na osnovu toga se podaci grupišu po pitanju. Uz podatke koje imamo o regijama, vrše se izračunavanja i na kraju se dobijaju podaci potrebni da se prikažu *first fixation duration*, *revisits*, *fixation sequence* i *heatmap* metrike. Primeri ovih metrika se mogu videti na slikama 5, 6, 7. i 8.



Slika 5 Prikaz regija od interesa



Slika 6. Prikaz *first fixation duration* i *revisits* u okviru regija



Slika 7. Prikaz *fixation sequence*



Slika 8. Prikaz *heatmap*-e

4. ZAKLJUČAK

U radu je predstavljena tehnologija za praćenje pokreta oka i platforma za testiranje znanja koja koristi ovu tehnologiju. Naglasak platforme je stavljen na kreiranje regija od interesa, komunikaciji između platforme i uređaja za praćenje pokreta oka i generisanju izveštaja uz odgovarajuće metrike. Omogućena je podrška za komunikaciju sa više uređaja istovremeno.

Opisana platforma ima za cilj da poboljša obrazovni sistem, tako što će profesorima olakšati kreiranje testova dajući im podatke o studentima koji ne mogu biti prikupljeni ni na jedan drugi način osim upotrebom tehnologije za praćenje pokreta oka. Ipak, treba biti oprezan sa uvođenjem ovakve tehnologije jer ona zahteva laboratorijske uslove, što podrazumeva konstantno snimanje studenata, a u nekim slučajevima čak i nošenje opreme prilikom testiranja. Sve ovo može da izazove nelagodu kod studenta i time stvori suprotni efekat od željenog. Takođe, pokazalo se da uređaji za praćenje pokreta oka u određenim trenutcima nisu precizni ukoliko student nosi naočare ili sočiva.

Trenutna zastupljenost uređaja, pre svega zbog visoke cene, ograničava upotrebu tehnologije za praćenje pokreta oka. Ipak nije teško zamisliti da će jednog dana ona početi intenzivno da se koristi pogotovo ako se nađe način da se dobiju precizni podaci od običnih kamera koje su ugrađene u prenosne računare i mobilne uređaje ili ako se i u te uređaje počnu ugrađivati infracrvene kamere.

Dalji razvoj platforme se može kretati u nekoliko pravaca. Prvi je da se omogući *export* podataka za određene ili sve testove u okviru platforme i da se nad njima izvrši još detaljnija analiza upotrebom mašinskog učenja. Drugi je da se napravi platforma slična opisanoj u ovom radu koja će služiti kao alat protiv varanja na online testovima. Takođe, pošto je iskorišćena samo jedna od savremenih tehnologija, može se razmišljati i o proširenju platforme tako da kombinuje tehnologiju za praćenje pokreta oka sa nekim drugim tehnologijama kao što su tehnologije za praćenje izraza lica ili glasa.

5. LITERATURA

- [1] S. Parhizgar, "Testing and Technology: Past, Present and Future", Theory and Practice in Language Studies 2(1), pp. 174, 2012.
- [2] A.T. Duchowski, "Eye tracking methodology: Theory and practice", Springer, 2017.
- [3] J.H. Goldberg. & A.M. Wichansky, "Eye tracking in usability evaluation: A practitioner's guide", In the Mind's Eye, pp. 493-516, 2003.
- [4] Z. Sharafi, T. Shaffer, B. Sharif & Y. G. Guéhéneuc, "Eye-tracking metrics in software engineering", In 2015 Asia-Pacific Software Engineering Conference (APSEC) pp. 96-103, IEEE, 2015.
- [5] Open Gaze API. dostupno na: <https://www.gazept.com/dl/Gazept API v2.0.pdf> (pristupljeno u oktobru 2020.)

Kratka biografija:

Ognjen Vlajić rođen je u Bačkoj Topoli 1995. godine. Master rad na fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva - Elektronsko poslovanje odbranio je 2020. godine.

**SNAGE U SLOŽENOPERIODIČNIM REŽIMIMA ELEKTROENERGETSKOG SISTEMA
POWERS IN NON-SINUSOIDAL PERIODIC REGIMES OF ELECTRIC POWER SYSTEM**Vladimir Olear, Dejan Jerkan, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu su definisane snage u prostoperiodičnom režimu, a zatim i u složenoperiodičnom režimu. Dati su izrazi za aktivnu, reaktivnu, prividnu i snagu deformacije. Programski paket Matlab je iskorišćen kako bi se pomoću modela realnog pogona u složenoperiodičnom režimu uporedili rezultati simulacija sa računskim rezultatima, shodno priloženim definicijama.

Ključne reči: EES, kvalitet električne energije, Furijeova analiza, viši harmonici.

Abstract – The paper defines various electric powers in both sinusoidal and periodic non-sinusoidal operating conditions. Expressions for active, reactive, apparent and deformation power are given. The Matlab software package was used to compare the simulation results with the computational results, according to the given definitions, using a model of real drive in periodic non-sinusoidal operating conditions.

Keywords: Power system, electricity quality, Fourier analysis, higher harmonics.

1. UVOD

Težnja elektroenergetskog sistema jeste da radi u prostoperiodičnom režimu rada. Ova činjenica posebno je izražena na mrežama viših napona, dok su u pogonima niskog napona moguća neznatna odstupanja.

Pri priključenju nelinearnih potrošača dolazi do pojave struja koji se u praksi nazivaju harmonici. Poželjno je da viši harmonici budu što manje prisutni u strujama i naponima s obzirom da oni izobličuju napon mreže.

Komponente energetskog sistema, kao i potrošači koji se priključuju na njega, podrazumevaju sinusne oblike napona i struje, pa svaka pojava viših harmonika donosi negativne efekte. Kvalitet električne energije je vezan za odstupanje talasnih oblika napona i struja od idealnog prostoperiodičnog oblika. Ukoliko kvalitet električne energije nije zadovoljavajući može se javiti dosta problema. Među najznačajnijim problemima do kojih može dovesti loš kvalitet električne energije, a pre svega harmonijska izobličenja, su: pregrevanje nultog provodnika, pregrevanje transformatora, pregrevanje i nepravilan rad električnih motora, uticaj harmonika na prekidače i zaštitne releje, uticaj na elektronsku opremu, uticaj na rad statičkih energetskih pretvarača, uticaj na telekomunikacione veze, naprezanje kondenzatorskih baterija za kompenzaciju reaktivne energije, izobličenje mrežnog napona usled toka harmonijskih struja, efekti toka harmonika struja kroz provodnike i sabirnice.

NAPOMENA:

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

2. SNAGE U SLOŽENOPERIODIČNOM REŽIMU

U ustaljenom složenoperiodičnom režimu se trenutna, srednja (aktivna) i prividna snaga mreže definišu redom kao i u prostoperiodičnom režimu:

$$p(t) = u(t)i(t) \quad (1)$$

$$P = \frac{1}{T} \int_{\tau}^{\tau+T} p(t) dt \quad (2)$$

$$S = UI \quad (3)$$

U odnosu na prostoperiodični režim situacija je složenija, jer se javljaju proizvodi harmonika napona i struje različitih frekvencija. Proizvod komponenata napona i struje istog indeksa odgovara trenutnoj ulaznoj snazi n -tog harmonika:

$$p^{(n)} = u^{(n)}i^{(n)} \quad (4)$$

Proizvod komponenata različitih indeksa jeste deo ukupne trenutne ulazne snage, nastao usled delovanja različitih harmonika, što možemo označiti kao:

$$p^{(n,m)} = u^{(n)}i^{(m)}, n \neq m \quad (5)$$

Srednja (aktivna) snaga u ustaljenom periodičnom režimu srednja snaga se posmatra za vreme jedne periode (ili celobrojnog umnoška perioda) i opisana je izrazom:

$$P = \frac{1}{T} \int_0^T p(t) dt \quad (6)$$

Znači da je u ustaljenom periodičnom režimu srednja snaga jednaka sumi srednjih snaga svih harmonika:

$$P = \sum_{n=0}^{\infty} p^{(n)} \quad (7)$$

Srednja snaga se i ovde, često, naziva i aktivnom snagom jer označava brzinu kojom se konstantno isporučuje energija mreži, tj. označava snagu koja se nepovratno disipira u mreži. Taj naziv ne treba mešati sa nazivima aktivna/pasivna mreža.

Prividna snaga u periodičnom režimu se definiše na isti način kao u prostoperiodičnom režimu: kao proizvod efektivnih vrednosti napona i struje na pristupu mreže:

$$S = UI \quad (8)$$

Ta snaga objektivno opterećuje izvor, mada se stvarno (aktivno) u prijemniku "troši" samo snaga P. S obzirom na definiciju efektivne vrednosti složenoperiodične veličine, prividna snaga je, dalje jednaka:

$$S = \sqrt{(u^{(0)})^2 + \sum_{n=1}^{\infty} (U^{(n)})^2} \sqrt{(i^{(0)})^2 + \sum_{n=1}^{\infty} (I^{(n)})^2} \quad (9)$$

Kao merilo iskorišćenja mreže kao (aktivnog) potrošača električne energije i u složenoperiodičnom režimu se može uvesti pojam faktora snage, što je, formalno, isto kao u prostoperiodičnom režimu:

$$k_p = P/S \quad (10)$$

Prividna snaga u ustaljenom prostoperiodičnom režimu se opisuje jednostavnim izrazom:

$$S^2 = P^2 + Q^2 \quad (11)$$

gde su P i Q aktivna i reaktivna snaga mreže. Analizom složenoperiodičnog režima, pokušano je da se i tu formira slična relacija.

$$S^2 = \left[\sum_{n=0}^{\infty} U^{(n)} I^{(n)} \cos \varphi^{(n)} \right]^2 + \left[\sum_{n=0}^{\infty} U^{(n)} I^{(n)} \sin \varphi^{(n)} \right]^2 + \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} [U^{(n)2} I^{(m)2} + U^{(m)2} I^{(n)2} - 2 U^{(n)} U^{(m)} I^{(n)} I^{(m)} \cos(\varphi^{(n)} - \varphi^{(m)})] \quad (12)$$

Unutar izraza (12) prepoznaje se aktivna snaga mreže u složenoperiodičnom režimu, pa je taj član zapravo, oblika:

$$\left[\sum_{n=0}^{\infty} U^{(n)} I^{(n)} \cos \varphi^{(n)} \right]^2 = \left[\sum_{n=0}^{\infty} P^{(n)} \right]^2 = P^2 \quad (13)$$

Drugi član na desnoj strani izraza (12) sadrži, unutar zagrade, sumu reaktivnih snaga harmonika. Po analogiji sa aktivnom snagom, ova suma se često naziva reaktivnom snagom mreže u složenoperiodičnom režimu i označava sa Q:

$$\sum_{n=0}^{\infty} U^{(n)} I^{(n)} \sin \varphi^{(n)} = \sum_{n=0}^{\infty} Q^{(n)} \quad (14)$$

Preostali deo izraza (12) nastaje kao proizvod različitih harmonika napona i struje. Taj član ima istu dimenziju kao ostali članovi u jednačini: to je kvadrat neke fiktivne snage:

$$D^2 = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} [U^{(n)2} I^{(m)2} + U^{(m)2} I^{(n)2} - 2 U^{(n)} U^{(m)} I^{(n)} I^{(m)} \cos(\varphi^{(n)} - \varphi^{(m)})] \quad (15)$$

Oznaka D sledi iz uobičajenog naziva te veličine: to je snaga deformacije (ili snaga izobličenja). Naziv je uveo

Budeanu 1927. godine i, mada je dosta osporavan, taj koncept se i danas, ponekad koristi. Saglasno uvedenim veličinama snagama P, Q i D, kvadrat prividne snage se može prikazati u obliku:

$$S^2 = P^2 + Q^2 + D^2 \quad (16)$$

Snaga deformacije se, ako je potrebno, može relativno lako odrediti na osnovu poslednjeg izraza kao:

$$D = \sqrt{S^2 - P^2 - Q^2} \quad (17)$$

s obzirom da su snage S, P i Q određene (relativno) jednostavnim izrazima (9), (7) i (14), koji su (u delovima) slični odgovarajućim izrazima za prostoperiodični režim. Jedinica mere, snage deformacije se navodi kao VAd-volt-amper deformacije.

U električnim kolima, koja se karakterišu prisustvom viših harmonika u struji i naponu, uobičajena predstava faktora snage kao $\cos \varphi$ nije tačna. Na bazi osnovne definicije faktor snage je iskazan složenijim izrazom:

$$\lambda = \frac{P}{S} = \frac{\sum_{n=1}^{\infty} U_n I_n \cos \varphi_n}{\sqrt{\sum_{n=1}^{\infty} U_n^2} \sqrt{\sum_{n=1}^{\infty} I_n^2}} \quad (18)$$

2.1. Definicija THD-a

Ukupna harmonijska distorzija (THD) se definiše kao kvadratni koren odnosa sume kvadrata efektivnih vrednosti pojedinačnih harmonijskih komponenti i efektivne vrednosti fundamentalnog harmonika i takođe se izražava u procentima:

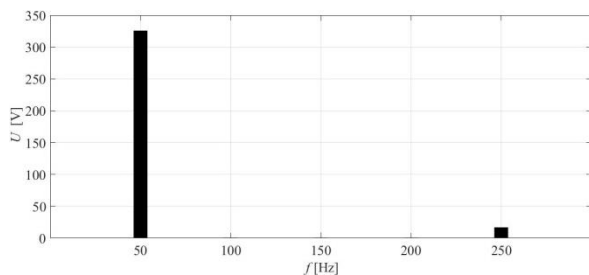
$$THDI = \sqrt{\frac{\sum_{n=2}^{\infty} I_n^2}{I_1^2}} 100 = \sqrt{\sum_{n=2}^{\infty} HDI_n^2} (\%) \quad (19)$$

3. NUMERIČKI PRIMER REALIZOVAN U MATLAB-U

Radi sagledavanja pojava napravljena je simulacija u Matlab Simulink programu, u ovom poglavlju biće diskutovani i predstavljeni rezultati simulacije. Simulacija se sastoji od mreže koja daje ne idealno sinusni signal već signal sa petim harmonikom u talasnim oblicima napona, zatim ispravljača snage od oko 5 KW, nominalno opterećenog asinhronog motora od 75 KW i elemenata kojima se modeluju impedase napojnih vodova.

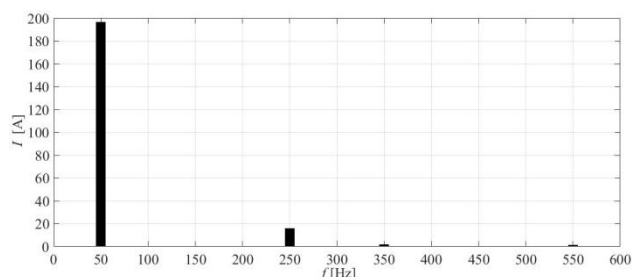
3.1. Električne veličine merene na korenu mreže

Model je korišćen u složenoperiodičnom režimu, napon mreže se sastoji od prvog i petog harmonika u vrednosti od 5%. Primenom Furijeove analize i Matlab Simulink programa mogu se dobiti odzivi koji prikazuju zavisnost amplitude od frekvencije. Primetno je postojanje prvog i petog harmonika u naponu mreže. Sa slike 3.1. može se očitati vrednosti harmonika: $U_1 = \frac{325,23}{\sqrt{2}} V$, $U_5 = \frac{16,22}{\sqrt{2}} V$.



Slika 3.1. Amplitudni spektar napona mreže

Primenom Furijeove analize i Matlab Simulink programa može se dobiti odziv koji prikazuje zavisnost amplitude struje od frekvencije. Sa slike 3.2. očitavaju se vrednosti struja po harmonicima: $I_1 = \frac{196,45}{\sqrt{2}} A$, $I_5 = \frac{15,76}{\sqrt{2}} A$, $I_7 = \frac{1,478}{\sqrt{2}} A$, $I_{11} = \frac{1,1}{\sqrt{2}} A$.



Slika 3.2. Amplitudni spektar struje mreže

Pomoću očitanih vrednosti sa slika iznad i izraza (7), (14) (9), (17) i (18) mogu se proračunati pojedinačno snage i faktor snage kod složenoperiodičnog režima:

Izračunata vrednosti-pozicija 1	Vrednost iz Matlab blokova mreže
$P_{mreže} = 88,27 \text{ kW}$	$P_{mreže} = 86,77 \text{ kW}$
$Q_{mreže} = 37,7 \text{ kVAr}$	$Q_{mreže} = 37,67 \text{ kVAr}$
$S_{mreže} = 96,27 \text{ kVA}$	$S_{mreže} = 94,59 \text{ kVA}$
$D_{mreže} = 7,37 \text{ kVAd}$	/

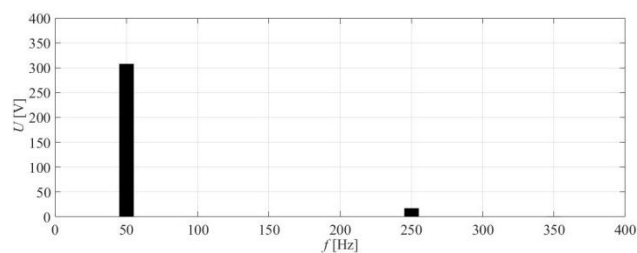
$\lambda = 0.9169$, odstupanje: $P_{mreže}$ je 1,71 %, $Q_{mreže}$ je 0,0716 %, $S_{mreže}$ je 1,74 %.

Iz tabele iznad vidi se da se vrednosti koje Matlab-ov blok pokaže i proračunate vrednosti malo razlikuju, u izraz za snagu $S_{mreže}$ treba uračunati i snagu $D_{mreže}$ koju blok u Matlab-u ne računa, te zbog toga postoje mala odstupanja.

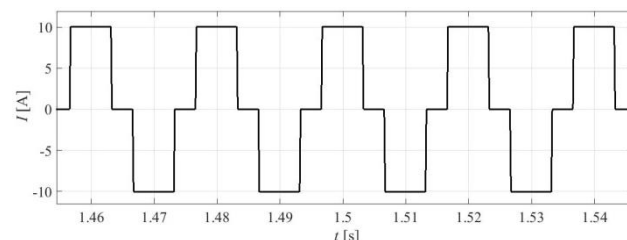
3.2. Električne veličine merene na ispravljaču

U modelu se koristi trofazni diodni ispravljač snage oko 5 KW, THD napona na priključcima ispravljača iznosi oko 5,5%. Dominantan je prvi i peti harmonik, dok je sedmi harmonik vrlo male vrednosti, slika 3.3.

Diodni ispravljač je nelinearan potrošač koji prouzrokuje izobličenje struje, unosi više harmonike, struja na ispravljaču je trapezoidalna zbog velike vrednosti prigušnice u kolu, slika 3.4.

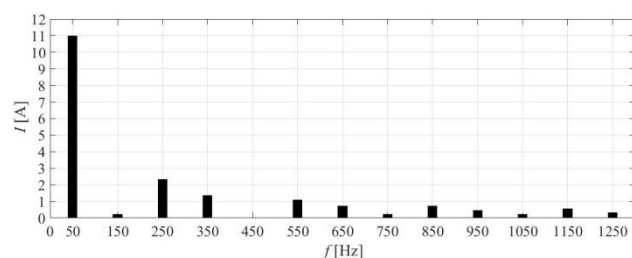


Slika 3.3. Amplitudni spektar napona na ispravljaču

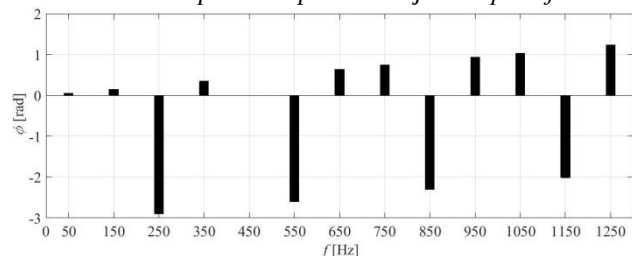


Slika 3.4. Fazna struja na ispravljaču

Sa slike 3.5. se očitavaju struje ispravljača po harmonicima. Struja sadrži dominantni prvi, peti, sedmi, jedanaesti, trinaesti, petnaesti, sedamnaesti, devetnaesti itd. harmonik. Sa slike 3.6. se može očitati argument struje.

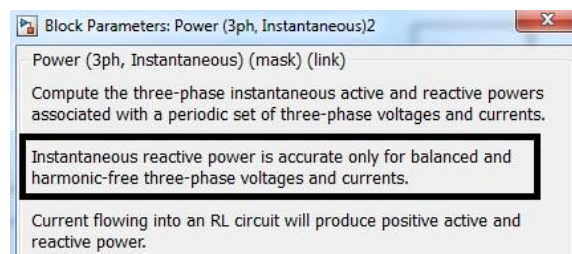


Slika 3.5. Amplitudni spektar struje na ispravljaču



Slika 3.6. Amplitudni spektar struje na ispravljaču

Pomoću očitanih vrednosti sa slika iznad i izraza (7), (14) (9) i (17) mogu se proračunati pojedinačno snage kod složenoperiodičnog režima.



Slika 3.7. Greška kod merenja reaktivne snage

Blok koji meri P i Q u Matlabu pravi grešku kod merenja snage Q, gde u slučaju ispravljača postoji najveće izobličenje struje pa je tu odstupanje najveće i iznosi čak

97,7%. $Q_{\text{ispravljača}} = 2,97 \approx 0 \text{ VAr}$, dok je snaga $D_{\text{ispravljača}}$ oko trećine od ukupne snage ispravljača.

Izračunata vrednost- pozicija 2	Vrednost iz Matlab bloka na ispravljaču
$P_{\text{ispravljač}} = 5.008 \text{ KW}$	$P_{\text{ispravljač}} = 5.06 \text{ KW}$
$Q_{\text{ispravljač}} = 2.969 \text{ VAr}$	$Q_{\text{ispravljač}} = 129.225 \text{ VAr}$
$S_{\text{ispravljač}} = 5287.5 \text{ VA}$	$S_{\text{ispravljač}} = 5.016 \text{ KVA}$
$D_{\text{ispravljač}} = 1.68 \text{ KVAd}$	/

$\lambda = 0.9471$, odstupanje: $P_{\text{ispravljač}}$ je 1.17%,
 $Q_{\text{ispravljač}}$ je 97.7%, $S_{\text{ispravljač}}$ je 5.14%.

3.3. Električne veličine merene na motoru

Pomoću očitanih vrednosti sa slika i izraza (7), (14) (9), (17) i (18) mogu se proračunati pojedinačno snage i faktor snage kod složenoperiodičnog režima i uporediti sa Matlab rezultatima.

Izračunata vrednost- pozicija 3	Vrednost iz Matlab bloka na motoru
$P_{\text{motor}} = 77,44 \text{ kW}$	$P_{\text{motor}} = 75,92 \text{ kW}$
$Q_{\text{motor}} = 37,689 \text{ kVAr}$	$Q_{\text{motor}} = 37,57 \text{ kVAr}$
$S_{\text{motor}} = 86,36 \text{ kVA}$	$S_{\text{motor}} = 81 \text{ kVA}$
$D_{\text{motor}} = 6,4 \text{ KVAd}$	/

$\lambda = 0.8967$, odstupanje: $P_{\text{ispravljač}}$ je 1,97%,
 $Q_{\text{ispravljač}}$ je 0,31, $S_{\text{ispravljač}}$ je 6,21%. Kod asinhronog motora su znatno manja izobličenja nego kod ispravljača, te snaga deformacije D_{motor} iznosi oko 8 % od ukupne snage motora.

4. ZAKLJUČAK

U slučaju linearnih opterećenja, snaga preuzeta iz naizmeničnog izvora ima tri komponente: prividnu, aktivnu i reaktivnu, dok u slučaju nelinearnih opterećenja postoji i četvrta komponenta, snaga deformacije. Nelinearna opterećenja kao što su diodni ispravljači, kada se priključe na prostoperiodični izvor napona, neće imati prostoperiodičan (sinusoidalni) talasni oblik struje što unosi zaprljanje. Uticaj naponskih harmonika: stvaraju dodatno zagrevanje asinhronih motora, sinhronih motora i generatora, pikovi viših harmonika mogu izazvati starenje izolacije kablova, namotaja i kondenzatora, neispravan rad elektronskih komponenti i kola koja koriste naponski talas za sinhronizaciju.

Matlab model dovoljno tačno prikazuje rezultate simulacije izuzev kod velikog zaprljanja mreže gde se pojavljuju mnogo viših harmonika tu je najveće odstupanje kod reaktivne snage, snaga deformacije je velike vrednosti (snaga deformacije ispravljača je oko trećine od prividne snage ispravljača), dok kod motora je snaga deformacije oko osam procenata od ukupne snage motora. U praksi je poželjno da viši harmonici budu što manje vrednosti zbog raznih neželjenih pojava. Kod složenoperiodičnog režima je potrebno da reaktivna snaga i snaga deformacije budu što manje vrednosti.

5. LITERATURA

- [1] [Branimir D. Reljin, Teorija električnih kola II, rešavanje kola u frekvencijskom domenu, Akademski misao 2009 Beograd.]
- [2] V.Katic: "Energetski elektronski pretvaraci kao izvori viših harmonika u distributivnoj mreži – I deo", Studija za SOUR Elektrovojvodina, Novi Sad, mart 1990.
- [3] V.Katic: "Kvalitet električne energije – viši harmonici", Monografija, Univerzitet u Novom Sadu - Fakultet tehničkih nauka, Edicija Tehnicke nauke - Monografije, Br. 6, Novi Sad, 2002.
- [4] F.Martzloff, T.Gruzs: "Power Quality Site Surveys: Facts, Fiction and Fallacies", IEEE Tran. on Industry Applic., Vol.24, No.6, Nov./Dec. 1988, pp.1005-1018.

Kratka biografija:

Vladimir Olear rođen u Vrbasu 1993. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Energetska elektronika i električne mašine odbranio je 2020.god.

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

TESTIRANJE GRAPHQL APLIKACIJA**TESTING GRAPHQL APPLICATIONS**Aleksandra Milivojević, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu su objašnjeni osnovni koncepti GraphQL specifikacije. Opisani su tipovi i strategije testiranja softvera prema ovom konceptu. Osmišljena je aplikacija za demonstraciju i implementaciju testova specifičnih za sisteme koji koriste GraphQL.

Ključne reči: GraphQL, testiranje softvera, integracioni testovi, unit testovi

Abstract – This paper explains the basic concepts of GraphQL specification. Types and strategies of software testing according to this concept are given. The application that demonstrates it is implemented. The tests specific for systems that use GraphQL are written and explained.

Keywords: GraphQL, software testing, integration testing, unit testing

1. UVOD

GraphQL [1] je upitni jezik za API-eve čija je specifikacija otvorenog koda. Ovaj novi koncept osmišljen je 2012. godine od strane programera Facebook aplikacije, kao alternativa tradicionalnom RestFULL pristupu radi povećanja efektivnosti i optimalnijeg korištenja resursa koji se dobavljaju sa servera. Ideja ovakvog GraphQL pristupa je da se novim upitnim jezikom specificira koji su atributi neophodni u odgovoru servera.

Testiranje softvera je proces koji se sastoji od svih aktivnosti u životnom ciklusu softvera. Oni bi trebalo da pokažu da li softver zadovoljava specificirane zahteve i ispunjava namenu, kao i da detektuju eventualne greške u sistemu. Kako je GraphQL pristup drastično promenio koncepte slanja zahteva sa klijentske strane i odgovora sa serverske strane, javile su se potrebe za testiranje ovakvih aplikacija na malo drugačiji način.

Za potrebe ovog rada, implementirana je web aplikacija za demonstraciju objašnjenih GraphQL koncepata, a naročito testova specifičnih za njih. Na kraju su prikazani i objašnjeni testovi delova aplikacija specifičnih za GraphQL koncepte.

2. GRAPHQL

GraphQL [1] je upitni jezik i kao nova tehnologija pruža potpun i opšti prikaz strukture podataka ka kojim manipuliše API. Samim tim daje klijentima mogućnost da pri dobavljanju istih, jednostavno traže samo ono što im je potrebno i ništa više od toga.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Goran Savić.

2.1 Istorijat i motivacija

Kada je odlučeno da se 2012. godine naprave nove Facebook mobilne iOS i Android aplikacije, bio je potreban novi i drugačiji API, koji bi omogućio dobavljanje sirovih podataka. GraphQL kao novi API standard napravljen je interno od strane razvojnog tima kompanije Facebook 2012. godine, dok je 2015. godine javno publikovana specifikacija, implementacija i oformljena zajednica korisnika.

2.2 Glavni koncepti

Glavni koncepti objašnjavaju na šta se sve odnosi sam termin GraphQL, zatim kako se pristupa ovakvom servisu, kako se definiše struktura podataka i na kraju kako se podaci dobavljaju. Dosadašnji koncept pristupanja API-u zahtevao je da postoje više krajnjih tačaka (*eng. endpoint*) koji vraćaju podatke fiksne strukture. Za GraphQL pristup, neophodan je jedan endpoint koji se gađa, dok se u telu zahteva detaljno specificira zahtev. Jezik za definisanje šeme (*eng. Schema Definition Language, SDL*) je način za definisanje i opis GraphQL šeme. Ovakva sintaksa predstavlja deo zvanične GraphQL specifikacije. Glavne komponente definicije šeme su **tipovi** i njihova **polja**, a pored ovih mogu se naći i dodatne informacije o šemi u vidu direktiva. Tip (*eng. Type*) u GraphQL-u bi predstavljao pandam klasi u objektno orijentisanom modelu podataka. Kada se tip definiše, neophodno je navesti njegovo ime i polja koja mu pripadaju. U kompleksnijim šemama moguće je definisati i koristiti još i enumeracije, interfejsse, fragmente i direktive.

Klijent svoj zahtev ka serveru piše u vidu upita, mutacija ili pretplata kojima mogu biti prosleđeni parametri. Upiti služe za dobavljanje podataka sa API-a, mutacije za iniciranje menjanja podataka (njihovim kreiranjem, ažuriranjem ili brisanjem), a pretplatama se otvara konekcija ka serveru preko koje će u realnom vremenu stizati odgovori vezani za tipove šeme na koje se klijent pretplatio. Uvedene su i tzv. **resolver** funkcije. One se pišu na serverskoj strani sistema i svaka pojedinačno odgovara jednom GraphQL upitu. Resolver funkcije zadužene su da obrade upit kom odgovaraju i kreiraju konačnu povratnu vrednost, uzimajući u obzir samo polja koja je klijent naveo da su mu potrebna.

2.4 GraphQL i Rest

Ovakav novi pristup izrazito odgovara razvijanju front-end aplikacija, jer je sva kompleksnost prebačena na serversku stranu. Glavni cilj nove GraphQL specifikacije bio je da postigne veću fleksibilnost i efikasnost pri komunikaciji između klijentske i serverske strane. Ono što je viđeno kao najčešći problem RestFULL aplikacija

je problem *overfetching*-a i *underfetching*-a. Takođe, uvođenjem GraphQL pristupa dobija se i detaljniji uvid u podatke koji su zaista potrebni sa serverske strane. Analizirajući upite koji dolaze sa klijentske strane, može se bolje razumeti kako i koji od dostupnih podataka se zaista koriste.

3. TESTIRANJE GRAPHQL APLIKACIJA

GraphQL aplikacije su promenile koncept interakcije i komunikacije između klijentske i serverske strane sistema. To je upravo dovelo do novih aspekata prilikom prolaska kroz neke od faza razvoja softvera, kao što je testiranje.

3.1 Testiranje softvera

Testiranje softvera je proces koji se sastoji od svih aktivnosti u životnom ciklusu softvera, vezanih za: planiranje, pripremu i izvršavanje zadataka. Oni bi trebalo da pokažu da li softver zadovoljava specificirane zahteve i ispunjava namenu, kao i da detektuju eventualne greške u sistemu. Iz prethodne definicije direktno sledi da proces testiranja veoma utiče na povećanje kvaliteta softvera. Sa druge strane, često se navodi da potpuno testiranje proizvoda nije izvodivo u praksi. Iz tog razloga važno je fokusirati se na kritične delove sistema čije testiranje će poboljšati kvalitet i otkloniti eventualne nedostatke. Kada je reč o veb aplikacijama, testiranjem se teži da ona bude stabilna u produkciji i funkcioniše prema zahtevima kada stvarni krajnji korisnici počnu da je koriste. Pisanje testova takođe utiče na pisanje kvalitetnijeg koda. Kao jedna od važnih svrha testiranja može se navesti i dokumentovanje funkcionalnosti. Dobro testiran kod objašnjava, koje je očekivano ponašanje aplikacije, koji granični slučajevi postoje, kao i koje greške se mogu pojaviti [4].

3.2 Vrste testiranja softvera

Ovaj rad usmeren je na funkcionalne zahteve softvera koji koristi GraphQL specifikaciju, te će samim tim i biti izučavane funkcionalne vrste testova, i to: jedinični (*eng. unit tests*), integracioni (*eng. integration tests*) i sistemski testovi (*eng. system tests*).

Jedinični testovi testiraju pojedinačne komponente softvera i one su po obimu, testovi najmanjeg nivoa granulacije. Proveravaju funkcionalnost manjeg dela koda, koji funkcioniše kao celina i uglavnom ne zavisi od ostalih spoljnih komponenti. **Integracioni testovi** namenjeni su otkrivanju nedostataka u interfejsima i interakciji između međusobno povezanih softverskih komponentata. Važno je pri njihovoj integraciji pisati i testove koji će proveriti funkcionalnost celokupnog integrisanog dela sistema. **Sistemski testovi** služe za kompletno testiranje integrisanog sistema da bi se utvrdilo da li sistem u celini ispunjava zahteve.

Ovakva vrsta testova poznata je još i po nazivu testiranja “s kraja na kraj” (*eng. end-to-end testing, e2e*), koji bolje naglašava testiranje akcija od samog početka korisnikove interakcije do krajnjeg odgovora sistema na tu akciju.

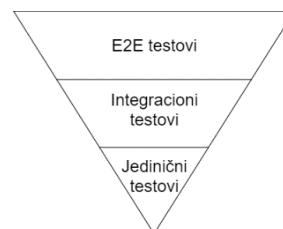
Piramida testiranja predstavlja jednu od strategija pisanja testova. Prvi put je spomenuta od strane Majka Kona (Mike Cohn) [2] a zasniva se na konceptu iz psihologije poznatog kao “Piramida potreba” [3] (*eng. Pyramid of Needs*).



SLIKA 3.1 STRATEGIJA PIRAMIDA TESTIRANJA

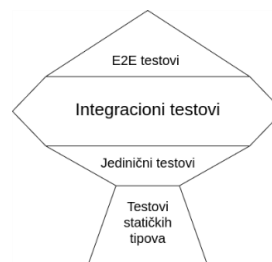
Ovakav prikaz strategije testiranja pokazuje kako se pojedinačni, integracioni i sistemski testovi oslanjaju jedan na drugi i to iz ugla programera. Na dnu piramide se nalaze pojedinačni testovi, što govori da je važno testirati što više pojedinačnih komponenti, jer se na njima zasniva rad cele aplikacije. Zatim su u sredini integracioni testovi, jer bi trebalo proveriti komunikaciju tih komponenti. Na kraju, na vrhu piramide se nalaze sistemski testovi, jer su oni najteži i najsporiji, a u većoj meri se mogu osloniti na prethodna dva tipa testova kojih bi bilo više.

Sa druge strane, posmatrano iz ugla vođe projekta (*eng. project manager*) sistemski testovi imaju najveću važnost. Zbog toga je nastala i struktura **obrnute piramide testiranja** (*eng. reverse testing pyramid*) koja pokazuje drugu strategiju odnosa ova tri tipa testova (Slika 3.2) [4].



SLIKA 3.2 STRATEGIJA OBRNUTE PIRAMIDE TESTIRANJA

Nakon razmatranja ovakvih strategija testiranja, došlo se do kompromisnog rešenja prikazanog, opet simbolično, figurom trofeja. Nova strategija razvoja i odnosa testova nazvana je **trofej testiranje** (*eng. Testing Trophy*) [5].



SLIKA 3.3 STRATEGIJA TROFEJ TESTIRANJA

3.3 Testovi specifični za GraphQL aplikacije

Poznavajući osnovne koncepte GraphQL specifikacije, može se zaključiti da testiranje ovakvih aplikacija zahteva dodatne testove specifične upravo za te koncepte. Sa serverske strane sistema, očekivani su pojedinačni (*unit*) testovi GraphQL šeme i *resolver* funkcija. Prilikom testiranja šeme proveravaju se da li su polja kreiranih tipova očekivana prema specifikaciji. Ovakvi testovi neće biti od ključne važnosti sve dok ne bude porasla kompleksnost šeme, ali oni će svakako doprineti povećanju pokrivenosti aplikacije testovima. Na serverskoj strani pišu se i integracioni testovi specifični za GraphQL koncepte. Nakon imitiranja (*eng. mocking*)

šeme i testiranja *resolver* funkcija, testovima se potvrđuje da li slanje upita, mutacije ili pretplate pogađa očekivanu *resolver* funkciju i vraća očekivani odgovor na klijentsku stranu [4]. Sa klijentske strane sistema, korisno je pisati testove za komponente koje dobavljaju podatke od GraphQL API-a, iniciraju njihovu promenu ili se pretplaćuju na događaje vezane za te iste podatke.

Da bi se testirala sama konekcija između klijentske i serverske aplikacije, dobro je prethodno testirati funkcionalnost samih *resolver* funkcija pojedinačnim ili integracionim testovima. Za ove potrebe takođe se koriste i razne biblioteke koje pomažu imitiranje podataka kakvi bi mogli doći kao odgovor servera na upit, tzv. mokovanje (*eng. mocking*).

4. STUDIJA SLUČAJA

Za potrebe pisanja ovog rada, osmišljena je aplikacija koja bi svojim funkcionalnostima pokrila i iskoristila sve osnovne koncepte GraphQL specifikacije. Aplikacija nazvana *sci-demo-lib*, omogućava pretragu baze naučnih časopisa i radova u njima, zatim autora radova i naučnih oblasti. Korisnici aplikacije bili bi istraživači, profesori ili naučnici sa određenog univerziteta, koji sa univerzitetskom email adresom kreiraju nalog, a zatim dobijaju pristup za pregled i pretraživanje dostupnog materijala.

4.1 Funkcionalnosti i pregled sistema

Registracija korisnika moguća je svakom ko poseduje univerzitetsku email adresu. Tada se popunjavaju polja sa email adresom, lozinkom, potvrdom lozinke, imenom i prezimenom. Prilikom prijave potrebno je uneti email adresu i lozinku već kreiranog korisnika. Prijavljen korisnik se može i odjaviti sa sistema. Na stranicu za prikaz i ažuriranje sopstvenog profila prijavljeni korisnik dolazi odmah nakon prijave na sistem ili kroz glavni meni.

Korisniku aplikacije su dostupni lista autora, časopisa, rada i oblasti, a zatim i informacije o svakom pojedinačno. On ima mogućnost da se pretplati na neki od njih, što znači da će ubuduće dobijati obaveštenja od aplikacije kada časopis objavi novo izdanje, zatim kada autor objavi novi rad ili kada se pojavi novi časopis, rad ili autor koji se bavi određenom oblašću.

4.2 Model podataka

Model podataka sistema preslikava se na odgovarajuće tabele u bazi podataka sistema. Njime su modelovani entiteti koje se pojavljuju u aplikaciji kao što su: *User*, *UserRole*, *Paper*, *Journal*, *ScienceArea*, *University*, *City* i *Country*. Ova aplikacije razlikuje tri tipa korisnika: administrator, autor i običan korisnik. Svaki korisnik, rad i časopis označeni su određenim naučnim oblastima kojima se bave.

Naučni rad obavezno mora imati jednog glavnog autora i može imati ni jednog ili više koautora rada, dok svaki časopis referencira na naučne radove koji su u njemu objavljeni. Kao što je već opisano, korisnik ima mogućnost da se pretplati na nekog od autora, naučni rad ili časopis, te zbog toga klasa korisnika takođe ima i N:N vezu ka ovim klasama.

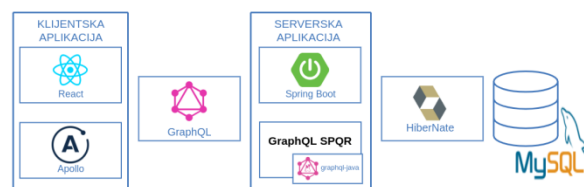
4.3 GraphQL šema

GraphQL šema igra jednu od vodećih uloga u komunikaciji između klijentske i serverske strane sistema. Analizom šeme aplikacije može se uvideti da svaka klasa iz modela podataka odgovara jednom novokreiranom tipu u šemi. Ovo se objašnjava time da se pri dobavljanju podataka iz baze dobijaju objekti tih klasa, koje se dalje prosleđuju na klijentsku stranu sistema, na primer kao odgovor na GraphQL upit.

4.4 Implementacija

Opisani sistem sastoji se od klijentske i serverske aplikacije. Klijentska aplikacija koristi GraphQL tako što šalje zahteve u vidu GraphQL upita, mutacija i pretplata, a na serverskoj strani se ovi zahtevi obrađuju i odgovore šalju nazad. Na slici 4.1 mogu se videti tehnologije koje su korišćene u ovom sistemu za demonstraciju. Za izradu klijentske aplikacije, korišćen je javascript programski jezik i ReactJS [6] biblioteka za izradu celokupnog korisničkog interfejsa. Za potrebe slanja zahteva i preuzimanja odgovora preko GraphQL-a, korišćena je Apollo biblioteka otvorenog koda [7] **Error! Reference source not found..** Pomoću ove dve biblioteke znatno je olakšano dobavljanje, keširanje i menjanje podataka u aplikaciji, dok se korisnički interfejs automatski osvežava. Serverska strana sistema implementirana je kao Spring Boot aplikacija [14]. Za implementaciju GraphQL specifikacije na serverskoj strani, odabrana je GraphQL SPQR biblioteka.

Apollo biblioteka osnovnu, ugrađenu podršku daje React aplikacijama. Na početku neophodno je dodati njenu konfiguraciju. Zatim, da bi se tako kreirani klijent objekat povezo sa React aplikacijom koristi se ApolloProvider koji služi kao omotač i omogućava da mu se pristupi iz bilo kog dela aplikacije.



SLIKA 4.1 TEHNOLOGIJE KORIŠĆENE PRI IZRADI SCI-LIB-DEMO APLIKACIJE ZA DEMONSTRACIJU

Kada je podešena, komponente su spremne za slanje zahteva u vidu upita, mutacija i pretplata. Od verzije 3.0 Apollo biblioteke, zahtevi se pišu u formi React *hook*-ova (*eng. React hooks*) [9]. Komponenta može da prepozna promene statusa odgovora i prikazuje drugačiji korisnički interfejs. Sa druge strane, za implementaciju GraphQL specifikacije serverskog dela sistema, korišćena je GraphQL SPQR biblioteka otvorenog koda [10]. Kada se razvijaju serverske GraphQL aplikacije, često se koristi pristup gde se prvo definiše šema, a kasnije se ona spaja sa biznis logikom aplikacije (*schema-first* pristup). Ovakav pristup dovodi do velike količine dupliranog koda. Da bi se takav pristup izbegao, u *sci-lib-demo* aplikaciji korišćena je GraphQL SPQR biblioteka, koja koristi pristup poznat kao *code-first*. Dovoljno je prilikom implementacije biznis logike aplikacije anotirati određene

java funkcije i klase kako bi ova biblioteka znala kako da generiše šemu. U listingu 4.1 može se videti primer anotiranja jedne *resolver* funkcije, da bi na kraju pomenuta biblioteka znala kako da izgeneriše *getUser* upit unutar šeme.

```
@GraphQLQuery(name="getUser")
public User
getUser(@GraphQLArgument(name="id")
Long id) {
    return userService.findOne(id);
}
```

LISTING 4.1 PRIMER ANOTIRANJA RESOLVER FUNKCIJE

4.5 Testiranje sci-lib-demo aplikacije

U testove specifične za GraphQL aplikacije spadaju testovi GraphQL šeme i *resolver*-a na serverskoj strani sistema, kao i testovi komponenti na klijentskoj strani. Prilikom testiranja šeme, neophodno je proveriti da li ona sadrži sve tipove koji se očekuju, i da li pojedinačno svaki tip sadrži određena polja određene vrste. Testovi koji proveravaju generisanu šemu implementirani su pomoću standardne JUnit biblioteke [11]. Potrebno je u testu kreirati GraphQLSchema objekat koji će generisati šemu za testiranje, a kasnije se iz tog objekta dobijaju i proveravaju određeni delovi šeme. Posle toga slede provere tipova i polja u tipovima.

Za potrebe pisanja integracionih testova za *resolver* funkcije korištena je takođe JUnit biblioteka [11]. Svakom testu ovog tipa potreban je instanciran GraphQLSchema objekat, da bi se pomoću njega kreirao GraphQL objekat koji omogućava izvršavanje upita. Kako se unutar *resolver* funkcije pozivaju servisi i dobavljaju ili menjaju podaci iz baze, njih je potrebno izimitirati (eng. *mocking*). Na ovaj način proverava se sam poziv upita u odnosu na njegov naziv i eventualno parametre. Za mokovanje *resolver* funkcija korištena je Mockito biblioteka [12]. Ona je iskorištena da bi se prosledila klasa UserResolver-a koja će tada biti instancirana. Pri testiranju vizuelnih komponenti na klijentskoj aplikaciji, potrebno je proveriti da li ona dobija od servera očekivani odgovor. To bi spadalo u integracioni test, a zatim bi trebalo da se proveriti da li se komponenta sa dobijenim podacima i prikazuje onako kako je predviđeno, što se dalje svrstava u jedinične testove. Za potrebe pisanja testova za komponente korisničkog interfejsa korišten je Jest [13] radni okvir za testiranje, dok je za pokretanje testova korištena React Test Renderer biblioteka [14]. Jedinično testiranje komponenti korisničkog interfejsa takođe zahteva mokovanje odgovora servera na poslate zahteve. Apollo biblioteka obezbeđuje MockedProvider komponentu koja upravo maskira pozive GraphQL server.

7. LITERATURA

- [1] The Linux Foundation. A query language for your API, <https://graphql.org/>
- [2] Mike Cohn, Succeeding with Agile: Software Development Using Scrum, Signature Book, 2009
- [3] Anne Mette Hass, Guide to Advanced Software testing, Artech House, 2014
- [4] Roy Derks, Testing GraphQL: From Zero to Hundred Percent, NDC Conference
- [5] Kent C. Dodds Static vs Unit vs Integration vs E2E Testing for Frontend Apps, www.kentcdodds.com/blog/unit-vs-integration-vs-e2e-tests
- [6] Facebook Inc., ReactJS <https://reactjs.org/>, pristupljeno: 15.10.2020.
- [7] Apollo Graph Inc., „Apollo GraphQL“, www.apollographql.com/, pristupljeno: 15.10.2020.
- [8] Pivotal „SpringBoot“ <https://spring.io/projects/spring-boot>, pristupljeno: 21.10.2020.
- [9] Introducing React Hooks, <https://reactjs.org/docs/hooks-intro.html>, pristupljeno: 21.10.2020.
- [10] Leangen, GraphQL SPQR Github repozitorijum, <https://github.com/leangen/graphql-spqr>, pristupljeno: 21.10.2020.
- [11] The JUnit Team, JUnit 5, <https://junit.org/junit5/>, pristupljeno: 21.10.2020.
- [12] Mockito, Mocking framework for unit tests in Java, <https://site.mockito.org/>, pristupljeno: 21.10.2020.
- [13] JestJS, www.jestjs.io/, pristupljeno: 21.10.2020.
- [14] Facebook Inc., React Test Renderer, <https://reactjs.org/docs/test-renderer.html>

Kratka biografija:



Aleksandra Milivojević rođena je 5. avgusta 1994. godine u Lazarevcu. Završila je Gimnaziju „Jovan Jovanović Zmaj“ u Novom Sadu.

Zvanje diplomirani inženjer elektrotehnike i računarstva stiče 13.10.2017. godine, sa odbranom diplomskog rada na temu „Kompleksnost lozinke korisničkog naloga na Windows operativnim sistemima“. Iste godine upisuje master akademske studije na smeru Softversko inženjerstvo i informacione tehnologije, odsek Elektronsko poslovanje.

SOLID RADNI OKVIR**SOLID FRAMEWORK**Ivana Marković, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu su predstavljene teorijske osnove Solid radnog okvira. Implementirana je aplikacija za čuvanje bilješki kao studija slučaja. U razvoju aplikacije korišten je React okvir za razvoj korisničkog interfejsa.

Ključne reči: Solid, veb aplikacije, decentralizovani sistem, pod

Abstract – This paper presents the theoretical foundations of the Solid framework. An application for keeping notes is implemented as case study. The React framework was used for user interface development.

Keywords: Solid, web applications, decentralized system, pod

1. UVOD

Ovaj rad za cilj ima predstavljanje Solid (eng. *Social Linked Data*) radnog okvira, odnosno decentralizovanu platformu za veb aplikacije.

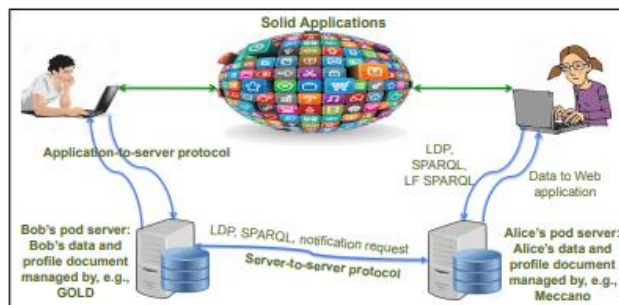
Unutar Solid radnog okvira, korisnikovim podacima se upravlja nezavisno od aplikacije koja ih kreira i koristi. Podaci korisnika se čuvaju u ličnom skladištu podataka dostupnom na vebu (Solid pod). Solid omogućava korisnicima da posjeduju više različitih Solid pod-ova različitih dobavljača pod-ova, dok je istovremeno omogućeno lako prebacivanje sa jednog dobavljača na drugi. Solid protokoli se zasnivaju na postojećim W3C preporukama za čitanje, pisanje i kontrolu pristupa sadržaju na korisnikovom pod-u.

U Solid arhitekturi, aplikacije mogu da pristupaju podacima, koji se nalaze u vlasništvu korisnika ili korisnik ima mogućnost pristupa podacima bez obzira na njihovu lokaciju na vebu. Takođe, korisnici mogu da kontrolišu pristup svojim podacima i imaju mogućnost prebacivanja između aplikacija u bilo kom trenutku. Ovo su neki od najznačajnijih koncepata Solid radnog okvira, čiji će detaljniji opis funkcionisanja biti prikazan u nastavku rada. Na slici 1. je predstavljen uopšten prikaz funkcionisanja Solid radnog okvira.

Prikazana su dva korisnika Solid aplikacija, Alice i Bob, koji koriste različite pod servere. Iako korisnici koriste različite pod servere, oba korisnika mogu koristiti iste aplikacije za pristup i održavanje svojih podataka.

NAPOMENA:

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



Slika 1. Uopšten prikaz Solid radnog okvira [1]

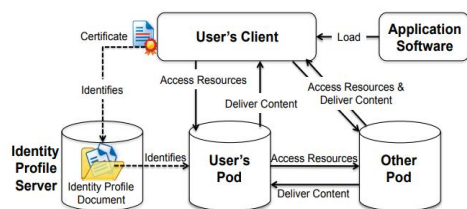
2. SOLID

Solid se zasniva na RDF i tehnologiji semantičkog veba pružajući nezavisnost podataka i omogućavajući jednostavan, ali moćan mehanizam upravljanja podacima. U okviru Solid platforme, svaki korisnik čuva svoje podatke unutar skladišnog prostora dostupnom na mreži, koji se naziva lična mrežna baza podataka (Solid pod). Solid pod-ovi su servisi za skladištenje podataka dostupni na vebu, koji mogu biti dostupni na ličnim serverima samih korisnika ili na javnim serverima dobavljača pod-ova, koji su slični trenutnim dobavljačima skladišta podataka čuvanim na cloud-u (npr. Dropbox).

Korisnici mogu da imaju više od jednog pod-a. Korisnik može da bira između različitih dobavljača pod-ova, pošto Solid aplikacije mogu da rade sa bilo kojim pod serverom, bez obzira na njegovu lokaciju ili usluge dobavljača. Različiti dobavljači pod-ova mogu ponuditi različite stepene privatnosti i pouzdanosti (npr. garancije dostupnosti i kašnjenja) ili pravnu zaštitu (npr. pravni okvir specifičan za zemlju porijekla pod-a).

Solid aplikacije su implementirane kao klijentske veb ili mobilne aplikacije, koje čitaju i zapisuju podatke direktno unutar pod-a. Solid omogućava lak razvoj i korištenje aplikacija, jer su podaci aplikacija uvijek dostupni na vebu, u skladu sa mehanizmom za upravljanje kontrolom pristupa. Aplikacija može objediniti podatke iz različitih izvora na internetu, pristupajući pod-u korisnika koji je pokrenuo aplikaciju i pod-vima koji pripadaju drugim korisnicima. Solid radni okvir omogućava da više aplikacija koriste iste podatke na pod-u. Korisnici mogu birati između različitih aplikacija, koje pružaju sličnu funkcionalnost. Ukoliko se korisnik prebaci na novu aplikaciju, ta aplikacija može pristupiti postojećim podacima korisnika budući da su po dizajnu sistema aplikacije odvojene od podataka koje koriste.

Na slici br.2 prikazana je arhitektura Solid radnog okvira.



Slika 2. Arhitektura Solid radnog okvira [2]

U Solid radnom okviru se podacima upravlja po REST principima, kako je definisano LDP (Linked Data Platform) preporukama [3]. Nove stavke podataka kreiraju se u kontejneru (koji bi se mogli nazvati kolekcijom ili direktorijumom), slanjem podataka na URL kontejnera sa HTTP POST zahtjevom ili izdavanjem HTTP PUT zahtjeva unutar njegovog URL prostora. Pojedinačne stavke podataka se ažuriraju pomoću HTTP PUT i HTTP PATCH zahtjeva, dok se podaci uklanjaju sa korisnikovog pod-a slanjem HTTP DELETE zahtjeva. Uz pomoć HTTP GET zahtjeva se dobija kolekcija podataka, koji se nalaze unutar kontejnera korisnikovog pod-a.

Podaci aplikacije se u Solid-u čuvaju u dokumentima, koji su identifikovani od strane jedinstvenog identifikatora resursa (URI). Solid razlikuje strukturirane podatke predstavljene pomoću okvira za opis resursa (*Resource Description Framework - RDF*) i nestrukturirane podatke bilo koje vrste (npr. video snimci, slike, veb stranice). Ovo omogućuje strukturiranim podacima da budu parsirani i serijalizovani u različitim sintaksama poput Turtle, JSON ili RDFa. RDF slijedi REST principe u kojima resursi imaju pojedinačne URI-je.

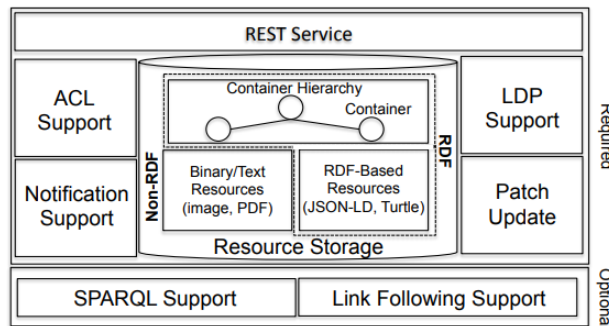
Solid aplikacije ne zahtijevaju potvrdu identiteta kao što to radi većina uobičajenih aplikacija, jer korisnik ne treba da potvrdi identitet kod dobavljača aplikacije. Umjesto toga, Solid aplikacije bi mogle da iznude „lažne“ zahtjeve za potvrdu identiteta kako bi pribavile WebID korisnika iz sertifikata klijenta. Iz tog razloga, autentifikacija se izvodi samo između klijenta (browser-a) i korisnikovog pod-a, gdje su podaci smješteni. Ovakav način autentifikacije uklanja potrebu korisnika da unosi WebID prilikom procesa prijavljivanja.

Da bi Solid funkcionisao na zaista decentralizovan način, neophodan je identitetski prostor u kojem korisnici mogu lako da upravljaju i proširuju sopstveni identitet i akreditive. Postojeći protokoli identifikacije, kao što su OAuth [4] i OpenID Connect [5] nude određeni stepen decentralizacije, ali se ipak ne uklapaju u strukturu podataka profila zasnovanih na RDF-u, što otežava proširivanje informacija o profilu dodatnim korisničkim atributima. Solid koristi WebID pri implementaciji globalne arhitekture upravljanja identitetom, zasnovanom na konceptu decentralizovanog dobavljača identiteta, što u kombinaciji sa WebID-TLS [6] omogućava jednokratnu prijavu na vebu.

Protokol WebID-TLS je decentralizovani protokol za potvrdu identiteta, koji omogućava sigurnu i efikasnu potvrdu identiteta na vebu. Protokol omogućava korisnicima da se autentifikuju na bilo kom sajtu, jednostavnim odabirom jednog od predloženih korisnikovih sertifikata. Sertifikate može kreirati bilo koja veb stranica za svoje korisnike. Za razliku od klasične autentifikacije sertifikata korisnika,

koja se oslanja na Public Key Infrastructure (PKI), WebID-TLS ne zahtijeva potpisivanje sertifikata od strane pouzdanog autoriteta za izdavanje sertifikata. Razlog tome je što WebID-TLS koristi korisnikov sertifikat kao sredstvo za obavljanje autentifikacije javnim ključem.

Na slici 3 predstavljena je arhitektura pod-a. U okviru pod-a je moguće čuvati RDF i ne-RDF resurse. Pod serveri podržavaju LDP, patch resursa, kontrolu pristupa, live ažuriranje i opciono SPARQL.



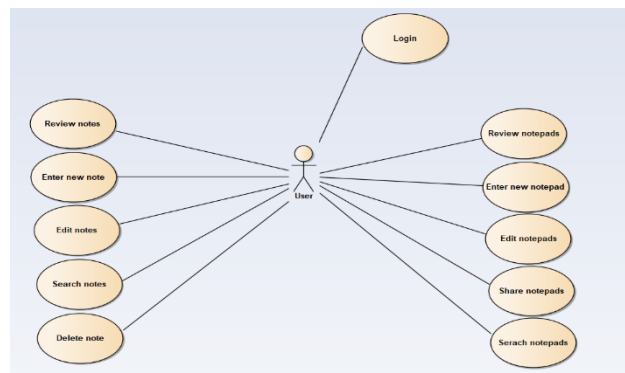
Slika 3. Arhitektura pod-a [2]

3. SPECIFIKACIJA

U ovom odeljku detaljno je opisana specifikacija aplikacije za čuvanje bilješki, koja je u ovom radu korištena kao studija slučaja. Predstavljene su funkcionalni zahtjevi koje je bilo neophodno implementirati.

3.1. Funkcionalni zahtjevi

U ovom odjeljku su opisane funkcionalnosti koje je neophodno podržati u razvoju aplikacije. Na slici 4 su predstavljeni funkcionalni zahtjevi UML dijagramom slučajeva korištenja.

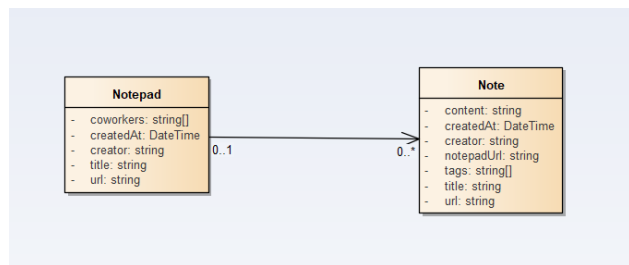


Slika 4. UML dijagram slučajeva korištenja Notepad aplikacije

Jedan od osnovnih zahtjeva, koje je neophodno podržati jeste prijava na sistem. U okviru ove aplikacije, prijava je nešto drugačija u odnosu na standardnu prijavu veb aplikacija. Prijava obuhvata unos WebID-ija korisnika, odnosno URL-a na kome se mogu pronaći informacije o korisniku i kako bi sama aplikacija znala odakle da čita i piše podatke. Neophodno je omogućiti korisniku prikaz svih bilježnica i njihovih bilješki, dodavanje novih, kao i editovanje i brisanje postojećih bilješki i bilježnica. Takođe, korisnik ima mogućnost da svoju bilježnicu sa drugim korisnicima, odnosno da omogućiti drugim korisnicima pristup njegovom pod-u i mogućnost da drugi korisnici izvršavaju izmjene nad datoj bilježnici.

3.2. Model podataka

U okviru aplikacije se razlikuju dva tipa entiteta, jedan je bilježnica, a drugi je bilješka. Na slici 5 je predstavljen model podataka.



Slika 5. Model podataka

4. IMPLEMENTACIJA

U ovom odeljku je prikazan detaljan opis implementacije Notepad aplikacije, čija je specifikacija prikazana u prethodnom poglavlju. Prilikom implementacije aplikacije za čuvanje bilješki, nazvane Notepad, korišten je React okvir za realizaciju korisničkog interfejsa aplikacije.

Pri realizaciji konekcije ka Solid-u, korištena je biblioteka „solid/react“ iz koje je uključena *LoginButton* klasa koja definiše izgled pop-up prozora u kojem je neophodno unijeti korisnikov WebID, radi dalje autentifikacije i prijave na sistem. Nakon unosa WebID-ija korisnik je u obavezi da unese kredencijale definisane prikom kreiranja pod-a, kako bi sistem dozvolio interakciju sa korisnikovim pod-om. Nakon unosa korisničkog imena i lozinke, neophodno je izvršiti autentifikaciju korisnika. Za autentifikaciju korisnika u ovom projektu je korišćena „solid-auth-client“ biblioteka.

Prilikom realizacije postupka prijave na sistem se može uočiti koncept decentralizacije Solid radnog okvira. U toku implementacije Notepad aplikacije, nije bilo potreba za razvojem sopstvenog mehanizma autentifikacije korisnika, nego su iskorištene dostupne Solid biblioteke za tu svrhu.

Ukoliko korisnik prvi put pokreće aplikaciju, nakon uspješne autentifikacije i prijave na sistem, neophodno je da dozvoli aplikaciji da pristupa njegovom pod-u. Na taj način je obezbjeđeno da korisnik ima potpunu kontrolu nad svojim pod-om, imajući mogućnost da definiše koje akcije aplikacija može izvršavati nad njegovim pod-om.

U okviru ovog projekta je definisana klasa *SolidBackend*, čija je osnovna namjena pristup i izvršavanje svih akcija nad korisnikovim pod-om. Biblioteka „rdflib“ je iskorištena za rad sa podacima čuvanim u okviru Solid okvira. Biblioteka „rdflib“ predstavlja opšti skup alata za upravljanje podacima u RDF formatu.

Omogućava skladištenje podataka, njihovo parsiranje i serijalizaciju u različite formate, ali takođe može da prati i promjene podataka koje dolaze iz aplikacije ili sa servera. Postoji nekoliko klasa „rdflib“ biblioteke, koje su korišteni u okviru *SolidBackend* klase, i njihovo značenje je navedeno u nastavku:

- *Store* – predstavlja strukturu podataka za čuvanje podataka u obliku grafa i omogućava izvršavanje upita nad podacima

- *Fetcher* – pomoćna klasa koji se koristi za povezivanje na veb, učitavanje podataka kao i njihovo čuvanje nazad
- *UpdateManager* – veoma moćna i korisna klasa koja omogućava slanje malih promjena serveru kako bi se podaci ispravili dok korisnik mijenja podatke u realnom vremenu i
- *Graph* – predstavlja bazu podataka za semantički veb

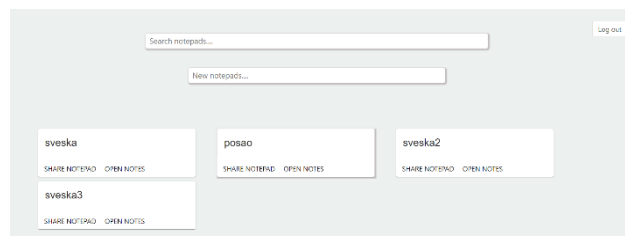
Prilikom implementacije funkcionalnosti dijeljenja bilježnica između korisnika neophodno je vršiti i izmjene nad .acl fajlovima bilježnica. U svrhu pristupa .acl fajlovima i njihovom editovanju iskorištena je biblioteka „solid-acl-utils“, to je JavaScript biblioteka za rad sa .acl fajlovima, koja omogućava dodavanje, mijenjanje i uklanjanje dozvola za pristup datotekama i pod-ovima korisnika.

5. DEMONSTRACIJA

U ovom odjeljku je predstavljen rad sa Notepad aplikacijom, s tim da nije stavljen akcenat na načinu implementacije funkcionalnih zahtijeva, nego na prikazu njihove realizacije, kao i na prikazu promjena nad korisnikovim pod-om.

Prilikom demonstracije rada aplikacije su korištena dva Solid pod naloga, jedan sa WebID-ijem <https://testic.inrupt.net/>, a drugi sa <https://ivanam96.inrupt.net/> WebID-ijem. Unutar oba profila korisnika je definisana *friends* kategorija, u čijem sadržaju se nalazi WebID drugog naloga.

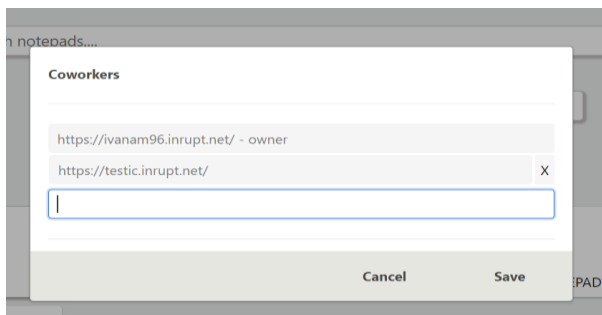
Demonstracija rada aplikacije počinje logovanjem korisnika WebID-ija <https://testic.inrupt.net/>. Na slici 6 je prikazan ekran korisnika nakon logovanja.



Slika 6. Prikaz ekrana nakon logovanja

Sve bilježnice i bilješke korisnika se čuvaju u *public/solidapp/* folderu, u odvojenim folderima za bilježnice notepads, a bilješke u folderu notes.

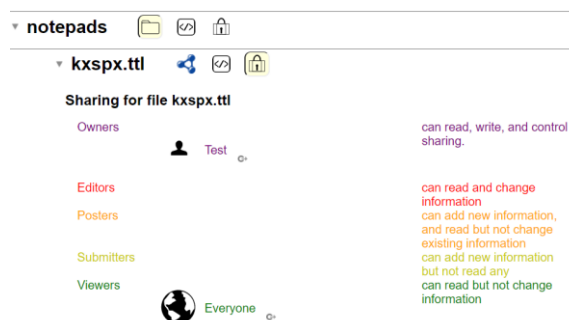
Korisnik u svakom trenutku može provjeriti i pregledati podatke sačuvane na svom pod-u, logovanjem na svoj profil unutar pod dobavljača (u ovom slučaju *inrupt.net*). Proširivanjem notepads i notes foldera su prikazani .ttl fajlovi sa nasumično kreiranim nazivima, koji u sebi sadrže informacije o svim bilježnicama i bilješkama korisnika. U toku demonstracije rada aplikacije je izvršeno dodavanje nove bilježnice nad kojom su kasnije izvršene određene izmjene. Korisniku je omogućeno editovanje svih polja unutar bilježnice, međutim kao posebna funkcionalnost izdvojeno je dijeljenje bilježnica između različitih korisnika. Na slici 7 je predstavljen pop-up prozor unutar kog se nalaze elementi korisničkog interfejsa za datu funkcionalnost.



Slika 7. Prikaz prozora za editovanje liste saradnika

Korisnik ima mogućnost dodavanja WebID-ija novog saradnika, kao i brisanje WebID saradnika iz liste klikom na dugme *X*, koje se nalazi u istom redu kao i dati WebID saradnika. Nakon unosa željenog WebID-ija novog saradnika i klika na taster *Enter*, WebID saradnika se dodaje u listu saradnika i prikaz se osvježava sa novom vrijednošću liste. Međutim, kako bi izmjene nad datom listom bile sačuvane i projektovane nad korisnikovom pod-u, neophodno je da korisnik klikne na dugme *Save*. Ukoliko korisnik želi odustati od unijetih izmjena ili jednostavno želi izaći iz datog prozora dovoljan je klik na dugme *Cancel*.

Takođe, je prikazan i rad sa .acl fajlovima bilježnica. Dati fajlovi definišu pravo pristupa podacima svakog pojedinačnog resursa na korisnikovom pod-u. Na slici 8 je dat prikazan izgled .acl fajla bilježnice.



Slika 8. Prikaz .acl fajla bilježnice

6. ZAKLJUČAK

U ovom radu su predstavljene teorijske osnove Solid radnog okvira, decentralizovane platforme za razvoj veb aplikacija. Pored teorijskog opisa, predstavljen je način implementacije Notepod aplikacije za čuvanje bilješki. Ova aplikacije je poslužila kao studija slučaja pri analizi Solid radnog okvira. Cilj ovog rada je predstavljanje osnovnih teorijskih aspekata Solid-a, kao i njihova kasnija primjena.

Notepod aplikacija predstavlja jednostavnu aplikaciju za unos bilješki čiji model podataka je prost. Ali bez obzira na to, kroz implementaciju date aplikacije je uspješno implementirano dosta bitnih mogućnosti Solid radnog okvira. Obezbijedena je decentralizovana autentifikacija korisnika aplikacije, kao jednog od osnovnih načela Solid-a.

Jedan od najbitnijih funkcionalnih zahtjeva Notepod aplikacije jeste mogućnost dijeljenja bilježnica između korisnika. Kroz implementaciju ovog funkcionalnog zahtjeva prikazan je i objašnjen način funkcionisanja Solid-a prilikom definisanja prava pristupa kako korisnikovom pod-u, tako i pojedinačnim fajlovima sačuvanim u korisničkom skladištu.

Na internetu kakav je danas, korisnik otprema svoje podatke svakom servisu koji koristi. Svi podaci koje korisnik postavi na svoj Facebook nalog se čuvaju u jednom silosu, svi podaci koje postavi na Twitter se čuvaju u drugom silosu itd.. Koncept Solid-a je drugačiji. Umjesto otpremanja podataka na udaljene servere, nego korisnik dozvoljava pristup podacima koji se nalaze unutar jednog od korisnikovih pod-ova servisu sa kojim želi ostvariti interakciju. Sveobuhvatni cilj je eliminisanje uticaja velikih korporacija koje imaju nad korisničkim podacima i negativnih posljedica koje iz toga proizilaze i da se korisnicima pruži konačna odlika o tome ko pristupa njihovim podacima. Međutim, u praksi postoji nekoliko prepreka koje se moraju prevazići da bi se to postiglo i one zahtijevaju velike strukturne promjene u načinu svakodnevnog korištenja interneta.

7. LITERATURA

- [1] E. Mansour, A. Vlad Samba, S. Hawke, M. Zereba, S. Capadisli, A. Ghanem, A. Aboulmaga and T. Berners-Lee, „A demonstration of the Solid platform for Social Web applications“ <http://emansour.com/publications/paper/www16-solid-essam.pdf>. Novembar, 2020.
- [2] E. Mansour, A. Vlad Samba, S. Hawke, M. Zereba, S. Capadisli, A. Ghanem, A. Aboulmaga and T. Berners-Lee, „A Platform for Decentralized Social Applications Based on Linked Data“, http://emansour.com/research/lusail/solid_protocols.pdf. Novembar, 2020.
- [3] A. Malhotra, J. Arwe, and S. Speicher. „Linked Data Platform Specification. W3C Recommendation“, <http://www.w3.org/TR/ldp/>. Novembar, 2020.
- [4] D. Hardt. The OAuth 2.0 authorization framework.
- [5] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, and C. Mortimore. Openid connect core 1.0. The OpenID Foundation.
- [6] T. Inkster, H. Story, and B. Harbulot., WebID-TLS Specification. „<http://www.w3.org/2005/Incubator/webid/spec/tls/>“. Novembar, 2020.

Kratka biografija:



Ivana Marković je rođena 02.08.1996. godine u Bijeljini. Školske 2015/2016. godine upisuje Fakultet tehničkih nauka u Novom Sadu, smjer „Primenjeno softversko inženjerstvo“. Diplomirala 2019. godine. Upisuje master studije na istom fakultetu, smjer „Računarstvo I automatika“. kontakt: markovic.ivana996@gmail.com

ПРИМЕРИ ПРИМЕНЕ SCADA СИСТЕМА У ИНДУСТРИЈИ НАФТЕ И ГАСА**SCADA SYSTEM IMPLEMENTATION IN OIL AND GAS INDUSTRY**Немања Лазић, Владо Поробић, *Факултет техничких наука, Нови Сад***Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО**

Кратак садржај – У овом раду представљени су различити начини интеграције комуникационе и управљачке опреме, као и различите методе прикупљања, складиштења и визуелизације процесних величина у SCADA систему. Као софтверска платформа за интеграцију овог система искоришћена је AVEVA (Wonderware) System Platform-a, са својим серверским компонентама и клијентским апликацијама.

Кључне речи: SCADA, HMI, рачунарске мреже, MODBUS, OPC, даљински надзор

Abstract – This paper presents various methods of communication and control equipment integration, as well as different methods of collecting, storage and visualisation of process values into the SCADA system. System Platform by AVEVA (Wonderware) was used as the software platform for integration of this system, with its server components and client applications.

Keywords: SCADA, HMI, network communication, MODBUS, OPC, remote monitoring

1. УВОД

Захваљујући наглом развоју индустријске аутоматике и унапређења управљања системима последњих десетак година многе сфере попут водопривреде, енергетике, пољопривреде, нафтне индустрије и других области имају потребу за технологијом која омогућава даљинско мерење и надзор, као и пренос информација са удаљених локација до оператора.

Да би се на нивоу комуникационог система остварила поуздана и ефикасна комуникација између елемената SCADA система, различитих по начину функционисања, неопходно је постојање одговарајуће мрежне архитектуре, као и постојање комуникационих протокола који би стандардизовали размену података.

2. КОМУНИКАЦИОНЕ (РАЧУНАРСKE) МРЕЖЕ

Рачунарска мрежа представља два или више рачунара (мрежна уређаја) који су међусобно повезани неким медијумом (ваздух, кабл) и користе различите мрежне протоколе за међусобну комуникацију и размену података. Хардверски део рачунарске мреже чине мрежни чворови (*nodes*) чија улога је обрада или усмеравање података.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Владо Поробић, ванр. проф.

Подела рачунарских мрежа остварена је на основу различитих параметара:

- 1) Према области покривености: LAN, MAN, WAN
- 2) Према типу линијске мреже између чворова: Point-to-Point, Point-to-Multipoint, Mesh
- 3) Према распореду чворова у мрежи: Звезда (Star), Стабло (Tree), Прстен (Ring), Магистрала (Bus)
- 4) Према начину на који се врши пренос података дуж комуникационих путева: Broadcast, Multicast, Unicast, Anycast
- 5) Према смеру преноса података између повезаних уређаја: Simplex, Half-Duplex, Full-Duplex

Мрежни протокол је скуп стандардних правила за приказ и сигнализацију података, као и за проверу грешака коју је потребно извршити да би се податак уопште послао.

Протокол модел је заједнички показатељ функција које одређени протоколи у колекцији (група протокола) поседују [1].

Два основна протокол модела су TCP/IP и OSI протокол модели.

Табела 1. Поређење слојева OSI и TCP/IP референтног модела

OSI	TCP/IP
7. Апликацијски ниво	4. Апликацијски ниво
6. Презентациони ниво	
5. Сесиони ниво	
4. Транспортни ниво	3. Транспортни ниво
3. Мрежни ниво	2. Мрежни ниво
2. Ниво вода података	1. Приступни ниво
1. Физички ниво	

3. БЕЖИЧНЕ МРЕЖЕ - WIFI И ЦЕЛУЛАРНА КОМУНИКАЦИЈА

Бежични медијум подразумева пренос података без физичке конекције између терминалних уређаја тј. пренос података преко радио таласа.

Под мобилним комуникационим системима се подразумевају радио системи у којима се остварује веза између корисника од којих се најмање један креће или је заустављен на унапред непознатој локацији [2].

4. SCADA СИСТЕМИ

Аквизиционо управљачки системи су системи за управљање индустријским процесима. То је заправо скуп наменских, просторно дистрибуираних, међусобно повезаних рачунарских модула, чији је заједнички циљ остварење функција надзора и/или управљања физичким процесом у реалном времену.

Захтеви који се постављају пред ове системе су:

- 1) Рад у реалном времену
- 2) Дистрибуција рачунарских ресурса
- 3) Постизање максималне поузданости и расположивости

Имплементација ових система се врши са циљем смањења трошкова производње, повећања отпорности на грешке, могућности лакшег проширивања истог (флексибилност базирана на модуларности хардверских и софтверских елемената), повећања поузданости и расположивости, као и побољшања перформанси попут времена одзива и пропусности система.

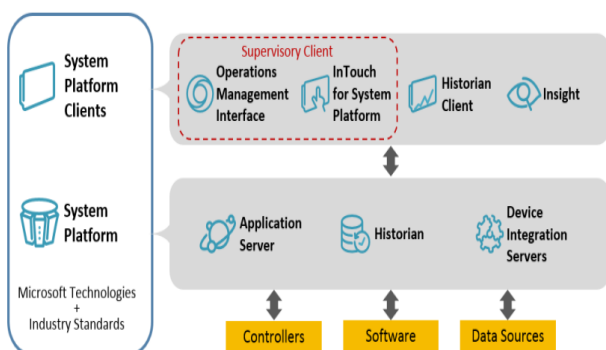
SCADA је централни чвор аквизиционо управљачког система чија је улога обједињавање и обрада свих процесних података, комуникација са оператерима (корисницима), извршавање управљачких акција највишег нивоа и комуникација са пословним информационом окружењем [3].

5. AVEVA (WONDERWARE) SCADA СИСТЕМ

AVEVA System Platform-а је индустријско софтверско решење базирано на *Archestra* технологији, која представља објектно-оријентисану платформу. Садржи интегрисан сет сервиса и имплементиран систем за прикупљање и манипулацију подацима, а све у циљу управљања аутоматизованим индустријским процесом.

Заједно са својим клијентским апликацијама System Platform-а омогућава *framework* и софтверске алате за развој, извршавање, мониторинг и визуелизацију апликација.

Сервиси за прикупљање података, хисторизацију и алармирање су компоненте које долазе са System Platform-ом, док су сервиси попут визуелизације и трендовања, клијентски.



Слика 1. Компоненте System Platform-е [4]

6. OPC И MODBUS КОМУНИКАЦИЈА

MODBUS је *master/slave* протокол са *half-duplex* преносом заснован на клијент-сервер архитектури,

који припада групи *Fieldbus* протокола. Са стране аквизиционо-управљачких система PLC, DCS или RTU има улогу сервера (*slave*), који на захтев (упит) SCADA клијента (*master*) шаље поруке у виду жељених порука [5].

MODBUS има 2 основна типа преноса података:

- 1) Серијски:
 - MODBUS ASCII – Порука се састоји од ASCII карактера
 - MODBUS RTU – Порука се састоји од бинарних бајтова
- 2) MODBUS TCP/IP - Интернет протокол где *master* (клијент) шаље поруку до специфициране IP адресе и порта. Сваки *slave* уређај (сервер) има своју специфичну IP адресу. По конвенцији многи *slave* уређаји користе Ethernet порт 502 као стандард. Код MODBUS TCP-а порука мора имати IP дестинациону адресу, и преноси се путем *Ethernet-a*.

OPC је клијент-сервер архитектура. Клијент иницира комуникацију са сервером, док сервер извршава наредбе инициране од клијента, односно уписује/чита вредности само за специфициране величине из контролера и то у временском интервалу захтеваном од стране клијента.

Један од највећих изазова у индустријској аутоматизи била је комуникација између различитих брендова произвођача комуникационо-управљачких уређаја (PLC, DCS, RTU, паметни сензори, драјвери итд...) као и између тих уређаја и софтвера.

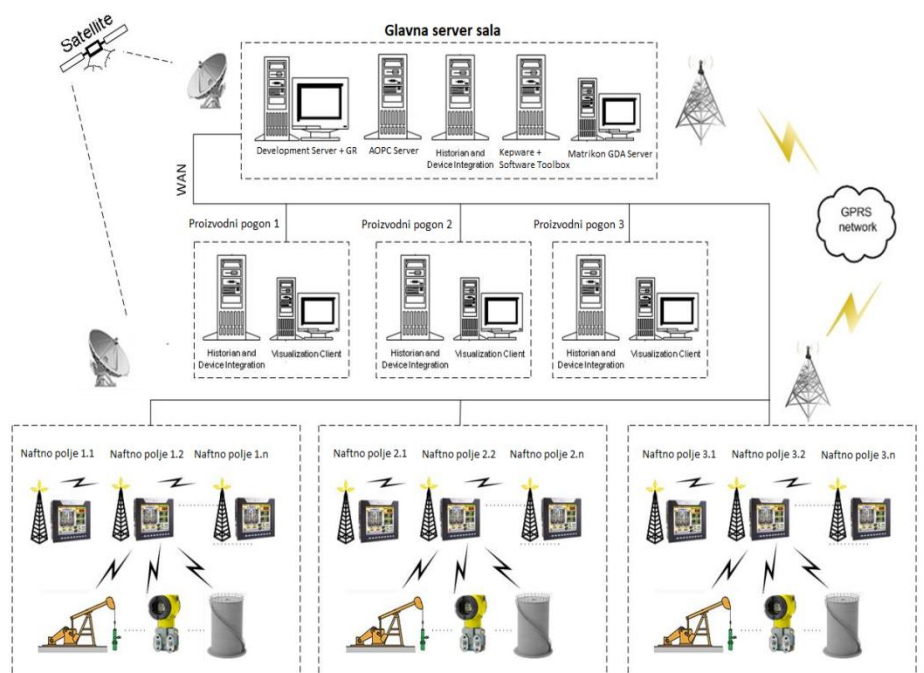
Иако већина контролера има имплементран метод комуникације, они не користе исте комуникационе протоколе, као ни исти тип очичења приликом остваривања серијске или *Ethernet* комуникације.

Те разлике доводе до губитака података и немогућности остваривања жељених конекција. Резултат појављивања OPC стандарда је то што иако различити произвођачи нуде различите OPC сервере, SCADA-и (одређеном произвођачу) је довољан само један OPC клијент.

7. УПРОШТЕН МОДЕЛ SCADA СИСТЕМА ЗА ИНТЕГРАЦИЈУ РАЗЛИЧИТИХ ПРОЦЕСА И ФИЗИЧКИХ УРЕЂАЈА У ИНДУСТРИЈИ НАФТЕ И ГАСА

На слици 2 приказан је упроштени модел SCADA система у коме је имплементиран процес експлоатације нафте и гаса, и у којем је интегрисана различита комуникациона опрема у јединствен систем даљинског надзора, алармирања и извештавања.

У условима велике географске дистрибуираности, постављена је хибридна комуникациона структура, која укључује изнајмљене линије (L3VPN), WiFi, GPRS (HSPA), и сателитске линкове у коју су интегрисане бушотине у различитим методама експлоатације, дизел електрични агрегати и мерила потрошње електричне енергије.



Слика 2. Архитектура SCADA система

У најнижем слоју ове хијерархије налазе се нафтна поља. На нафтним пољима се налазе сви објекти и комуникациони уређаји који се телеметришу у SCADA систем (бушотине, дизел агрегати, мерила потрошње ел. енергије, сва комуникациона опрема итд.). На сваком нафтном пољу инсталирана је SCADA HMI клијентска апликација која омогућава мониторинг индустријског процеса који се одвија на истом.

Следећи слој хијерархије чине 3 рачунара за три производна погона. Они имају сличну улогу као и рачунари на нафтним пољима само што се SCADA HMI клијентске апликације састоје од објеката који се налазе на свим нафтним пољима који припадају једном погону.

У самом врху хијерархије налази се серверска сала коју чини кластер сервера од којих сваки има различиту улогу у складу са имплементираним архитектуром.

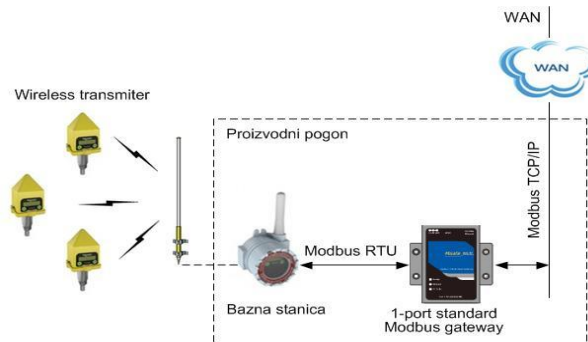
9. ИНТЕГРАЦИЈА РАЗЛИЧИТИХ ОБЈЕКТА У SCADA СИСТЕМ

Како би се прикупили подаци од интереса (процесне величине), односно како би се интегрисали објекти у пољу у SCADA систему, постоји небројено много могућности, што од избора контролера, комуникационих уређаја, па до паметних мерила и сензора.

9.1. Интеграција еруптивних бушотина (ER метода)

Најчешће примењени метод интеграције бушотина у еруптивној методи рада је интеграција применом мреже бежичних сензора (WSN).

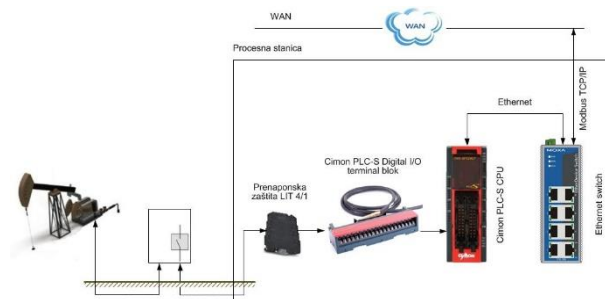
Реализовани систем се састоји из сензора притиска и базне станице. Једна станица бежично комуницира са до 100 сензора и информацију о притисцима шаље у свој адресни простор, коме се приступа преко MODBUS RTU протокола.



Слика 3. Интеграција бушотина WSN системом

9.2. Интеграција дубинских пумпи (DP метода)

Интеграција бушотина у DP методи реализује се најчешће на два начина која су проистакла из чињенице да је један део третираних бушотина повезан са процесном станицом постојећим сигналним каблом, док остале то нису и услед дистрибуираности као решење искориштен је GPRS (HSPA) комуникациони уређај.

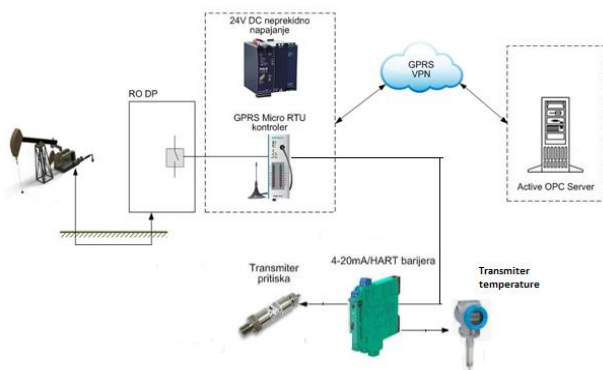


Слика 4. Интеграција бушотина преко Cimon PLC-a

Информација о основном релевантном податку (тренутни статус пумпе - On/Off) се са слободног контакта моторне склопке или помоћног релеја, постојећим сигналним каблом, уводи у концентратор података (CIMON PLC), који је кроз локалну мрежну

инфраструктуру повезан са SCADA апликацијом, која истом приступа по *MODBUS TCP/IP* протоколу.

Са бушотинама које нису сигналним каблом повезане са процесном станицом линк је успостављан посредством комуникационог уређаја *MOXA ioLogik W5340* или *MOXA ioLogik 2542-HSPA/GPRS*.

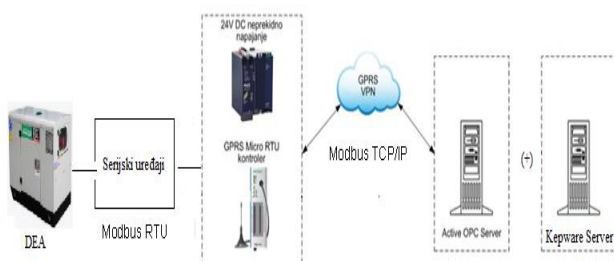


Слика 5. Интеграција бушотина GPRS (HSPA) линком

Уређаји подржавају комуникацију са надзорним центром по *MODBUS TCP/IP* протоколу, са опцијом *report-by-event*. Променом статуса дигиталних, аналогних и релејних сигнала, иницира се слање информације SCADA апликацији. Са друге стране, SCADA, у улози *master*-а, периодично проверава присуство свих својих *slave* уређаја. На сличан начин са другим сетом података телеметришу се и DEA (9.3).

9.3. Интеграција дизел електричних агрегата (DEA)

Дизел електрични агрегати веома су важан енергетски извор напајања у индустријским процесима где су опрема и уређаји за њену телеметрију распоређени на великим географским подручјима, и на локацијама где једноставно не постоји други начин да се спроведе напајање електричном енергијом.

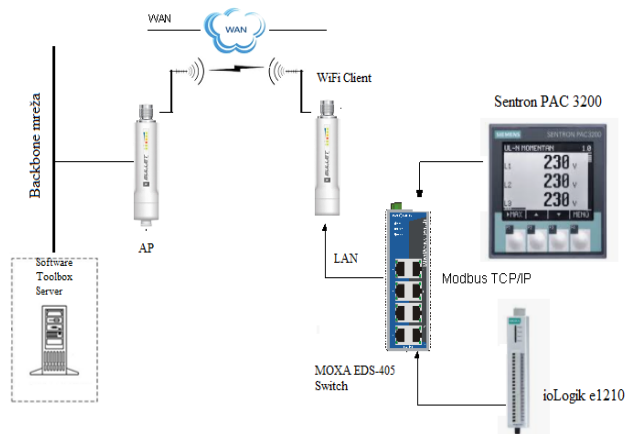


Слика 6. Интеграција DEA GPRS (HSPA) линком

9.4. Интеграција мерила потрошње ел. енергије

Комуникација између SCADA апликације и мерила потрошње електричне енергије најчешће је успостављена помоћу два *Ubiquity Network* уређаја, где се један уређај користи као AP (*Access Point*), и повезан је на *backbone* мрежу (*L3VPN*), а други као *WiFi Client*, који је повезан на AP, и излаз на мрежу има преко њега. За AP се користи уређај *Bullet M2*, а за клијента *Nanostation M2*, са тим што је је могла

бити и обрнута ситуација, јер оба уређаја подржавају оба *WiFi* мода.



Слика 7. Интеграција мерила потрошње ел. енергије *WiFi* линком

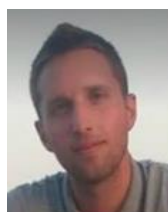
10. ЗАКЉУЧАК

Циљ овог рада био је да прикаже једно модерно решење даљинског надзора и управљања у једном већем систему које задовољава данашње захтеве развоја привреде као што су флексибилност, ефикасност, поузданост, сигурност и прецизност.

11. ЛИТЕРАТУРА

- [1] LINK group, Računarske mreže, skripta sa predavanja
- [2] <http://elektronika.elfak.ni.ac.rs/index.php/subjects/27-osnovne-akademske-studije/84-racunarske-mreze-i-interfejsi>
- [3] <http://www.esi.ftn.uns.ac.rs/images/predmeti/SwKO/predavanja/Softver%20sa%20kritcnim%20odzivom%20-Predavanja%201.pdf>
- [4] Application Server 2017 Update 3 - Training Manual, AVEVA, 2019.
- [5] Igor Tot, "Industrijska implementacija modbus protokola", Fakultet tehničkih nauka, Novi Sad, 2019.

Кратка биографија



Немања Лазих рођен је 20.06.1993. године у Зрењанину. Дипломирао је у октобру 2017. године на Факултету техничких наука у Новом Саду, смер Енергетика, електроника и телекомуникације – Енергетска електроника и електричне машине.

Проф. др Владо Поробић, запослен је на Факултету техничких наука у Новом Саду, у звању ванредног професора. Предмет интересовања су му управљање претварањима енергетске електронике и *automotive* област.

STRATEGIJE ZA RAD SA NEDOSTAJUĆIM VREDNOSTIMA**STRATEGIES FOR DEALING WITH MISSING VALUES**

Sreten Petrović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – Ovaj rad se bavi predstavljanjem strategija za rad sa nedostajućim vrijednostima i pokazivanjem njihovih prednosti, mana i efikasnosti u kombinaciji sa algoritmima mašinskog učenja prilikom predviđanja popularnosti mobilnih aplikacija.

Ključne reči: strategije, nedostajuće vrijednosti, algoritmi, mašinsko učenje, predikcija, mobilne aplikacije

Abstract – This paper deals with the presentation of strategies for dealing with missing values and showing their advantages, disadvantages and efficiency in combination with machine learning algorithms while predicting the popularity of mobile applications.

Keywords: strategies, missing values, algorithms, machine learning, prediction, mobile applications

1. UVOD

Cilj rada jeste definisanje i podjela strategija za rad sa nedostajućim vrijednostima i navođenje primjera korišćenja istih kako bi se riješio konkretan problem koji će se, takođe, opisati u radu. Izvršiće se podjela nedostajućih vrijednosti na određene tipove sa kratkim pregledom osobina i funkcionalnosti. Upotrebom različitih metoda i tehnika za analizu, razumijevanje i sređivanje skupa podataka, formiraće se skup podataka nad kojim će se primjenjivati opisane strategije za rad sa nedostajućim vrijednostima. Kratak pregled algoritama mašinskog učenja, koji će se koristiti u kombinaciji sa strategijama, će biti predstavljen. Pregled implementacije, analiza dobijenih rezultata i diskusija o efikasnosti strategija i algoritama prilikom rješavanja problema biće navedena kako bi se pojasnilo koja kombinacija strategije i algoritma je bila najuspješnija prilikom rješavanja.

2. TEORIJSKI PREGLED PROBLEMA

Problem koji će se analizirati jeste na koji način strategije za rad sa nedostajućim vrijednostima utiču na konkretan projekat, da li je i na koji način primjena strategija dovela do poboljšanja rezultata. Strategije će se primjenjivati za formiranje skupa podataka, a taj skup podataka će služiti kao osnova za predviđanje popularnosti mobilnih aplikacija. Popularnost mobilnih aplikacija će se vršiti na osnovu određenih obilježja aplikacija koja se nalaze u okviru skupa podataka. Prije primjene samih strategija neophodno je prikupiti, analizirati, razumjeti i provjeriti podatke,

kako bi se što bolje definisalo koji podaci su odgovarajući, a koji ne i kako bi se započeo proces njihove obrade.

Organizacija podataka u okviru skupa podataka predstavlja važan preduslov za dalje čišćenje, uklanjanje ili imputaciju nedostajućih podataka. Uzroci zbog čega se vrše navedene manipulacije nad podacima su nedostajuće vrijednosti, pogrešno unijeti podaci, kao i domenski pogrešni podaci.

Prema Rubinu [1] postoji nekoliko tipova nedostajućih vrijednosti podataka:

1. Potpuno slučajno - *Missing completely at random (MCAR)* - predstavlja vjerovatnoću da nedostajući podaci nisu povezani sa određenom vrijednošću koja bi trebala da se dobije, kao ni sa vrijednostima ostalih obilježja
2. Slučajno - *Missing at random (MAR)* - za razliku od *MCAR*-a ovaj tip podataka ima vezu sa vrijednostima ostalih obilježja
3. Neslučajno - *Missing not at random (MNAR)* - ovaj tip podataka je najproblematičniji i jedini način da se dobiju nepristrasne procjene nedostajućih podataka jeste da se napravi odgovarajući model koji bi ih obradio korišćenjem metoda za predviđanje nedostajućih vrijednosti

Rješavanje ovog problema omogućava da se korisniku obezbijedi brz i kvalitetan pristup aplikacijama koje pripadaju kategorijama najpopularnijih aplikacija i koje se poklapaju sa interesnom sferom korisnika.

3. METODOLOGIJA

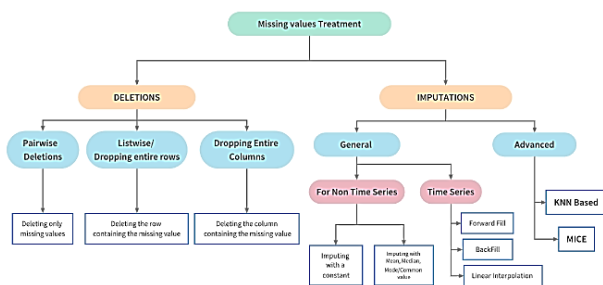
Strategije za rad sa nedostajućim vrijednostima predstavljaju proces pretvaranja i mapiranja neobrađenih podataka u drugu formu sa ciljem da se podaci učine pogodnim za dalju upotrebu i istraživanje. Podjela strategija na određeni broj faza i podfaza, kao i izučavanje i detaljno opisivanje svake pojedinačno, izvršeno je na osnovu literature [3-5].

Strategije za rad sa nedostajućim vrijednostima možemo podijeliti na dvije osnovne grupe, a to su brisanje i imputacija podataka, koje se dalje dijele u odgovarajuće podgrupe.

U nastavku je dat kratak pregled strategija za rad sa nedostajućim vrijednostima, a sumirani pregled prikazan je na slici 1, koja je preuzeta iz literature [2].

NAPOMENA:

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



Slika 1. Dijagram podele strategija za rad sa nedostajućim vrijednostima

3.1 Brisanje podataka

Predstavlja najjednostavniju strategiju za rad sa nedostajućim vrijednostima. Brisanje podataka se može podijeliti na tri strategije, a to su:

1. Brisanje cijelog opažaja (**Listwise deletion**) - strategija za brisanje cijelog reda skupa podataka ukoliko postoji nedostajuća vrijednost u okviru tog reda
2. Brisanje nedostajuće vrijednosti u okviru opažaja (**Pairwise deletion**) - na osnovu korelacionih matrica može se izmjeriti veza između dva obilježja; za svaki par obilježja za koji su podaci dostupni, koeficijent korelacije će te podatke uzeti u obzir
3. Brisanje cijele kolone (**Dropping entire column**) - vrši brisanje kolone ukoliko postoje nedostajuće vrijednosti podataka u kolonama

3.2 Imputacija

Cilj imputacije jeste upotreba poznatih odnosa u postojećim vrijednostima skupa podataka koji bi se koristili u procjeni nedostajućih vrijednosti. Imputacija sprečava gubitak podataka i na taj način se poboljšava tačnost skupova podataka. Može se podijeliti u dvije grupe, a to su opšte i napredne imputacije. Opšta grupa imputacija se dalje dijeli na **Non Time Series** i **Time Series**. **Non time series** se dijeli na **mean**, **median** i **mode** imputacije i na imputaciju sa konstantnom vrijednošću. **Time series** se dijeli na **Last Observation Carried Forward (LOCF)**, **Next Observation Carried Backward (NOCB)** i linearnu interpolaciju. Naprednoj grupi imputacija pripadaju **k** najbližih komšija (**k nearest neighbors** - **k-NN**) i višestruka imputacija korišćenjem ulančanih jednačina (**Multiple imputation by chained equations** - **MICE**).

Mean, **median** i **mode** imputacije imaju za cilj da nedostajuće vrijednosti u okviru jednog obilježja zamijene vrijednošću dobijene na osnovu poznatih vrijednosti tog obilježja, ne uzimajući u obzir odnos sa drugim vrijednostima iz drugih obilježja. Nedostajuća vrijednost se korišćenjem **mean** imputacije dobija kao srednja vrijednost poznatih vrijednosti u okviru obilježja u kojoj se nalazi nedostajuća vrijednost.

Za razliku od **mean** imputacije, **median** imputacija predstavlja zamjenu nedostajućih podataka sa vrijednošću koja se nalazi u sredini sortiranog skupa. U slučaju da je ukupan broj podataka paran broj, **median** se računa kao srednja vrijednost dvije vrijednosti koje se nalaze u sredini sortiranog skupa. **Mode** imputacija predstavlja

zamjenu nedostajuće vrijednosti podatka sa vrijednošću podatka koji se najčešće pojavljuje u jednom obilježju.

Upotrebom imputacije sa konstantnom vrijednošću se nedostajuća vrijednost zamjenjuje sa nepromjenjivom vrijednošću koja se određuje na osnovu tipa i vrijednosti podataka iz obilježja.

Last Observation Carried Forward (LOCF) i **Next Observation Carried Backward (NOCB)** se koriste ukoliko su u pitanju podaci vremenskih serija. Podaci vremenskih serija predstavljaju niz vrijednosti podataka koji su dobijeni uzastopno, često sa jednakim intervalima između njih. Razlika između **LOCF** i **NOCB** imputacije jeste ta što se uz pomoć **LOCF** imputacije svaka nedostajuća vrijednost mijenja sa posljednjom posmatranom vrijednošću, dok se sa **NOCB** svaka nedostajuća vrijednost mijenja sa prvom posmatranom vrijednošću, ali unazad.

Linearna interpolacija predstavlja linearni odnos između podataka i koristi vrijednosti koje ne nedostaju iz susjednih podataka za izračunavanje nedostajuće vrijednosti. Interpolacija predstavlja matematičku metodu koja prilagođava funkciju podacima i koristi je za ekstrapolaciju podataka koji nedostaju.

k-NN imputacija koristi algoritam **k** najbližih komšija za ocjenjivanje i zamjenu nedostajućih vrijednosti. Prilikom implementacije **k-NN** ne mora se pripremiti prediktivni model za svako obilježje sa nedostajućim vrijednostima. Nedostatak ovog algoritma predstavlja mala brzina izvršavanja. Nije uvijek lako odrediti **k** i mjeru sličnosti koja će se koristiti. Osjetljiv je na obilježja koji nisu od značaja.

Višestruka imputacija korišćenjem ulančanih jednačina (**MICE**) može da obrađuje promjenljive različitih tipova. Prvi zadatak prilikom implementacije ove strategije je utvrđivanje koje promjenljive će biti uključene u proces imputacije. U **MICE** proceduri pokreće se niz regresionih modela pri čemu se svaka promjenljiva sa podacima koji nedostaju modeluje uslovno prema ostalim promjenljivim u podacima. Generalno se izvodi 10 ciklusa, ali je potrebno istraživanje kako bi se identifikovao optimalan broj ciklusa prilikom unosa podataka pod različitim uslovima. Ideja je da do kraja ciklusa raspodjela parametara koji upravljaju imputacijama konvergiraju u smislu da postane stabilna. Kada se završi naznačeni broj ciklusa, čitav postupak imputiranja se ponavlja da bi se generisali višestruki imputirani skupovi podataka. Posmatrani podaci će biti isti u imputiranim skupovima podataka, razlikovaće se samo vrijednosti koje su nedostajale.

Metoda klasifikacije je metoda potpornih vektora (**SVM**) koja predstavlja algoritam koji se može koristiti i za klasifikaciju i za regresiju. Obično se koristi kada su podaci numerički i kada imaju samo dvije klase. Ukoliko je potrebno može se proširiti na više klasa i moguć je rad i sa nenumeričkim podacima. Cilj algoritma je da na osnovu hiper-ravni (**hyperplane**) koja razdvaja podatke kreira model uz pomoć koga će se predvidjeti kojoj klasi pripada nedostajući podatak. Ukoliko su podaci linearno razdvojeni, potrebno je pronaći maksimalnu marginu koja u odnosu na hiper-ravan razdvaja podatke. Zatim se na osnovu hiper-ravni, odnosno, njene jednačine kreira model. Na osnovu modela se računa rastojanje podataka

od hiper-ravni i na osnovu dobijenog rezultata se određuje kojoj klasi pripada. Margina predstavlja rastojanje od hiper-ravni do potpornih vektora, dok potporni vektori predstavljaju podatke koji su najbliže hiper-ravni i na taj način potporni vektori utiču na položaj ravni. Korišćenjem potpornih vektora se pronalazi maksimalna margina hiper-ravne. Potporni vektori pomažu u izgradnji optimalne metode potpornih vektora i ukoliko ih nema blizu hiper-ravni, klasifikacija će biti relativno laka. Ukoliko su podaci linearno nerazdvojni koristi se nelinearni **SVM**. Osnovna ideja jeste da se omogući da se podaci linearno razdvoje, a to se postiže preslikavanjem ulaznog vektorskog prostora u višedimenzionalni prostor. Razumijevanje načina rada **SVM** i **k-NN** algoritama izvršeno je na osnovu rada [6].

4. PREGLED IMPLEMENTACIJE I ANALIZA REZULTATA

Nakon detaljne pretrage više izvora podataka, pronađena su dva skupa podataka, koja predstavljaju dobru osnovu za dalju realizaciju projekta. Prvi skup podataka sadrži osnovne podatke o aplikacijama, a obilježja u okviru njega su: **Application name, Category, Rating, Reviews, Size, Installs, Type, Price, Content Rating, Genres, Last Updated, Current Ver** i **Android Ver**. Drugi skup podataka sadrži podatke o komentarima koje su korisnici mobilnih aplikacija ostavljali, a obilježja koja se mogu naći u okviru tog skupa su: **App, Translated_Review, Sentiment, Sentiment_Polarity, Sentiment_Subjectivity**. Nakon pronalazjenja odgovarajućih skupova podataka bilo je potrebno analizirati i razumjeti same podatke. Bilo je potrebno izbaciti nepotrebne podatke iz svakog pojedinačnog skupa, odrediti koja obilježja su bitna za predikciju i da li među njima ima koreliranih obilježja.

U prvom skupu podataka, koji sadrži osnovne podatke o aplikacijama, postojale su aplikacije koje nisu imale jedinstvene vrijednosti, postojali su duplikati aplikacija po imenu, pa su ti duplikati uklonjeni. Nad drugim skupom podataka, koji sadrži podatke o komentarima koje su korisnici mobilnih aplikacija ostavili, kao i o dodatnim informacijama u vezi značenja komentara primjenjen je isti postupak uklanjanja duplikata. Pod duplikatima su se smatrali aplikacije koje su imale u potpunosti iste vrijednosti obilježja u okviru drugog skupa podataka. Uklonjene su aplikacije koje imaju isto ime, komentar, sentiment, polaritet i subjektivnost. Nakon uklanjanja duplikata formiran je skup podataka spajanjem prethodno uređena dva skupa. Istraživanjem mašinskih algoritama, primjećeno je da u većini slučajeva isti bolje rade sa numeričkim vrijednostima tako da je svako obilježje nenumeričkog tipa konvertovano u numerički tip. Ciljno obilježje broj korisničkih preuzimanja/installiranja mobilne aplikacije (**Installs**) je sadržalo karakter +, koje je uklonjeno i na taj način je konvertovan u brojnu vrijednost (npr. 1000+ je konvertovano u 1000). Obilježje **Price** sadržalo je karakter \$ koji je uklonjen kako bi se vrijednost obilježja mogla konvertovati u numerički tip. Analizom vrijednosti obilježja **Reviews** uočeno je da jedna od vrijednosti sadrži karakter **M**, za koju je pretpostavljeno da predstavlja oznaku za broj milion. Nakon uklanjanja karaktera, vrijednost obilježja je pomnožena sa milion kako bi se ostvarila konzistentnost

sa ostalim vrijednostima u okviru **Reviews** obilježja. Radi očuvanja konzistentnosti, obilježje **Size** koje je bilo izraženo u **MB (megabyte)** i **kB (kilobyte)** je konvertovano u **B (byte)**. Tip aplikacije je dobio numeričku vrijednost u zavisnosti od toga da li je u pitanju plaćena ili besplatna aplikacija. Za plaćenu aplikaciju vrijednost je 1, dok za besplatnu vrijednost je 0. Napravljene su izmjene u nazivima žanrova aplikacije (**Genres**) da se ne bi dva ista žanra tretirala kao dva različita (npr. **Education, Education** se tretira kao **Education**) i svakom jedinstvenom žanru se dodjeljuje odgovarajuća numerička vrijednost. Sličan postupak je sproveden nad kategorijom aplikacije (**Category**), obilježjem koje daje informaciju kojoj ciljnoj grupi korisnika je namijenjena aplikacija (**Content Rating**), kao i za ime aplikacije (**Name**). **Sentiment** komentara sadržao je nenumeričke vrijednosti **Positive, Neutral, Negative**. Sve vrijednosti koje su imale pozitivan sentiment dobile su vrijednost 1. Vrijednosti sa neutralnim sentimentom zamijenjene su brojem 0, a za vrijednosti negativnog sentimenta dodijeljena je vrijednost -1.

Obilježja koja su uklonjena iz skupa podataka su datum posljednjeg ažuriranja (**Last Updated**), trenutna verzija aplikacije (**Current Ver**) i **Android** verzija (**Android Ver**), jer se na osnovu analize domena problema i rada [7] došlo do zaključka da navedena obilježja nisu potrebna za predviđanje popularnosti prilikom primjene izabranih mašinskih algoritama. Obilježje koje je izostavljeno u ovom skupu podataka su korisnički komentari (**Translated_Review**), jer su na osnovu njih već definisana ostala obilježja u samom skupu podataka. Prilikom primjene algoritama i strategija nad skupom podataka nad kojim su izvršene gore navedene izmjene, dobila se velika tačnost i mala standardna greška procjene. Sve to je ukazivalo da rezultati nisu realistični i da postoji određena zavisnost među obilježjima čije postojanje je određeno koeficijentom korelacije. Prilikom poređenja ciljnog obilježja **Installs** sa drugim obilježjima iz skupa podataka, najveći koeficijent korelacije dobijen je za obilježja **Reviews** i **Size**, pa je zbog toga odlučeno da se navedena obilježja uklone iz skupa podataka. Na ovaj način je formiran novi skup podataka nad kojim su dalje primjenjeni odgovarajući algoritmi mašinskog učenja i strategije za rad sa nedostajućim vrijednostima.

Prilikom primjene strategija nad formiranim skupom podataka, kreiran je novi skup podataka za svaku primjenjenu strategiju. Zatim je svaki skup podataka podijeljen na trening i test skup u odnosu 70:30%. Nad trening skupom su primjenjeni klasifikatori **k** najbližih komšija – **k-NN** i mašine potpornih vektora (**Support Vector Machine – SVM**).

Koeficijent determinacije **R²** je korišćen kao mjera tačnosti koja pokazuje koliko promjene jedne promjenljive utiču na promjene druge promjenljive. Za evaluaciju rješenja korišćena je standardna greška procjene (**root mean squared error**). Što su obilježja više nezavisna, to je i manja standardna greška procjene.

4.1 Rezultati

U ovom poglavlju će biti prikazani rezultati koji su dobijeni primjenom različitih kombinacija strategija i algoritama u okviru Tabele 1.

Tabela 1. Pregled rezultata

Strategija	k-NN		SVM	
	Tačnost	Standardna greška procjene	Tačnost	Standardna greška procjene
<i>ListWise</i>	0.7418884433102443	71615195.2865384	0.3106088224571637	157315693.23087773
<i>Dropping column</i>	0.48601864181091875	140923004.58218968	0.3111003510470887	162417004.70113793
<i>Mean</i>	0.7415567122624379	103308206.14519987	0.3011741919864423	159221101.60640624
<i>Median</i>	0.743251422345963	103296162.41850609	0.30407940927248517	152529390.80998662
<i>Mode</i>	0.743251422345963	103300617.96097405	0.29947948190291734	150434220.04650003
<i>Imputing with Constant</i>	0.7447040309889844	103300611.1564436	0.3136424161723762	150510067.28282884
<i>LOCF</i>	0.7427672194649558	103308065.35110208	0.3165476334584191	157414701.41565114
<i>NOCB</i>	0.743251422345963	103314107.51389693	0.31013194528507443	154134711.7035762
Linearna interpolacija	0.7425251180244522	103323591.34203984	0.31678973489892265	165293235.85005927
<i>Imputation using k-NN</i>	0.7415567122624379	103308206.14519987	0.3153371262559012	155146761.0632167
<i>MICE</i>	0.7415567122624379	103308206.14519987	0.30492676431424764	149713477.606884

SVM algoritam je dao znatno lošije rezultate u odnosu na *k-NN* u kombinaciji sa bilo kojom strategijom. Najveća tačnost prilikom upotrebe *SVM* algoritma je 0.31678973489892265, a *k-NN* je pokazao bolji rezultat čak iako se izvrši poređenje sa najmanjom tačnosti upotrebom *k-NN* algoritma koja iznosi 0.48601864181091875. Najmanja tačnost dobila se prilikom upotrebe *mode* strategije i *SVM* algoritma (0.29947948190291734). Najveći koeficijent determinacije dobio se prilikom primjene *k-NN* algoritma za klasifikaciju i imputacije sa konstantnim vrijednostima kao strategije za rad sa nedostajućim vrijednostima i on iznosi 0.7447040309889844.

5. ZAKLJUČAK

U radu je jasno definisan način na koji strategije za rad sa nedostajućim vrijednostima utiču na rješavanje konkretnog problema. Izvršena je podjela nedostajućih vrijednosti na određene tipove, kao i kratak pregled svakog od njih. Izvršen je kratak pregled metodologije koja je korišćena za pripremanje skupa podataka za sam proces predikcije. Predstavljena je podjela strategija za rad sa nedostajućim vrijednostima i opisivanje svake od njih. Osnovne osobine algoritama mašinskog učenja su, takođe, navedeni.

Izvršeno je izbacivanje duplikata, nepotrebnih i koreliranih obilježja, i određeno je koja obilježja su bitna za dalju analizu i upotrebu. Nad spojenim skupom podataka urađena je konverzija podataka u numeričke vrijednosti. Nad uređenim skupom podataka izvršena je primjena svih strategija koja je dovela do formiranja novih skupova podataka gdje se svaki od njih vezuje samo za jednu strategiju. Nad svakim skupom podataka dalje su se primjenjivali algoritmi mašinskog učenja namjenjeni za klasifikaciju.

SVM algoritam je dao znatno lošije rezultate u odnosu na *k-NN* u kombinaciji sa bilo kojom strategijom. Najmanji koeficijent determinacije se dobio prilikom upotrebe *mode* strategije i *SVM* algoritma, a najveća tačnost se dobila prilikom primjene *k-NN* algoritma za klasifikaciju

i linearne interpolacije kao strategije za rad sa nedostajućim vrijednostima.

6. LITERATURA

- [1] D. B. Rubin, "Inference and missing data", Biometrika 1976.
- [2] <https://www.kaggle.com/parulpandey/a-guide-to-handling-missing-values-in-python> (pristupljeno u septembru 2020.)
- [3] Z. Zhang, "Missing data imputation: focusing on single imputation", 2016.
- [4] H. Kang, "The prevention and handling of the missing data", 2013.
- [5] J. Zhang, D. Chen, "Interpolation calculation made EZ"
- [6] Jasmina Đ. Novaković, "Rešavanje klasifikacionih problema mašinskog učenja", 2013.
- [7] G. Lee, T. S. Raghu, "Determinants of Mobile Apps Success: Evidence from App Store", 2014.

Kratka biografija:



Sreten Petrović rođen je u Bijeljini 1995. god. Završio je gimnaziju JU Srednjoškolski centar "Vuk Karadžić" u Loparama 2014. godine. Osnovne akademske studije na Fakultetu tehničkih nauka u Novom Sadu završio je 2018. godine. Master rad na Fakultetu tehničkih nauka iz oblasti Računarstva i automatike odbranio je 2020.god. Kontakt: sreten.95@hotmail.com

ПРИМЈЕНА СОФТВЕРСКИХ АЛАТА ЗА ПРОРАЧУН НАПАЈАЊА СОЛАРНОМ ЕНЕРГИЈОМ КУЋЕ У ДВОРОВИМА**SOFTWARE TOOLS APPLICATION IN CALCULATION OF SOLAR ENERGY POWER SUPPLY FOR THE HOUSE IN DVOROV**

Зоран Вуковић, Владимир Катић, *Факултет техничких наука, Нови Сад*

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У раду је уопштено речено о обновљивим изворима енергије и њиховој подјели. Након тога је дат већи значај соларног енергији и фотонапонским соларним системима. Израчуната је производња соларне електране и упоређена са потрошњом домаћинства. Одрађен је модел фотонапонске електране у матлаб Симулинк окружењу уз помоћ којег су добијени одзиви произведене снаге. Извршена је дискусија резултата, гдје су упоређени резултати добијени мјерењем и симулацијом.

Кључне ријечи: Соларна електрана, Фотонапонски системи, Соларни панели

Abstract – In this work it has been generally said something about renewable energy sources and their division. After that bigger importance has been given to solar energy and photovoltaic solar systems. Production of solar power plant has been calculated and compared with the household consumption. The model of photovoltaic power plant has been done in Matlab Simulink, by which the response of produced power has been gained. The discussion about the results has been led and the results gained by measurements and simulations have been compared..

Keywords: Solar power plant, Photovoltaic systems, Solar panels

1. УВОД

Обновљиви извори енергије подразумевају изворе енергије који се налазе у природи и који се обнављају у цјелости или дјелимично, посебно енергија водотокова, вјетра, неакумулирана сунчева енергија, биомаса, биомаса животињског поријекла, геотермална енергија, биогорива, биогаз, синтетички гас, депонијски гас, гас из постројења за третман комуналних вода и отпадних вода из прехранбене и дрвно – прерађивачке индустрије које не садрже опасне материје. Обновљиви извори енергије се користе за производњу електричне, механичке и топлотне енергије. Фотонапонски (ФН) соларни системи раде на принципу фотонапонске конверзије сунчевог зрачења која се врши у соларним, ФН ћелијама које се израђују од полупроводничких материјала.

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Владимир Катић, ред. проф.

ФН систем подразумева систем помоћу кога се врши снабдијевање једносмјерном (DC), наизмјеничном (AC) или и једносмјерном и наизмјеничном струјом. Системи на којима се врши конверзија сунчевог зрачења и у електричну и у топлотну енергију се називају мултифункционални или хибридни системи. ФН ћелија је полупроводнички склоп који врши конверзију соларне у електричну енергију, односно сунчевог зрачења у једносмјерну струју.

Максимални излазни напон једне соларне ФН ћелије се креће у опсегу 0,6 V – 0,7 V. Међутим, приликом рада, генерисана струја производи пад напона на унутрашњој отпорности ћелије, тако да се номинални напон креће између 0,4 V и 0,5 V. Према пројектованом излазном напону и снази ФН ћелије се везују редно у модуле, а они даље редно и/или паралелно у ФН панеле. Излазни напон ФН панела прати потребе примјене, односно формира се тако да одговара неком једносмјерном (DC) систему или акумулатору, односно да буде погодан за рад инвертора (DC/AC претварача) у случају прикључења на мрежни (AC) систем [1].

2. СОФТВЕРСКИ АЛАТИ ЗА МОДЕЛОВАЊЕ И АНАЛИЗУ ФОТОНАПОНСКИХ ЕЛЕКТРАНА

За моделовање и анализу ФН електрана постоји већи број софтверских алата [2]. У овом раду биће коришћени и описани неки од њих, односно:

- PVGIS (Photovoltaic Geographical Information System),
- Homer Pro (Hybrid Optimization of Multiple Electric Renewables), и
- Matlab (MATrix LABoratory).

2.1. PVGIS (Photovoltaic Geographical Information System)

Овај софтверски алат служи за процјену производње електричне енергије самосталног ФН система или ФН система прикљученог на мрежу. Суштина софтвера је да покаже какав утицај има изабрана локација на планирање ФН система, као и база података о сунчевом зрачењу на тој локацији за дужи временски период. Помоћу PVGIS-а могуће је израчунати колико енергије се може добити из различитих врста ФН система на готово било ком мјесту на свијету [3].

2.2. Homer Pro

Homer Pro омогућава ручно или аутоматски, путем унесених координата мјеста, одређивање података о сунчевој радијацији, температури, брзини вјетрова,

итд. Након тога се у софтверу формира модел за који се жели направити симулација.

На основу модела и унесених података о сунчевој радијацији и параметара (оптерећења, ФН панела и конвертора) можемо добити симулацију рада система и приказ укупне годишње производње и потрошње енергије [4].

2.3. Matlab/Simulink

Матлаб је виши програмски језик намјењен обради података у форми матрица и бројева. Сам назив Матлаб изведен је од *MATrix LABoratory* (лабораторија за матрице), чиме се сугерише да су матрице основни ентитети за нумеричке прорачуне, за разлику од неких нижих програмских језика као што је *Fortran* или *Pascal*.

Симулинк је графичко програмско окружење засновано на Матлабу за моделовање, симулацију и анализу вишедоменских динамичких система [5].

3. ОПИС ОБЈЕКТА И ЕНЕРГЕТСКЕ КАРАКТЕРИСТИКЕ

Објекат (кућа), који је потребно напајати из ФН панела, налази се у мјесту Дворови (БИХ) на свега 3 km од Бијељине, у сјевероисточном дијелу Републике Српске, у Семберији, равничарској области између Посавине, Подриња, Срема, Мачве и мајевичког подбрђа.

Дворови се налазе у средишту између ријека Саве и Дрине и путева који преко Раче и Павловић моста из Србије воде у Бијељину и даље до Бања Луке и Источног Сарајева.

3.1. Годишња потрошња електричне енергије

У објекту се налазе стандардни кућни апарати, системи гријање, климатизације и још неки додатни неопходни електрични апарати и уређаји. У табели 1. је приказана потрошња електричне енергије по мјесецима. Ако се претпостави да просјечни однос потрошње више и ниже тарифе износи 70% на према 30%, и чињеница да су дужина трајања зимске и љетне сезоне подједнаки, долази се до просјечне цијене ел. енергије по kWh од 0,1116 KM, чиме се могу израчунати и месечни, односно годишњи трошкови за електричну енергију.

3.2. Енергетске карактеристике

На основу програма *PVGIS* добија се да је соларни енергетски потенцијал на локацији објекта у Дворовима такав да се из 1 kWp ФН панела нагнутог под углом 30° према југу, уз уважавање 14% електричних губитака у водовима и инвертору, може добити 1.197 kWh електричне енергије.

Да би се покрила годишња потрошња објекта, потребно је инсталирати 6.796 kWh/1.197 kWh, односно соларну ФН електрану излазне снаге 5,7 kW. С обзиром да се снага укупна панела узима 10%-20% већа од излазне снаге, а да је просечна површина једног ФН панела снага 240-290 Wp око 1,62 m², произилази да би за такву кровну ФН електрану требало од 38,2 m² до 46,2 m² кровне површине.

Табела 1. Годишња потрошња електричне енергије по мјесецима

Мјесец	Потрошња (kWh)	Уплата (KM)
Јануар	606	67.62
Фебруар	547	61.05
Март	579	64.61
Април	583	65.06
Мај	552	61.60
Јун	517	57.69
Јул	572	63.83
Август	567	63.27
Септембар	601	67.08
Октобар	537	59.93
Новембар	525	58.59
Децембар	610	68.07
Годишња потрошња	6.796	758.4

4. ИДЕЈНО РЈЕШЕЊЕ НАПАЈАЊА ЕЛЕКТРИЧНОМ ЕНЕРГИЈОМ ПРЕКО СОЛАРНИХ ПАНЕЛА

4.1. Прорачун могуће снаге кровне ФН електране

За прорачун могуће снаге електране и одређивање броја панела који се могу монтирати на кров куће (објекта) користи се онлајн алат Base K2 Systems. То је алат за пројектовање развијен од стране њемачке фирме K2 Systems која се бави производњом и продајом носећих конструкција за соларне панеле [6].

Приликом пројектовања кућног фотонапонског система за изабрани соларни панел Benq SUNVIVO PM060MW2, онлајн алат Base K2 Systems је на основу задатих димензија крова одредио да је могуће поставити максимално 44 панела.

Снага кровне електране би била између:

$$P_{min} = P_{panela\ min} \cdot N_{panela} = 240\ W_p \cdot 44 = 10,56\ W_p \quad (1)$$

$$P_{max} = P_{panela\ max} \cdot N_{panela} = 290\ W_p \cdot 44 = 12,76\ W_p \quad (2)$$

Инвертор који одговара овом прорачуну треба да задовољи око 80% - 90% укупне снаге соларних панела.

Поређењем овог и резултата из подпоглавља 3.2, може се констатовати да је на кров могуће сместити ФН електрану, која је оквирно двоструко веће снаге од енергетских потреба објекта.

С обзиром на променљивост сунчевог зрачења и одређену несигурност таквог напајања, инсталирање и тако увећаног капацитета се може оправдати.

4.2. Идејно рјешење кровне ФН електране

Како је већ речено на кров куће може стати 44 ФН панела и да ће снага ФН електране бити 12,76 kWp. Идеја је да се формирају 4 ФН низа (стринга) и да се сваки од низова састоји од 11 панела чиме се добија ФН електрана од 44 соларна панела.

4.3. Годишња производња електричне енергије добијена преко софтверског алата PVGIS

Софтвер PVGIS је описан у подпоглављу 2.1 при чему га сада користимо да бисмо одредили укупну производњу електричне енергије соларне електране, која је приказана у табели 2.

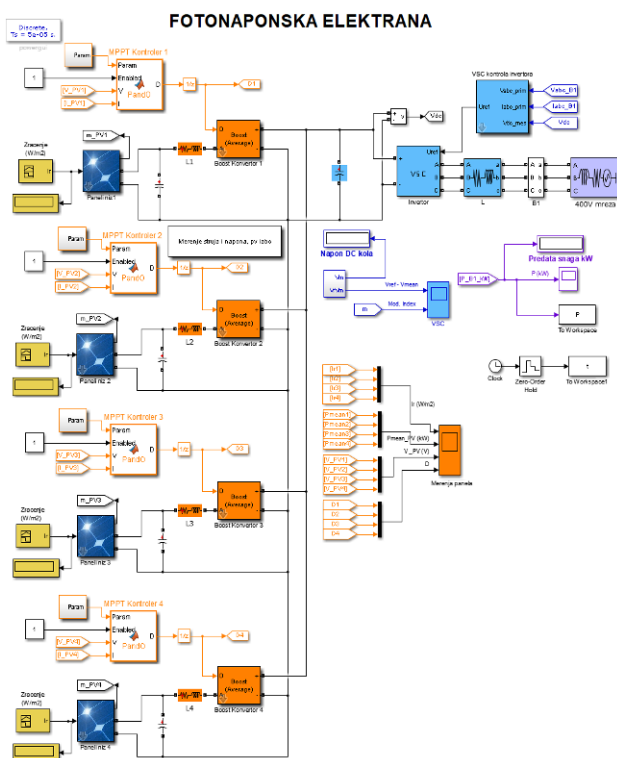
Табела 2. Годишња производња електричне енергије ФН система по мјесецима према PVGIS [3]

Мјесец	Em	Hm
Јануар	613	58.57
Фебруар	666.76	64.58
Март	1206.49	121.46
Април	1489.20	155.94
Мај	1622.07	173.65
Јун	1671.54	181.93
Јул	1843.76	203.68
Август	1734.73	190.8
Септембар	1372.78	145.79
Октобар	1112.44	113.24
Новембар	800.60	78.08
Децембар	539.29	51.75
Укупно годишње	14672	1539

Легенда: *Em* – просјечна мјесечна производња електричне енергије у разматраном систему (kWh); *Hm* – просјечна мјесечна сума зрачења глобалне радијације коју прима модул разматраног система (kWh/m²)

5. МОДЕЛ ФН ЕЛЕКТРАНЕ У МАТЛАБУ

Модел ФН електране чине четири низа ФН панела повезана на заједнички инвертор (слика 1). Инвертор је управљан импулсно – ширинском модулацијом, а излазни напон је преко једноставног филтра повезан на мрежу. Мрежа је представљена преко импедансног модела.



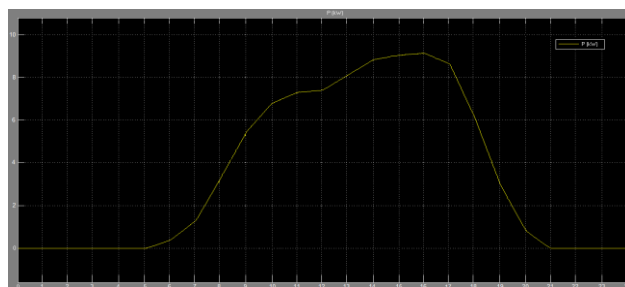
Слика 1. Комплетан Симулинк модел ФН електране

Први низ ФН панела је означен у моделу као блок „Панели низ 1“. У том блоку су наведени подаци о панелима као и броју панела коју низ садржи. На овај блок су повезани подаци о сунчевом зрачењу који су означени као „Зрачење (W/m²)“. У њему ће бити задато сунчево зрачење у току једног дана за сва четири годишња доба.

6. СИМУЛАЦИЈА ДНЕВНОГ РАДА ФН ЕЛЕКТРАНЕ

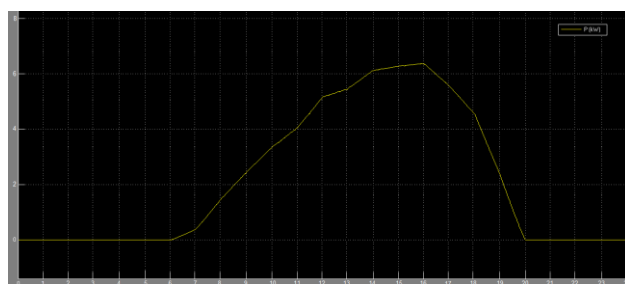
6.1. Одзиви све четири соларна низа при промјенљивој радијацији

На слици 2. приказан је одзив снаге ФН електране који је добијен на основу задате промјенљиве радијације за један дан у јуну. Како радијација почиње да расте у јутарњим часовима тако почиње и производња снаге. Највећа производња снаге биће у послије-подневним часовима.

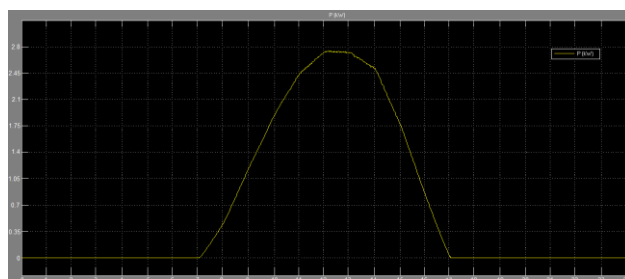


Слика 2. Одзиви снаге ФН електране за један дан у јуну

На слици 3. приказан је одзив снаге ФН електране који је добијен на основу задате промјенљиве радијације за по један дан у септембру и марту. Разлог што су на истом графику приказани одзиви за ова два мјесеца лежи у томе што у овим мјесецима (прољеће и јесен) владају приближно исте временске прилике са истим интензитетом сунчевог зрачења. На слици 4. приказан је дијаграм промјене снаге за један дан у децембру.



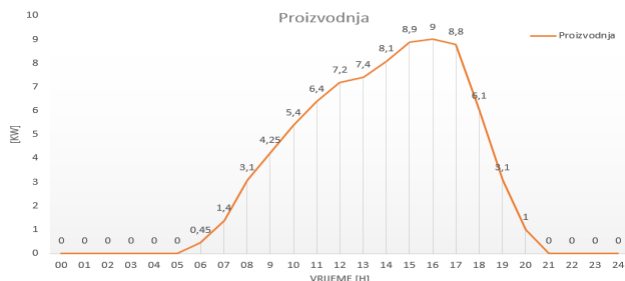
Слика 3. Одзиви снаге ФН електране за један дан у септембру и марту



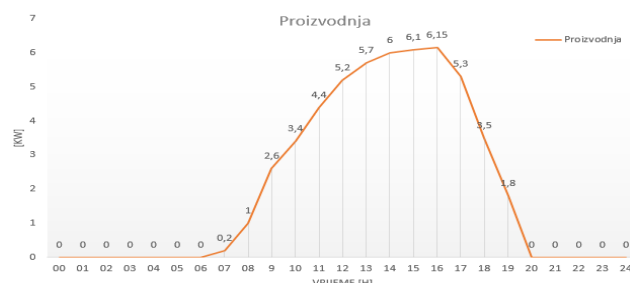
Слика 4. Одзиви снаге ФН електране за један дан у децембру

6.2. Генерисање дневних дијаграма за производњу соларне електране

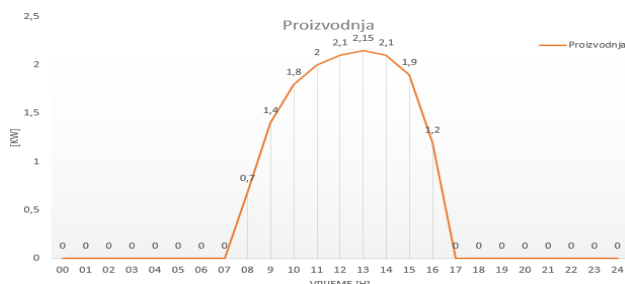
На следећа три дијаграма (5, 6, 7) приказана је произведена снага соларне електране на дневном нивоу за четири годишња доба, добијена помоћу софтверског алата PVGIS. У разматрање су узети по један дан из децембра, марта, јуна и септембра чиме су обухваћена све четири годишња доба.



Слика 5. Графички приказ произведене снаге добијене помоћу PVGIS-а за један дан у јуну



Слика 6. Графички приказ произведене снаге добијене помоћу PVGIS-а за један дан у септембру и марту



Слика 7. Графички приказ произведене снаге добијене помоћу PVGIS-а за један дан у децембру

6.3. Поређење и дискусија

Визуелним поређењем симулираних резултата и резултата добијених помоћу PVGIS-а може се уочити велика сагласност. Да би се ово запажање потврдило, израчуната је разлика у вриједностима добијена симулацијом и вриједностима добијених из софтверског алата PVGIS за свако годишње доба и представљена у табели 3. Види се да су и апсолутна и релативна грешке прихватљиве, тј. да је модел ФН електране прихватљив.

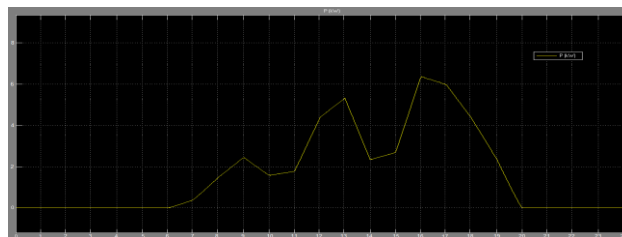
Табела 3. *Поређење резултата*

	Љето	Прољеће/Јесен	Зима
Δx	0,27	0,13	0,06
δ	4.78	3.18	3.65

Легенда: Δx – апсолутна грешка мјерења (kW); δ – релативна грешка мјерења (%)

6.4. Рад фотонапонске електране при промјенљивим временским условима

На слици 8. приказан је одзив снаге ФН електране при промјенљивим временским условима када у току дана постоје велика наоблачења, која често заклањају сунце и тако праве већу сјенку за ФН електрану.



Слика 8. Одзив снаге при промјенљивим условима

7. ЗАКЉУЧАК

Циљ рада је био да се прикаже колико је реално извести један овакав пројекат у којем би се поставили соларни панели на кров куће у Дворовима. Рад се највише базира на моделовању и анализирању ФН електране примјеном савремених софтверских алата. Анализирано је понашање ФН електране за све четири годишња доба.

Производња ФН електране ће бити највећа у лјетним мјесецима при јачем интензитету сунчевог зрачења. Модел је верификован поређењем са резултатима мерења.

8. ЛИТЕРАТУРА

- [1] Ч. Зельковић, „Обновљиви извори енергије – соларна енергетика“, Универзитет у Бањој Луци и Академска мисао, Бања Лука и Београд, 2018.
- [2] А. Гавриловић, В. Катић, „Преглед расположивих софтвера за соларне електране“, Зборник радова Факултета техничких наука, Нови Сад, Год. 34, Бр. 3, 2019.
- [3] <https://ec.europa.eu/jrc/en/PVGIS/docs/starting>
- [4] <https://www.homerenergy.com>
- [5] <https://www.mathworks.com>
- [6] <https://k2-systems.com/en/base-3>

Кратка биографија:



Зоран Вуковић рођен је у Доњем Вакуфу 1994. год. Факултет техничких наука, студијски програм Енергетика, електроника и телекомуникације уписао је 2013. год. Дипломирао је 2018. год, након чега је уписао мастер студије на смјеру „Електроенергетика – Енергетска електроника и електричне машине“.



Владимир Катић рођен је 1954. год. у Новом Саду. Докторирао је на Универзитету у Београду 1991. год. Од 2002. је редовни професор Универзитета у Новом Саду. Области интересовања су енергетска електроника, обновљиви извори електричне енергије, електрична возила и квалитет електричне енергије.

PREDLOG REŠENJA ZA MONITORING STANJA ASINHRONOG MOTORA**PROPOSED SOLUTION FOR MONITORING STATUS OF AN INDUCTION MOTOR**Marko Živković, Dejan Jerkan, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu je dat predlog za monitoring asinhronog motora koji je realizovan mikrokontrolerom ESP32, koji merene i obrađene podatke šalje ka serveru. Obradeni podaci se čuvaju na serveru i prikazuju se operateru koji može da prati stanje asinhronog motora i uvidi da li postoji potencijalni kvar u elektromotornom pogonu.

Ključne reči: Asinhroni motor, elektromotorni pogoni, monitoring, mikrokontroler ESP32, server, operater

Abstract – The paper presents an example of monitoring the status of the asynchronous motor which is realized by microcontroller ESP32. The microcontroller sends the data to the server. Processed data is stored on the server and it is shown to the operator who can follow the status of the asynchronous motor and check if the electric drive potentially has some fault.

Keywords: Induction motor, electric drive, monitoring, microcontroller ESP32, server, operator

1. ASINHRONI MOTOR

Asinhronne mašine su rotacione električne mašine koje spadaju u grupu naizmeničnih električnih mašina. Nikola Tesla je prvi konstruisao asinhroni motor 1887. godine, čiji se princip rada zasniva na obrtnom polju, koje je Nikola Tesla otkrio 1882. godine. Asinhroni motori su najzastupljeniji elektromehanički pretvarači u industriji zbog njihovih osnovnih osobina, a to su: robusnost, mala masa i zapremina, mogućnost rada sa veliki brzinama obrtaja, jednostavno održavanje, visok stepen korisnog dejstva.

U današnje vreme, najčešće se proizvode kavezne asinhronne mašine. Štapovi od kojih se sastoji kavez, prave se od aluminijuma (asinhronne mašine manje snage) ili od bakra (asinhronne mašine veće snage).

Rotor je sa obe strane kratkospojen prstenovima. Žlebovi u koje treba smestiti provodnike u vidu štapova, izoluju se tako što se tretiraju rastvorom kiseline, da bi se stvorio tanak izolacioni sloj soli gvožđa. Stator asinhronne mašine sastoji se od tri fazna namotaja, čije su ose prostorno pomerene za 120°.

Statorski namotaji mogu se povezati u zvezdu ili u trougao. Magnetno kolo statora i rotora izrađuje se od paketa limova, da bi se umanjili gubici usled vihornih struja.

NAPOMENA:

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

1.1. Zaštita asinhronog motora

Asinhroni motori zbog robusnosti i pouzdanosti su široko rasprostranjeni u današnjim elektromotornim pogonima. Oko 95% električnih motora u pogonima su asinhroni motori. Niskonaponski asinhroni motori se proizvode za snage do nekoliko stotina kilovata, dok se za veće snage, odnosno snage do 10MW proizvode za napone od 6kV, 10kV i 20kV.

Posledice kvara se mogu grupisati na sledeći način:

1. previsok napon dodira opasan po ljudski život,
2. obustava proizvodnje i oštećenje radne mašine,
3. troškovi popravke asinhronog motora i troškovi zamene oštećenih delova.

Zaštita elektromotornog pogona ima ulogu da kontroliše karakteristične veličine posmatranog uređaja (električni motor ili radne mašine ili električnog motora i radne mašine) i da u slučaju kvara ili opasnog pogonskog stanja, zaštitna oprema automatski preuzme sve neophodne mere da se kvar izbegne ili da se na minimum svedu posledice kvara ili opasnog pogonskog stanja, a takođe zaštita treba da upozori operatera ili osoblje koje se bavi monitoringom elektromotornog pogona.

Podela zaštitnih funkcija asinhronog motora u zavisnosti od vrste zaštite:

- zaštita od termičkog preopterećenja – isključuje motor u slučaju preopterećenja motora. Detekcija preopterećenja se vrši merenjem struja ili temperature motora,
- zaštita od neujednačenosti faznih struja – isključuje motor u slučaju pojavljivanja visokih neujednačenosti struja, koje za rezultat imaju povišenje gubitaka i temperature motora,
- zaštita od pogrešnog redosleda faza – usled pogrešnog redosleda faza motor će se obrtati u suprotnom smeru od željenog, što može dovesti do mehaničkih oštećenja radne mašine ili delova motora,
- zaštita od ispada faze – usled ispada jedne faze, zaštita isključuje motor da bi se sprečio jednofazni rad trofaznog motora, koji kao rezultat ima povećanje struje motora, smanjenje momenta i nemogućnost pokretanja.

1.1.1. Zaštita asinhronog motora od preopterećenja

Asinhroni motori kao i drugi električni uređaji poput sinhronih mašina, mašina jednosmerne struje i transformatora izrađuju se za odgovarajuće opterećenje koje u električnim uređajima rezultuje pojavu određenih temperatura. Maksimalnu temperaturu električnog motora određuje konstrukcija električne mašine, odnosno određuje je temperaturna klasa izolacije. Na natpisnoj pločici električnog uređaja stoji klasa izolacije i u zavisnosti od klase

izolacije propisana je dozvoljena maksimalna temperatura.

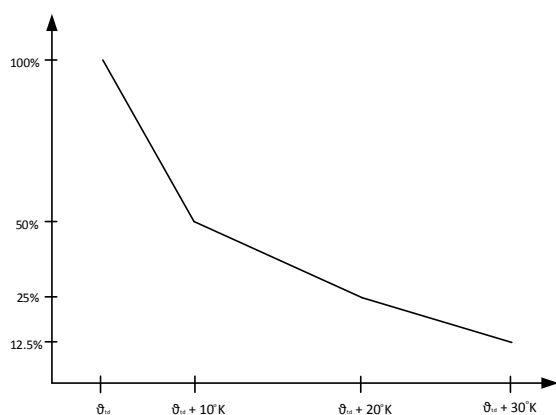
U tabeli 1 date su maksimalne dozvoljene vrednosti temperature u zavisnosti od klase izolacije upotrebljene za izolaciju namotaja električnog uređaja:

Tabela 1. dozvoljena nadtemperatura u zavisnosti od TKI

Klasa izolacije	A	E	B	F	H
Dozvoljena nadtemperatura [°C]	60	70	80	100	125

Da bi se ovi uslovi obezbedili, neophodno je obezbediti odgovarajuću zaštitnu opremu i zaštitne elemente, koje isključuju motor u slučaju opterećenja ili pojave temperature veće od dozvoljene.

Ako se u mašini ima temperatura veća od dozvoljene (slika 1), ona skraćuje životni vek izolacije.



Slika 1. Smanjenje životnog veka izolacije u zavisnosti od povećanja temperature

Najjednostavniji način zaštite motora od preopterećenja se izvodi ugradnjom uređaja koji isključuje motor sa mreže prilikom nastanka kvara ili ako nastane strujno ili termičko preopterećenje ili bilo kog drugog stanja koje bi moglo da ošteti asinhroni motor.

Za dozvoljene vrednosti struja motora zaštita ne sme da reaguje, a u slučaju da se te vrednosti struje premaše, potrebno je podesiti i izabrati zaštitu tako da brzina delovanja zaštite zavisi od veličine i trajanja prekoračenja opterećenja.

Potrebno je podesiti zaštitu tako da ona ne reaguje prilikom struje pokretanja asinhronog motora. Takvu karakteristiku delovanja imaju uređaji koji se koriste za zaštitu od preopterećenja:

- bimetalni releji,
- topljivi osigurači,
- elektronski releji za preopterećenje.

1.1.2. Zaštita asinhronog motora od jednofaznog rada

Zaštita motora od jednofaznog rada može se obaviti na dva načina, odnosno može se obaviti preko dva zaštitna uređaja:

- prekostrujna termička zaštita može da spreči jednofazni rad motora, jer ako se to desi, motor će biti preopterećen i prekostrujna termička zaštita će reagovati,

- postoje posebne vrste uređaja koji isključuju motor ako se pojavi jednofazni rad motora, a neki uređaji ne dozvoljavaju da se motor uključe ako je potencijalno takav rad moguć. U svakoj fazi kod ovog uređaja nalazi se elektromagnet kojeg struja faze drži uključenog. U slučaju da dođe do prekida faze, elektromagnet se aktivira i taj elektromagnet deluje na okidački mehanizam prekidača.

Ako se pravilno odabere prekostrujna termička zaštita i ako se pravilno podesi ova zaštita, ona može dobro da obuhvati i preopterećenje motora izazvano jednofaznim radom, tako da se u većini slučajeva odustaje od posebne zaštite od jednofaznog rada motora.

1.1.3. Zaštita asinhronog motora od obrnutog redosleda faza

Zaštita od obrnutog redosleda faza koristi se u slučajevima ako bi obrnut smer obrtanja asinhronog motora imao negativne posledice po rad radne mašine, poput mehaničkih oštećenja asinhronog motora ili mehaničkih oštećenja radne mašine.

Zaštita od obrnutog redosleda faza izvodi se primenom releja priključenog na filter inverzne komponente napona ili struja. U normalnim uslovima na izlazu filtra nema napona ili struje inverzne komponente. Ako bi se promenio redosled faza (na primer zamene se faze L1 i L2), pojavio se pun iznos ove komponente i zaštita isključuje motor, tj. zaštita neće dozvoliti uključanje motora i signaliziraće da je potrebno zameniti redosled faza.

1.1.4. Zaštita asinhronog motora od prekida faze i neuravnoteženosti struja

Motor se štiti od prekida faze i neuravnoteženosti struja zbog dva razloga:

- povećanja struje motora,
- dodatnog povećanja struje kao posledice gubitaka uzrokovanih inverznom struje dvostruke učestanosti.

Kvarovi ovog tipa se detektuju filterom komponente inverznog redosleda. U slučaju rada asinhronog motora na dve faze, ako je došlo do pregorevanja osigurača u jednoj fazi, ovakav rad će dovesti do preopterećenja motora i može se otkloniti delovanjem zaštite od preopterećenja.

1.1.5. Zaštita od blokiranja rotora

U slučaju zaglavlivanja motora, odnosno rotora motora iz mehaničkih razloga, struje će se povećati do vrednosti polazne struje motora, a to za posledicu ima povećanje gubitaka u rotoru, a takođe kako je rotor zaglavljen, nema strujanja vazduha u motoru, odnosno nema ventilacije koja se dobija kao posledica kretanja rotora. Ako u pogonu postoji mogućnost da se rotor blokira, potrebno je obezbediti zaštitu. Prekostrujni releji koj se koriste za detektovanje ovog kvara se podešavaju na vrednost struje koja je manja od polazne struje.

1.1.6. Zaštita od učestalog pokretanja

Ako motor nije projektovan za pogon u kojem se ima učestalo pokretanje, takav način eksploatacije motora može uzrokovati oštećenja motora. Zaštita od učestalog pokretanja obezbeđuje se relejom koji obezbeđuje

vremensko kašnjenje pokretanja motora, koja se automatski ograničava i to na:

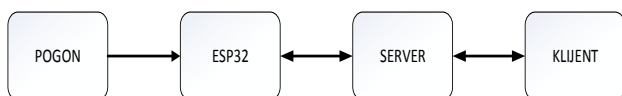
- broj pokretanja tokom nekog perioda,
- vremenski interval između dva pokretanja.

1.1.7. Zaštita od predugačkog trajanja pokretanja motora

Zaštita motora od predugačkog trajanja pokretanja obezbeđuje se primenom releja sa zadržskom čije je delovanje veće od vremena pokretanja motora, gde se meri struja ili brzina. Merenjem brzine ili struje može se jasno videti da li je proces pokretanja izvršen pravilno, odnosno da li su brzina ili struja jednake naznačenim vrednostima.

2. PREDLOŽENO REŠENJE SISTEMA ZA MONITORING

Sistem i predloženo rešenje za monitoring kvara asinhronog motora sastavljen je iz nekoliko komponenti i podsistema (slika 2), a komponente koje se nalaze u sistemu su: pogon i mrežni merni uređaj, mikrokontroler ESP32 koji koristi mrežni merni uređaj za merenje i za očitavanje karakterističnih veličina (poput napona, struje, faznog stava struje, aktivne snage, reaktivne snage, harmonijskog izobličenja napona i struje) i napajanje mikrokontrolera ESP32, server koji dobija podatke od strane mikrokontrolera ESP32, obrađuje podatke i čuva ih u bazi podataka i takođe u zavisnosti od merenih vrednosti dobija se mejl da je sistem u redu ili postoji neki kvar, klijentski deo aplikacije koji prikazuje podatke u human-readable formatu, odnosno formatu u kojem čovek može da ih pročita.



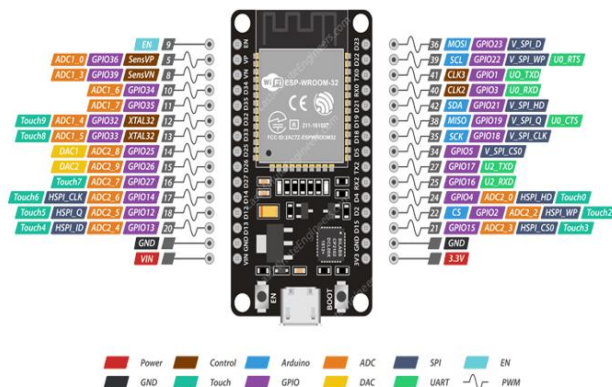
Slika 2. Blok dijagram sistema za monitoring

Strelicama na slici 2 prikazan je tok merenja i tok slanja podataka. Mrežnim mernim uređajem se očitava vrednost karakterističnih veličina, i merena veličine se šalje ka mikrokontroleru ESP32. Mikrokontroler ESP32 čita karakteristične veličine za elektromotorni pogon i obrađuje ih.

Nakon obrade struja, putem HTTP zahteva, podaci se šalju ka serveru. U slučaju da je zahtev validan, server dobija podatke i obrađuje ih. Nakon što server obradi podatke, u zavisnosti od izmerene vrednosti struje sa mrežnog mernog uređaja i postavljenog uslova (mejl upozorenja se šalje u slučaju da je merena struja veća od dvostruke vrednosti nominalne struje motora), vrednosti se smeštaju u bazu podataka i ispisuje se poruka na klijentskoj strani aplikacije i šalje se mejl upozorenja sa kompletnim izveštajem u trenutku kada je izmerena kritična vrednost struje.

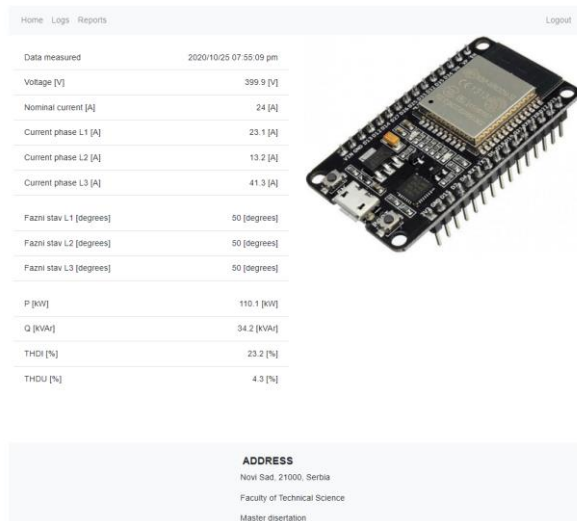
Izgled korišćenog mikrokontrolera ESP32 predstavljen je na slici 3.

Mikrokontroler ESP32 šalje podatke ka serveru u JSON formatu i podaci koji se šalju ka serveru su napon, nominalna struja, struja sve 3 faze (struje faza L1, L2 i L3), fazni stavovi struja L1, L2 i L3, aktivna snaga, reaktivna snaga, totalno harmonijsko izobličenje struje i totalno harmonijsko izobličenje napona koje server dodatno obrađuje i čuva u bazi podataka.



Slika 3. Izgled mikrokontrolera ESP32

Stranica se osvežava na svakih 10 minuta, tako da se imaju uvek sveže mereni podaci i ukoliko postoji greška u elektromotornom pogonu (ako je merena struja jednaka ili veća od dvostruke vrednosti nominalne struje), poslaće se mejl. Mikrokontroler ESP32 šalje podatke ka serveru na svakih 30 minuta.



Slika 4. Početna stranica sistema za monitoring

Primer merenih podataka, kao i izgled početne stranice dat je na slici 4.

4. VERIFIKACIJA PREDLOŽENOG REŠENJA

Za verifikaciju predloženog rešenja korišćen je asinhroni motor nominalnog napona 400V, nominalne snage 22kW, faktora snage 0.88. Motor je priključen direktno na mrežu. Pretpostavlja se da su asinhroni motor i električna mreža na koju je priključen asinhroni motor idealno simetrični i da su fazni stavovi jedne faze u odnosu na drugu pomereni idealno za 120 stepeni. Zbog preglednosti tabele, nisu dodati ukupno izobličenje napona i ukupno izobličenje struje koje se takođe šalju putem mikrokontrolera ESP32 i takođe data je samo struja faze L1, jer je pretpostavljeno da se radi o simetričnom režimu i struje faza L2 i L3 imaju istu merenu vrednost kao struja faze L1, samo su fazno pomerene za 120 stepeni jedna u odnosu na drugu.

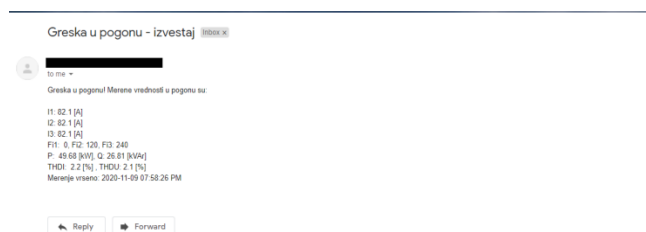
U rešenju je predviđeno da ako struja motora dostigne vrednost od dvostruke vrednosti nominalne vrednosti, da se u tom trenutku pošalje mejl upozorenja sa kompletnim izveštajem merenih vrednosti.

U tabeli 2 može se primetiti jedna kritična vrednost od 82.1A što je više od dvostruke vrednosti nominalne struje gde se očekuje da predloženo rešenje pošalje mejl.

Tabela 2. podaci dobijeni od mrežnog mernog uređaja koji se šalju mikrokontrolerom ESP32 (podaci mereni pola smene)

U	I _n	I ₁	φ ₁	φ ₂	φ ₃	P	Q
400.0	36.1	36.1	0	120	240	22023.40	11887.63
399.8	36.1	36.1	0	120	240	22012.39	11881.69
397.8	36.1	36.1	0	120	240	21902.27	11822.25
400.2	36.1	35.2	0	120	240	21496.99	11603.49
400.3	36.1	37.1	0	120	240	22663.00	12232.87
396.5	36.1	82.1	0	120	240	49675.72	26813.60
395.5	36.1	36.1	0	120	240	21775.64	11753.89
400.1	36.1	34.2	0	120	240	20881.06	11271.03

Predloženo rešenje je poslalo mejl u trenutku kada se imala struja vrednosti 82.1 A (slika 5), odnosno čim je primećena kritična vrednost struje, poslat je mejl upozorenja sa svim merenim parametrima dobijenim od strane mikrokontrolera ESP32.



Slika 5. Izveštaj pogona za kritičnu vrednost struje

5. ZAKLJUČAK

Predloženim sistemom za monitoring moguće je posmatrati i proveravati periodično karakteristične veličine poput napona mreže, struje faza L1, L2 i L3, faznih stavova struja L1, L2 i L3, aktivna snaga, reaktivna snaga i totalno harmonijsko izobličenje napona i totalno harmonijsko izobličenje struje. Sve posmatrane veličine su bitne za praćenje stanja jednog elektromotornog pogona, a takođe predloženo rešenje omogućava slanje mejla upozorenja i izveštaja u slučaju greške u elektromotornom pogonu u slučaju struje veće od dvostruke vrednosti nominalne struje motora.

Predloženi sistem za monitoring može se koristiti za integraciju nekog postojećeg sistema za nadzor ili monitoring, gde je potrebno obezbediti mikrokontroler ESP32, mrežni merni uređaj koji će komunicirati sa mikrokontrolerom ESP32 i slati mu neophodne podatke i potrebno je obezbediti server.

Predloženim rešenjem moguće je daljinsko praćenje podataka elektromotornog pogona, koje će povećati stepen složenosti elektromotornog pogona, ali ima višestruke prednosti, poput da operater ne mora da bude direktno u neposrednoj blizini asinhronog motora da bi pratio njegovo stanje, već može da bude u kontrolnoj sobi i prati stanje više asinhronih motora u pogonu, takođe u trenutku kada se desi greška u elektromotornom pogonu dobiće se automatski mejl sa upozorenjem.

Predloženo rešenje zbog korišćenih tehnologija moguće je nadograditi novim opcijama i omogućiti detaljnije praćenje stanja elektromotornog pogona.

6. LITERATURA

- [1] Vladan Vučković, „Električni pogoni“, Akademska misao, 2002.
- [2] Slobodan N. Vukosavić, „Električne mašine“, Akademska misao, 2010.
- [3] Emil Levi, Vladan Vučković, Vladimir Strezoski, „Elektroenergetski pretvarači“, FTN Izdavaštvo, 2013.
- [4] Vladislav Teodorović, „Električne pogonske mašine I“, HORIZONT Beograd, 1967.
- [5] Strahil J. Guševac, „Osnovni principi projektovanja u mrežama srednjeg i niskog napona“, Fakultet tehničkih nauka Novi Sad, 2014.

Kratka biografija:

Marko Živković rođen je u Novom Sadu 1995. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Energetska elektronika i električne mašine odbranio je 2020.god.

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

POZICIONA KONTROLA U ELEKTROMOTORNIM POGONIMA - ANALIZA TIPIČNOG INDUSTRIJSKOG REŠENJA

MOTOR DRIVES POSITION CONTROL - ANALYSIS OF THE TYPICAL INDUSTRIAL SOLUTION

Rade Jerkić, Dragan Milićević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu su analizirane karakteristike i mogućnosti pozicione kontrole primenom industrijskog rešenja baziranog na frekventnom pretvaraču serije Danfoss FC300 i specijalizovane MCO 305 kartice za kontrolu pokreta vratila asinhronne mašine. Posebna pažnja posvećena je analizi i praktičnoj realizaciji pozicione kontrole u laboratorijskim uslovima. U radu su realizovana dve eksperimentalne postavke kojima su testirani manje i više zahtevni algoritmi pozicione kontrole.

Ključne reči: Asinhroni motor, frekventni pretvarač, poziciona kontrola, Danfoss FC302, Danfoss MCO 305

Abstract – In this work, the characteristics and possibilities of position control are analyzed using an industrial solution based on a Danfoss FC 300 series frequency converter and specialized MCO 305 card for shaft position control of asynchronous machines. Special attention is dedicated to the analysis and practical realization of position control in laboratory conditions. For purpose of testing more or less demanding position control algorithms, two experimental setups were realized in this work.

Key words: Asynchronous motor, frequency converter, position control, Danfoss FC302, Danfoss MCO 305

1. UVOD

Široka lepeza primene asinhronih mašina u pogonu i industriji nametnula je potrebu za razvijanjem uređaja za kontrolu položaja njihovog vratila. Upravljanje pozicijom rotora otvara širok spektar pogodnosti u procesima industrijskih pogona. Kao posledica, nastaju uređaji za kontrolu rada asinhronih motora u oblasti pozicioniranja.

Tema ovog rada jeste princip funkcionisanja i primena uređaja pozicione kontrole u industrijskim elektromotornim pogonima. Jedno od najboljih rešenja pozicione kontrole na tržištu nudi Danfoss sa svojom MCO 305 programabilnom karticom. Glavna svrha MCO 305 kartice jeste da obezbedi tačnu, preciznu i efikasnu pozicionu kontrolu u raznim pogonima.

NAPOMENA:

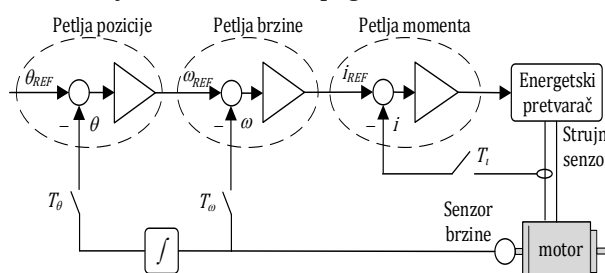
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dragan Milićević, vanr. prof.

2. OSNOVE UPRAVLJANJA POZICIJOM

U cilju regulacije neke veličine u pogonu, u regulisanim elektromotornim pogonima regulator pogona dinamički menja parametre napajanja mašine. Njihovom promenom regulator menja mehaničku karakteristiku pogona i tako pogon ulazi u neku novu radnu tačku koja je u datom trenutku izabrana sa potrebama pogona i regulacije. U zavisnosti od toga za šta su regulisani elektromotorni pogoni namenjeni, potrebno je regulisati:

- elektromagnetni momenat,
- brzinu rotora,
- položaj rotora.

Na slici 1 prikazan je klasična kaskadna regulaciona struktura u jednom motornom pogonu.



Slika 1. Regulacione petlje u regulisanom elektromotornom pogonu [1]

Što se tiče rada ovako strukturisane regulacije, rad nadređene regulacione petlje direktno zavisi od rezultata regulacione petlje koja je njoj podređena. Primera radi, regulator brzine zavisi od njemu podređene regulacione petlje momenta, ako unutrašnja petlja ne ostvari željeni moment, regulator brzine ne može da vrši svoj posao jer praktično ne upravlja momentom koji motor razvija. Da bi se obezbedilo da unutrašnja petlja na vreme reguliše veličinu sa kojom spoljna petlja želi da upravlja, razdvajaju se dinamike tih petlji, tako što se razdvajaju propusni opsezi ovih struktura, tako da unutrašnja petlja bude barem 5 do 10 puta brža od njoj nadređene petlje [2,3].

Regulisanje brzine rotora proizilazi iz Njutnove jednačine mehaničkog podsistema, čije je rešenje za mehaničku brzinu:

$$[\omega_r](t) = \omega_r(0) + \frac{1}{J} \int ([m_{el}] - [m_{opt}]) dt = \omega_r(0) + \frac{1}{J} \int [m_d] dt \quad (1)$$

gde je:

$[\omega_r]$ – brzina rotora,
 $[m_{el}]$ – električni momenat,
 $[m_{opt}]$ – momenat opterećenja,
 $[m_d]$ – dinamički momenat ($m_d = m_{el} - m_{opt}$).

Stoga, brzinu je moguće regulisati samo kontrolom električnog momenta koji motor razvija. Promena brzine sa početne vrednosti na neku novu referentnu vrednost ostvaruje se promenim momenta motora u odnosu na dati momenat opterećenja, a kada je potrebno održavati zadatu vrednost brzine pri promeni opterećenja, kao poremećaja, opet je potrebno reagovati kontrolom momenta, kako bi se poništilo dejstvo poremećaja.

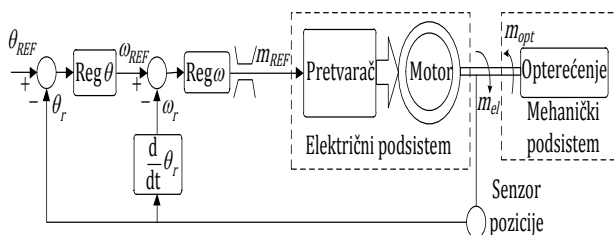
Položaj rotora, sa druge strane, se reguliše brzinom, što se jasno uočava iz:

$$[\vartheta_r](t) = \vartheta_r(0) + \int [\omega_r](t) dt \quad (2)$$

gde je:

$[\vartheta_r]$ – početni položaj,
 $[\omega_r]$ – brzina rotora.

Na slici 2 prikazana je blok šema pogona regulisanog po poziciji rotora. Referentna vrednost položaja se zadaje i poredi sa stvarnim položajem dobijenim integracijom brzine, te se tako generiše signal greške koji dolazi do regulatora položaja Reg θ . Dalje, regulator položaja zadaje referentnu vrednost brzine, pa stoga zaključujemo da optimalna regulacija pozicije moguća samo onda kada postoji unutrašnja regulaciona petlja po brzini [3].



Slika 2. Elektromotorni pogon sa regulisanim položajem vratila [1]

3. OPIS IZABRANOG REŠENJA

Za analizu industrijskog rešenja pozicione kontrole u ovom radu je korišćeno rešenje proizvođača Danfoss. Za praktičnu realizaciju pozicione kontrole izabrani proizvođač nudi modularnu koncepciju po kojoj se na klasičan pretvarač učestanosti dodaje specijalizovana dodatna kartica kojom se rad pretvarača proširuje i u domen pozicione kontrole.

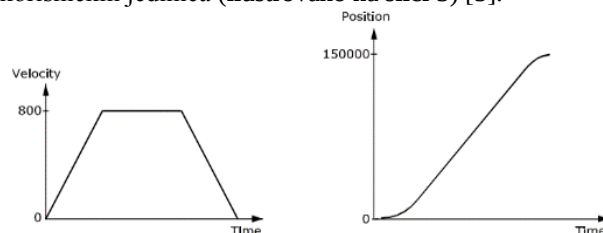
Konkretno, industrijsko rešenje frekventnog pretvarača serije FC300 sa unapređenim hardverskim karakteristikama i sa proširenim softverskim mogućnostima se proširuje karticom pozicione kontrole oznake MCO305 (eng. Motion Control Option 305).

Sam proizvođač frekventnih pretvarača u ponudi ima širok spektar snaga, pa se iz tog razloga izrađuju u nekoliko različitih veličina odnosno kućišta. Kao posledica toga, načini spajanja MCO 305 kartice se razlikuju. Za pretvarače najmanjih snaga tj. za veličine kućišta A2, A3, B3 ilustracija spajanja data je na slici ispod. Može se primetiti sistem povezivanja u stilu knjige, odnosno sa strane pretvarača [4].

Pozicioniranje, kao najvažnija funkcija MCO 305 uređaja, se može podeliti na nekoliko tipova:

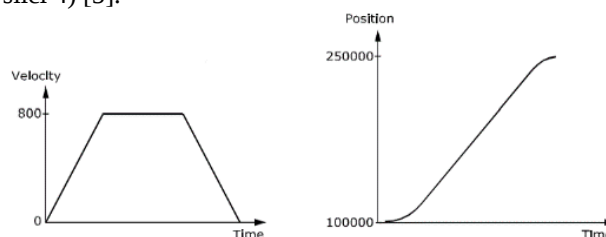
- Apsolutno pozicioniranje,
- Relativno pozicioniranje i
- Pozicioniranje pomoću dodirne sonde.

Apsolutno pozicioniranje se uvek odnosi na neku predefinisiranu apsolutnu nultu pozicionu tačku koja po pravilu mora biti definisana pre sprovođenja postupka apsolutnog pozicioniranja. U slučaju korišćene kartice, ako se koriste inkrementalni enkodori, nulta tačka se definiše upotrebom funkcije *Home*. Kod apsolutnih enkodera, nultu tačku daje sam enkoder. Na primer, ako je očitana pozicija 0 (nula) korisničkih jedinica, a zadata apsolutna vrednost 150000 istih, vratilo motora će otići na zadatu poziciju i prilikom toga će preći put od 150000 korisničkih jedinica (ilustrovano na slici 3) [5].



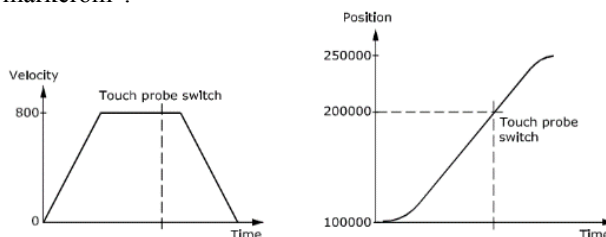
Slika 3. Apsolutno pozicioniranje [5]

Relativno pozicioniranje uvek je povezano sa stvarnom pozicijom. U tom smislu, moguće je izvršiti postupak pozicioniranja bez prethodnog definisanja nulte tačke. Na primer, ako je očitana trenutna pozicija 100000 korisničkih jedinica, a zadata relativna pozicija 150000 istih, vratilo pogonjenog motora će otići na poziciju 250000 tj. preći će put od 150000 jedinica (ilustrovano na slici 4) [5].



Slika 4. Relativno pozicioniranje [5]

Pozicioniranje pomoću dodirne sonde povezano je sa stvarnim položajem kada se aktivira ulaz sonde na dodir. To znači da je ciljani položaj definisan položajem sonde na dodir plus udaljenost pozicioniranja. Ovakvo pozicioniranje predstavlja relativno zadavanje položaja vratila u odnosu na aktivnost postojećeg senzora. Sonda je senzorski uređaj koji može biti mehanički, optički, senzor blizine ili nekog drugog tipa. Na primer, pozicioniranjem dodirne sonde u 50.000, pogon će raditi do aktivacije dodirne sonde na 200.000 i nastavlja do ciljanog položaja od 250.000 (ilustrovano na slici 5). Pozicioniranje sonde na dodir još se naziva i "pozicioniranje povezano sa markerom".



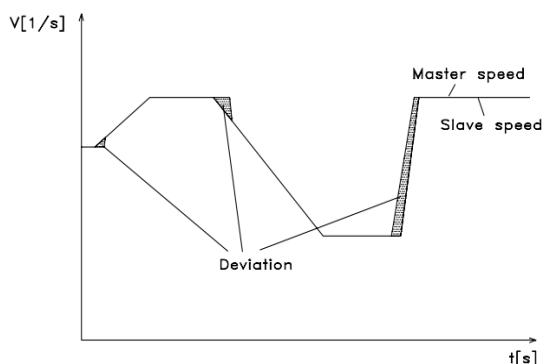
Slika 5. Pozicioniranje dodirnom sondom [5]

Takođe, izabrano rešenje nudi mogućnost i sinhronizacije i to one po brzini (SYNCV), po položaju (SYNCP) i po markerima (SYNCM).

Sinhronizacija po brzini predstavlja kontrolu brzine u zatvorenoj povratnoj sprezi. Brzina koja se prati (brzina *master* uređaja) pomnožena sa prenosnim odnosom čini zadatu vrednost za tzv. *slave* uređaj, uređaj koji ima ugrađenu MCO305 karticu.

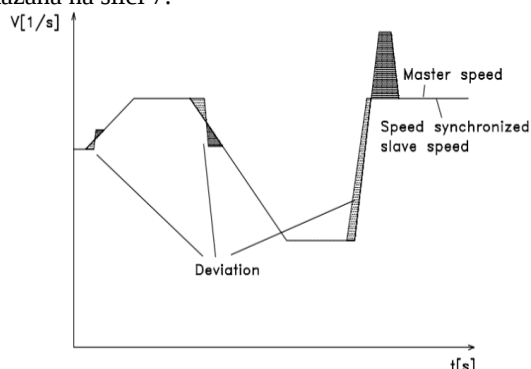
Ponašanje celokupnog sistema je takvo da pratilac, odnosno frekventni pretvarač sa integrisanom MCO305 karticom, sve vreme rada pokušava brzinu motora koji on kontroliše uskladiti sa brzinom nadređenog tj. *master* pretvarača.

Situacija ponašanja sistema u slučaju sinhronizacije po brzini je prikazana na slici 6.



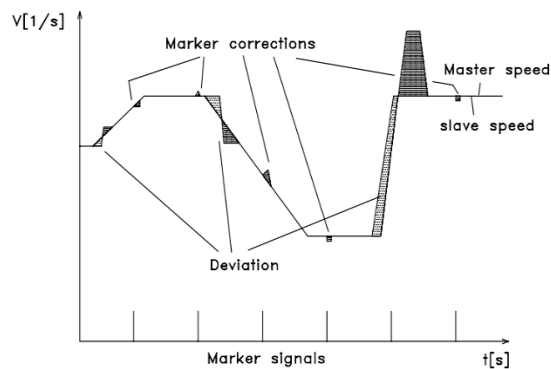
Slika 6. Sinhronizacija po brzini [5]

Sinhronizacija po položaju predstavlja kontrolu položaja u zatvorenoj povratnoj sprezi. U ovoj situaciji položaj nadređenog sistema pomnožen prenosnim odnosom predstavlja zadati položaj podređenog *slave* pretvarača. Kada se pozicija *master* sistema dostigne, podređeni *slave* sistem se dovodi u identičan položaj usklađen sa nadređenim sistemom. Poželjno je da *slave* jedinica bude brža i dinamičnija od *master* sistema kako bi sinhronizacija bila što tačnija na velikim brzinama, ubrzanjima i usporanjima. Situacija pozicione kontrole je prikazana na slici 7.



Slika 7. Sinhronizacija po položaju [5]

Sinhronizacija markerima predstavlja proširenu sinhronizaciju položaja gde se radi poravnanja *slave* markera sa *master* markerom vrši dodatna korekcija položaja. Signali glavnog i njemu podređenog markera mogu da budu i nulti impulsi enkodera ili neki spoljni senzori (povezani na digitalne ulaze). Pored toga što se mogu podesiti funkcije kao kod sinhronizacije položaja, može se podesiti i opseg markera.



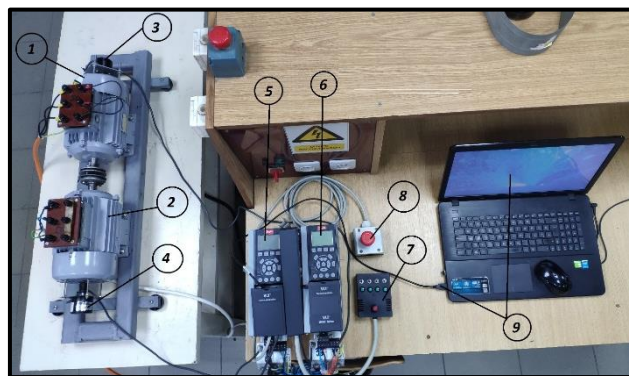
Slika 8. Sinhronizacija markerima [5]

4. LABORATORIJSKA IZVEDBA EKSPERIMENTATA

4.1 Opis eksperimentalne postavke

Eksperimentalna postavka na kome je izvršeno ispitivanje mogućnosti pozicione kontrole ispitivanog rešenja je prikazana na slici 9 gde je redom označeno [6]:

1. Četvoropolna kavezna asinhrona mašina nazivne snage 0,75 kW – Master mašina,
2. Četvoropolna kavezna asinhrona mašina nazivne snage 0,75 kW – Slave mašina,
3. Master inkrementalni enkoder 3600 imp/obr,
4. Slave inkrementalni enkoder 3600 imp/obr,
5. Frekventni pretvarač Danfoss FC-302 – Slave,
6. Frekventni pretvarač 2 Danfoss FC-302 – Master,
7. Komandni tasteri i preklopke,
8. Sigurnosni taster,
9. Računar sa instalisanim MCT10 softverom [8].



Slika 9. Eksperimentalna postavka

4.2. Primer 1 – Horizontalno klatno

Na slici 10 je prikazana eksperimentalna postavka kojom je realizovan ovaj eksperiment. Eksperimentom se proverava održavanje položaja vratila pri različitim opterećenjima.

Zamisao je da se za definisani položaj vratila koju kontroliše master mašina bude praćen po pozicionom mehanizmu od strane slave mašine. Poremećaj u ovom slučaju nastaje povećanjem opterećenja na vratilu slave mašine. Kako bi se ponašanje vratila moglo vizuelno bolje uočiti, na vratilo je pričvršćena poluga koja se usled ponašanja regulacionih mehanizama kontrole položaja ponašala na sličan način kako se ponaša klatno (otuda je vežba i dobila naziv).

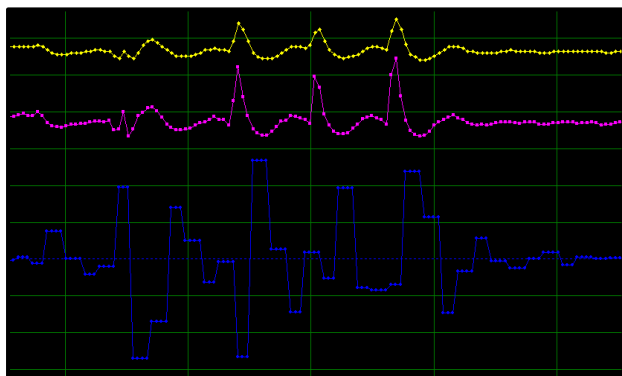


Slika 10. Izgled postavke oscilarnog klatna

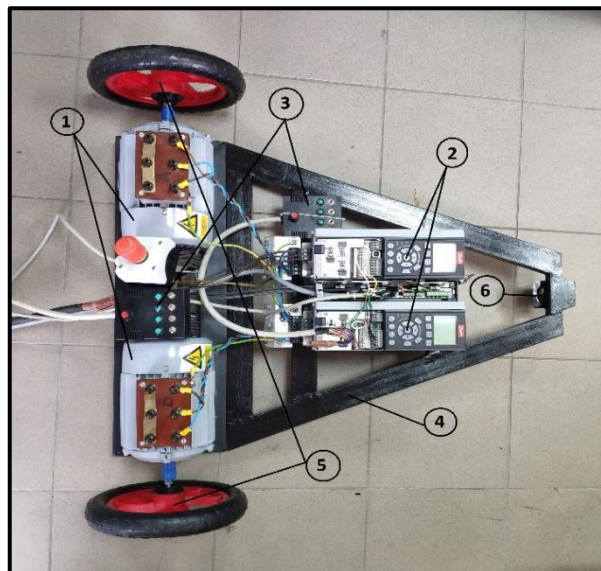
Na slici 11 je prikazan tipičan uočeni odziv sistema. Na slici su efektivna vrednost napona i efektivna vrednost struje prikazani ljubičastom u žutom bojom respektivno, dok je plavom bojom prikazana pozicija *slave* mašine. Slika daje situaciju ponašanja snimanih parametara u trenutku naglog opterećenja klatna. Uočava se ponašanja *slave* mašine koja pokušava mehanizmima automatske kontrole da se pozicionira i poravna svoju poziciju sa vratilom *master* mašine. Vidljivo je očekivano ponašanje celokupnog sistema koji nakon dovođenja poremećaja u vidu povećanja opterećenja klatna reaguje korekcionim merama kojima pokušava svesti grešku pozicije na nulu. Fizički, pozicija *slave* vratila se nekoliko puta pokreće gore – dole sve dok se ponovo ne uspostavi nulta greška odnosno dok se pozicija vratila *slave* mašine ne poravna sa pozicijom vratila *master* mašine. Na osnovu kretanja pozicije vratila *slave* mašine da se zaključiti da je odziv prigušeno oscilatoran. Treba napomenuti da prilikom izvođenja eksperimenta vratilo *master* jedinice nije pomerano ni u jednom smeru. Korišćeni regulator je PI tipa.

4.3. Primer 2 – Električna trokolica

Za realizaciju drugog eksperimenta je izabran sistem kod koga se vršilo pokretanje obe mašine, i *slave* i *master* jedinice. U tom smislu je osmišljena i realizovana postavka koja je prikazana na slici 12. Sadržina postavke je identična onoj iz eksperimenta 1.



Slika 11. Odzivi sistema – struja, napon i pozicija



Slika 12. Izgled postavke autića

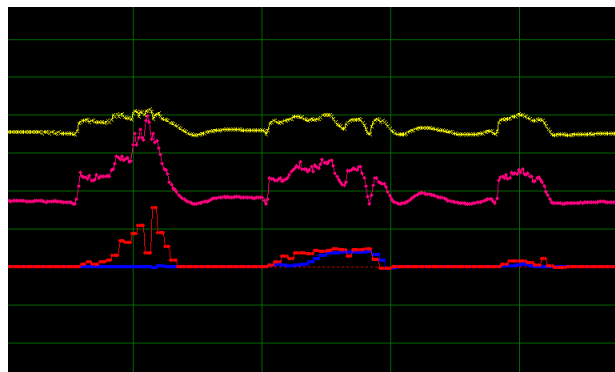
U ovom slučaju je iskorišćena kontrolni mehanizam praćenja brzine. Sistem se u praksi često naziva i imenom električno vratilo.

Na slici 13. je dat prikaz relevantnih veličina koje su praćeni tokom izvođenja jednog od eksperimenata.

Crvenim i plavim signalima su prikazane brzine *slave* i *master* mašine, respektivno. Signal roze boje predstavlja promenu efektivne vrednosti napona, dok žuta boja predstavlja signal efektivne vrednosti struje master jedinice.

Na slici se uočavaju tri segmenta.

Prvi segment na grafiku prikazuje situaciju u kojoj slave jedinici nije data komanda za praćenje brzine. Vidljivo je da i pored promene brzine master jedinice izostaje reakcija slave mašine. u sačinjavaju Slave i Master brzine koje su očevidno jednake, nakon čega sledi skretanje u jednu stranu. Drugi i treći segment prikazuju rezultate ponašanja sistema u slučaju kada je uključen mehanizam sinhronizacije po brzini. Vidljivo je očekivano ponašanje sistema po kome slave jedinica nakon nekog vremena dostiže brzinu master mašine što za rezultat ima pravolinijsko kretanje eksperimentalne postavke. Logično, povećanjem brzine dolazi do povećanja vrednosti kontra elektromotorne sile, a samim tim i vrednosti napona. Zapaža se i usklađena promena odziva struje što je posledica dinamične promene zahteva za momentom na vratilu mašine u toku kretanja.



Slika 13. Odzivi sistema – napon, struja, brzine

5. ZAKLJUČAK

U ovom radu analiziran je jedan od mogućih načina kako se danas u industriji rešavaju najkompleksniji zahtevi i to zahtevi regulisanja položaja pogonskog vratila. Prikazano je rešenje eminentnog i svetski priznatog proizvođača Danfoss.

Sprovedenim eksperimentima su pokazane dobre strane korišćenog rešenja ali su i uočene pojedini problemi koji mogu uneti probleme prilikom korišćenja ovakvih uređaja. Kao osnovni problem, tokom rada su prepoznata podešavanja korišćenih regulatora. Sprovedeni eksperimenti i realizovane eksperimentalne postavke predstavljaju dobru polaznu tačku za nastavljajanje istraživanja u oblasti pozicione kontrole.

6. LITERATURA

- [1] Darko Marčetić: „Mikroprocesorsko upravljanje energetskim pretvaračima“, Fakultet tehničkih nauka, Novi Sad, 2014.
- [2] Veran Vasić, Đura Oros, „Energetska elektronika u pogonu i industriji“, Fakultet tehničkih nauka, Novi Sad, 2012.
- [3] S. Vukosavić: „Digitalno upravljanje električnim pogonima“, Akademska Misao, Beograd, 2003.
- [4] „Design Guide FC 302 Automation Drive“, dostupno na: <https://www.galco.com> (pristupljeno u septembru 2020.)
- [5] „VLT MCO 305 Design Guide“, dostupno na: <http://files.danfoss.com> (pristupljeno u septembru 2020.)
- [6] D. Milićević, N. Vukajlović: „Energetska energetika u pogonu i industriji praktikum“, Fakultet tehničkih nauka, Novi Sad, 2019.
- [7] „Programing Guide Automation Drive FC 301/302“ dostupno na: <https://files.danfoss.com/download/Drives/MG33MO02.pdf> (pristupljeno u septembru 2020.)
- [8] „VLT Motion Control MCO 305“, dostupno na: http://files.danfoss.com/download/Drives/doc_MG33K302.pdf (pristupljeno u septembru 2020.)

Kratka biografija:



Rade Jerkić rođen je u Zvorniku 1996. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Energetska elektronika i električne mašine odbranio je 2019.god.
kontakt: rade.jerkic@hotmail.com



Dragan Milićević rođen je u Tuzli 1977. god. Doktorirao je na Fakultetu tehničkih nauka 2014. god., a od 2019. god. je zvanju vanredni. Oblast interesovanja je energetska elektronika, regulisani elektromotorni pogoni i obnovljivi izvori električne energije.
Kontakt: draganmilicevicftn@gmail.com

IZAZOVI U RAZVOJU iOS APLIKACIJA VISOKE UPOTREBLJIVOSTI**CHALLENGES IN THE DEVELOPMENT OF HIGH USABILITY iOS APPLICATIONS**

Olga Savić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu opisani su izazovi sa kojima se susreću programeri iOS mobilnih aplikacija prilikom njihovog razvoja. Istraživanje se vrši sa aspekta upotrebljivosti, a uključeni su i različiti načini za poboljšanje efikasnosti, radi kreiranja iOS aplikacija visokog kvaliteta.

Ključne reči: Izazov, iOS, aplikacija, razvoj, upotrebljivost

Abstract – This paper describes the challenges, which iOS mobile application developers face during development. The research is done from the aspect of usability, and it includes different ways of improving efficiency, in order to create high-quality iOS applications.

Keywords: Challenge, iOS, application, development, usability, user

1. UVOD

Analiza izazova koji se javljaju prilikom razvoja iOS aplikacija visoke upotrebljivosti jeste tema ovog rada. U toku kreiranja različitih platformi, kao i raznih pojedinačnih aplikacija može da dođe do potencijalnih problema, koje je potrebno otkloniti ili makar prilagoditi korisniku, tako da date aplikacije ili platforme postanu u potpunosti upotrebljive.

Ovaj rad sastojace se iz sedam poglavlja: uvod, motivacija problema, heuristika upotrebljivosti za mobilne aplikacije, koraci u razvoju iOS aplikacija, izazovi u razvoju iOS aplikacija, uloga korisnika i najbolje prakse i zaključak. Motivacija i opis problema je deo rada koji će se baviti razlozima zbog kojih je jedno ovakvo istraživanje potrebno, kao i motivima za pisanje istog. U trećem poglavlju biće sprovedeno istraživanje o heuristikama upotrebljivosti za mobilne aplikacije. Razvoj svake iOS aplikacije sastoji se iz više koraka koje je potrebno sprovesti, kako bi se postigao krajnji cilj, što je obrađeno u poglavlju četiri. Proces kreiranja aplikacija namenjenih za iOS operativni sistem i izazovi prilikom tog kreiranja biće opisani u okviru poglavlja broj pet. Poglavlje šest namenjeno je za opis uloge korisnika i najboljih praksi. Na kraju rad će se baviti analizom dobijenih rezultata i informacija, odnosno eventualnim nedostacima, kao i načinima da se ti nedostaci nadoknade.

2. MOTIVACIJA I OPIS PROBLEMA

Razvoj mobilnih aplikacija je u rapidnom porastu, zbog povećane popularnosti pametnih telefona.

NAPOMENA:

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

Sa milijardama preuzetih aplikacija, Apple-ova prodavnica (*App Store*) i Google-ova prodavnica (*Google Play*) su uspele da nadjačaju mobilne uređaje. Tokom poslednjih deset godina, došlo je do neprekidnog naglog porasta u objavljivanju i preuzimanju različitih aplikacija sa *App Store*-a.

Trenutno, najzastupljenije platforme su Android i iOS. Ipak, ovaj uspeh izaziva programere da stvaraju aplikacije visokog kvaliteta, da bi privukli i zadovoljili krajnje korisnike. Praktično je nemoguće odrediti tačan broj dostupnih iOS aplikacija, jer postoje razni načini za preuzimanje i objavljivanje.

2.1. Zašto baš iOS aplikacija?

Izrada iOS aplikacija namenjena je korisnicima *iPhone*-a – mobilnih uređaja kompanije *Apple*. Gledajući statistike, primetan je veći broj korisnika Android operativnog sistema. Ipak, činjenica je da aplikacija treba da bude dostupna svima, pa je potrebno obezbediti isti kvalitet korisničkog iskustva i za korisnike *iPhone*-a, pogotovo uzimajući u obzir nezanemarljive prednosti iOS platforme.

Kao prvo, generalno gledano, iOS pruža mogućnost bržeg i kontinualnijeg funkcionisanja, zahvaljujući visokom nivou optimizacije.

Zatim, dizajn aplikacija je veoma privlačan, lake su za pronalazak, korišćenje i ocenjivanje. Gledajući proces korišćenja aplikacija, kod iOS-a je jasno vidljiva mogućnost jednostavnijeg ocenjivanja i preporuke, upravo zbog prisustva manjeg broja dostupnih uređaja, zbog čega je lakše i održati konzistentnost. Čak i iz ugla programera, iOS je veoma često radiji izbor.

2.2. Specifičnosti korisnika iOS aplikacija

Osvrćući se na period od pre 10-15 godina, kada pametni telefoni praktično nisu ni postojali, niko nije mogao zamisliti kakav će se napredak tehnike u međuvremenu dogoditi. Danas, Android više nije privilegija, već skoro svako može sebi priuštiti jedan pametan telefon sa ekranom osetljivim na dodir. Veruje se da odgovarajuće zajednice korisnika imaju jasne razlike i specifičnosti, od lične komunikacije i karakteristika, pa sve do svakodnevnih navika. *Apple* korisnici su veoma lojalni, kupuju više aplikacija i znatno češće se angažuju oko svojih uređaja nego na primer korisnici Android-a, koji se neretko odlučuju za piratske verzije. Takođe, korisnici *iPhone*-a su većinom više profilisani i prilagođeni filozofiji aplikacija. Razlika je u ceni, odnosno u tome da su iOS korisnici ipak spremni da potroše veću količinu novca za nešto što će im doneti veću vrednost. Takođe karakteristika korisnika iOS aplikacija jeste visok procenat mladih, visokoobrazovanih ljudi sa u određenoj meri većom platežnom sposobnošću.

3. HEURISTIKA UPOTREBLJIVOSTI

Upotreba pametnih telefona bitno napreduje na današnjem tržištu, što je dovelo do potrebe za ubrzanim razvojem i poboljšanjem aplikacija napravljenih za ove uređaje. Sa progresom u korišćenju mobilnih uređaja i njihovih aplikacija, dolazi do stalne pojave novih izazova i određene jedinstvenosti, kao što je upotrebljivost, moraju biti dodatno istražene i ispitane. Upotrebljivost je definisana kao „kapacitet koji je moguće iskoristiti“, to jest, maksimalan kapacitet koji određeni uređaj poseduje [1]. Tokom razvoja proizvoda, upotrebljivost može biti ispitana putem inspekcije upotrebljivosti ili različitih testova upotrebljivosti. Upotrebljivost je definisana pomoću pet komponenta kvaliteta:

- vreme učenja
- efikasnost
- lakoća pamćenja
- greške
- zadovoljstvo

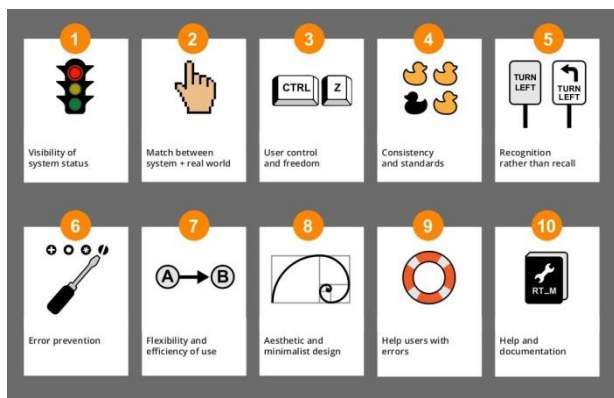
3.1. Evaluacija po heuristikama

Evaluacija po heuristikama ima za cilj da oceni proizvod na osnovu principa, odnosno heuristika upotrebljivosti, pri čemu se vrši utvrđivanje šta je ispravno i ispoštovano, a sa druge strane, šta nije.

Prilikom evaluacije upotrebljivosti mobilnih uređaja, u obzir se moraju uzeti važne specifičnosti samih uređaja: korisnik najviše koristi šake i prste pri radu sa telefonom, uređajima se upravlja bežično i podržana je dodavanje novih aplikacija i internet konekcije za određene funkcionalnosti.

3.2. Heuristike upotrebljivosti

Heuristike upotrebljivosti nose ovaj naziv, jer predstavljaju važne i detaljne smernice za kreiranje razumljivog dizajna bliskog korisniku. Kao takve, prvi put su uvedene od strane Malone-a, 1982. godine, sa primenom isključivo kod probleme visokog nivoa u mobilnim igrama. Nadaleko poznate su Nielsen-ove heuristike (Jakob Nielsen, 1995), napisane sa glavnim osvrtnom na kvalitet samog proizvoda. Vodič za razvoj mobilnih aplikacija sadrži deset principa i u masovnoj je potrebi i dan danas:

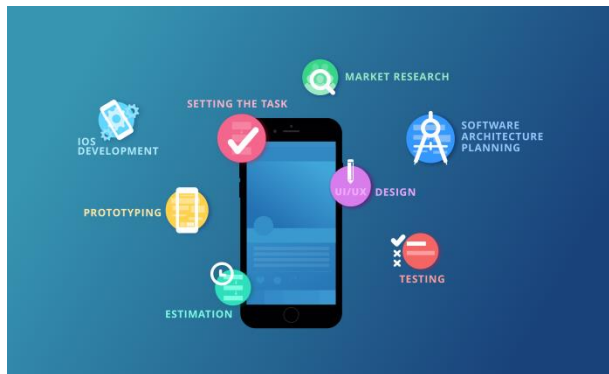


Slika 1. Grafički prikaz heuristika za mobilne telefone

- vidljivost statusa sistema
- ekvivalencija između sistema i realnog sveta
- korisnikova kontrola i sloboda
- konzistentnost i standardi
- prevencija grešaka
- prepoznavanje umesto pamćenja

- fleksibilnost i efikasnost korišćenja
- estetski i minimalistički dizajn
- pomoć korisnicima za utvrđivanje, prepoznavanje i oporavku od grešaka
- pomoć i dokumentacija

4. KORACI U RAZVOJU IOS APLIKACIJA



Slika 2. Prikaz koraka u razvoju iOS aplikacija

1. **Istraživanje** – Prvi korak u izradi iOS aplikacije je istraživanje i upoznavanje sa trenutnom situacijom na tržištu.

2. **Stvaranje konkretnih ideja** – Ideja predstavlja osnovu kvalitetne aplikacije. Masovno se koriste različiti modeli za prezentaciju modela, a neki od njih su: brainstorming, mape uma i skice. [2]

3. **Kreiranje prototipa** - Jednostavno rečeno, žičani okvir je inicijalni nacrt. Cilj je predstavljanje pojednostavljenog vizuelnog koncepta aplikacije, a ne objašnjavanje dizajna.

4. **UI/UX dizajniranje** - Krucijalan korak u razvoju iOS aplikacije, u kom je neophodno praćenje specifičnih smernica i uputstava. Faza dizajniranja omogućuje timu pregled funkcionalnosti ekrana, kao što su grafike, animacije i ikonice. Dizajniranje je važan gradivni blok naizgled komplikovanog procesa razvoja aplikacije.

5. **Planiranje arhitekture softvera** - Dok se faza dizajniranja fokusira na implementaciju funkcionalnosti aplikacije, planiranje softverske arhitekture se dešava paralelno sa ciljem da se obezbedi stabilnost i skalabilnost same aplikacije. 4+1 model gledišta za arhitekturu softvera: use case view, logički pogled, procesni pogled, implementacioni pogled, deployment view.

6. **Razvoj iOS aplikacije** – Razvoj određene aplikacije predstavlja temeljno kreiranje iste. Više o ovoj temi i izazovima biće objašnjeno u poglavlju broj 5.

7. **Testiranje i objavljivanje** – Proces provere ispravnosti dobijenog proizvoda uz postupak objave rezultata, pa i same aplikacije.

8. **Redovna ažuriranje** – Praćenje stanja aplikacije i popravljavanje nedovoljno kvalitetnih delova iste.

5. IZAZOVI U RAZVOJU IOS APLIKACIJA

Razvoj mobilnih aplikacija je relativno novi pojam, koji rapidno raste, zbog sve veće sveprisutnosti i popularnosti mobilnih telefona kod krajnjih korisnika. Postoje značajne razlike u tome da li telefoni koriste Android ili iOS operativni sistem, koje su posebno primetne u procesu razvoja aplikacija. U toku razvoja korisnik se susreće sa velikim brojem različitih izazova.

5.1. Opšti izazovi za programere kod iOS aplikacija

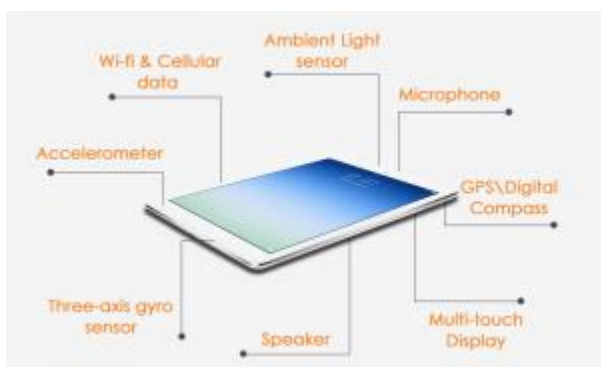
- dobijanje odobrenja od App Store-a
- fragmentacija umesto ujedinenja
- praćenje, analiza i podrška testiranju
- otvorene/zatvorene platforme za razvoj
- kreiranje univerzalnog korisničkog interfejsa
- količina podataka
- održavanje koraka sa čestim promenama
- održavanje
- buduće provere aplikacije

5.2. Specifičnosti pametnih telefona na iOS platformi

Pametni telefoni na iOS platformi imaju veliki broj specifičnosti među kojima su: multi-touch uređaj, relativno jeftin uređaj visokih performansi, korisnički interfejs, problem unosa teksta, preciznost adesiranja i velicina objekta, vizuelno prekrivanje, ivice ekrana, objekti van ekrana u fokusu, interakcija sa drugim aplikacijama i uparivanje senzorima.

Preciznost adresiranja i veličina objekta - U slučaju telefona sa iOS platformom, postoji problem sa preciznim adresiranjem manjih objekata. Ovo je posebno primetno u aplikacijama koje pružaju neke lokacijske usluge, pa se radi poboljšanja koriste dvostruko kuckanje (*TapTap*) i namagnetisanog štapa (*MagStick*). Znatno napredak se uviđa u pogledu objekata koji se nalaze na rubu ekrana.

Upravljanje senzorima - Svaki Apple uređaj ima sledeće senzore: žiroskop sa 3 ose, akcelerometar, Wi-fi prenos podataka, senzor za ambijentalno svetlo, mikrofoni, GPS/digitalni kompas, zvučnik, multi-touch ekran.

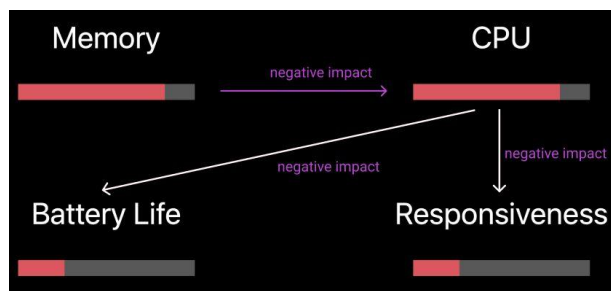


Slika 3. Prikaz senzora na iOS uređaju

5.3. Hardverska i softverska ograničenja

U ograničenja hardverskog i softverskog tipa spadaju: mala veličina ekrana i niske rezolucije, ograničena tastatura, nivo procesorske snage i količina memorije, trajnost baterije, raznolikost tehnologije i sadržaja, nedostatak kompatibilnosti aplikacija i bezbednost.

Nivo procesorske snage i količina memorije - Sa ciljem što verodostojnijeg prikaza i grafike u mobilnim aplikacijama, a pogotovo u igrama, uvode se jači procesori i veća memorija. Primetan je napredak u memorijskim resursima, ali je i dalje primetna ogromna razlika u poređenju sa računarima. Međutim, opcija priključenja eksternih memorijskih modula dovela je do znatnog poboljšanja performansi., što je i bio cilj.



Slika 4. Veza memorije, procesora, baterije i odgovora

Izazovi u razvoju iOS aplikacija još mogu biti: razvoj sa više platformi, trenutne prakse za testiranje kao i izazovi analize i testiranja [3].

6. ULOGA KORISNIKA I NAJBOLJE PRAKSE

Recenzije korisnika igraju veoma važnu ulogu u industriji industriji mobilnih telefona. Korisnici, programeri i tržište aplikaciju imaju korist od ocena i komentara korisnika. Postojeće metode za ostavljanje recenzija nisu odgovarajuće za kranje korisnike aplikacija. U međuvremenu, korisnici mogu ostaviti svoju povratnu informaciju na različitim nivoima kvaliteta, što otežava proizvođačima aplikacija izdvajanje korisnih informacija za održavanje softvera. Cilj ovog poglavlja jeste poboljšanje kvaliteta i upotrebljivosti iOS mobilnih aplikacija, na osnovu iskustva korisnika.

Biće predstavljene različite metode za prikupljanje povratnih informacija, uz analizu iskoristivosti ovih metoda. Dodatno, primena metode za dizajniranje, koja stavlja korisnika u prvi plan (*user-centered design method*) pokazala je veoma povoljne rezultate u kreiranju aplikacija.

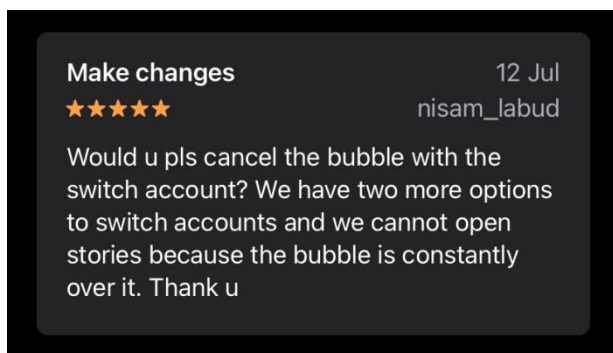
Ukoliko bi postojala mogućnost da programeri se u razvoju aplikacija u potpunosti pridržavaju smernica koje će biti pomenute, kao ishod bi se dobila skoro idealna aplikacija, u potpunosti po meri korisnika.

6.1. Recenzija korisnika

Komentari i ocene korisnika su klasifikovani u različite tipove, u zavisnosti od sadržaja. Stoga, postoje i odgovarajući načini za prikupljanje recenzija korisnika, koji koriste mobile aplikacije.

Recenzija korisnika predstavlja povratnu informaciju u tekstualnoj formi, napisanu od strane potrošača i korisnika proizvoda, nakon određenog perioda korišćenja. Korisnici prezentuju svoje mišljenje sa različitih tačaka gledišta. Predstavlja ne samo pohvalu ili kritiku posmatranog objekta, već i izveštaj o korisnikovom korišćenju u određenom kontekstu. Važno je naglasiti da se recenzija obično ostavlja dobrovoljno.

U softverskoj industriji korisnici imaju veliku ulogu u raznim metodama razvoja. Tipovi mogućih recenzija su: prijave greške, budući zahtevi i korisničko iskustvo. Tip recenzije koji opisuje probleme primećene u radu sa aplikacijom, u slučaju neočekivanog ponašanja jeste prijava greške. Pod buduće zahteve spada recenzija koji uključuje sugestije za eventualna poboljšanja aplikacije, kao što je dodavanje funkcionalnosti u narednim verzijama [4].



Slika 5. Primer recenzije sa predlogom za poboljšanja

Korisničko iskustvo se sastoji iz više pohvala i žalbi i predstavlja stepen zadovoljstva krajnjeg korisnika u toku rada.

6.2. Dizajn fokusiran na korisnika

Dizajn predstavlja kreativnu aktivnost, koja donosi nešto novo, što nije postojalo ranije. Dizajn aplikacije je važan korak u procesu razvoja same aplikacije i uključuje definisanje komponenti, interfejsa i drugih karakteristika. Norman (1988) je razgradio koncept *User-centered design-a* (UCD) i sugerisao postavljanje korisnika i njegovih zahteva na centralno mesto u dizajnu. Potrebno je obezbediti jednostavno i brzo usvajanje nove aplikacije, bez ulaganja dodatnog truda na proces učenja. Cilj je utvrditi kako krajnji korisnici funkcionišu, bez prinude na promenu. UCD za razvoj mobilnih aplikacija se ne razlikuje mnogo od klasičnog UCD-a.

6.3. Najbolje prakse u razvoju iOS mobilnih aplikacija

Postoji pet faza u ciklusu razvoja iOS mobilnih aplikacija, sa stanovišta dizajna fokusiranog na korisnika. Podrazumeva se da je razvoj cikličan i da aplikacije prolaze kroz više iteracija tokom svog životnog ciklusa. Prvo se postavlja pitanje zašto je aplikacija koja se razvija neophodna, koji je razlog njenog nastanka i koje probleme rešava. Takođe, vrši se istraživanje da li postoji aplikacija slične namene i utvrđivanje nedostataka. Nakon toga vrši se dizajniranje i razvijanje same aplikacije. Neophodno je odrediti prioritete za ciljanu iOS platformu i prodavnicu aplikacija. Nakon razvoja, vrši se pregled i dorada aplikacije, do potpunog ispunjenja svih inicijalno postavljenih zahteva.

Koraci:

1. Procena trenutne situacije – tržišta, ciljanih korisnika i aplikacija sličnog tipa.
2. Razumevanje krajnjeg korisnika
3. Šta će aplikacija biti u mogućnosti da ponudi?
4. Razmatranje dizajna mobilnog telefona
5. Pregled i prerada

7. ZAKLJUČAK

Kreiranje i implementacija mobilnih i računarskih aplikacija je složen proces sa velikim brojem komplikacija i detalja na koje treba obratiti pažnju. Naročito se ističu velika raznovrsnost mobilnih tehnologija, kao i značajne razlike ljudskih osobina krajnjih korisnika.

Neophodno je posvetiti posebnu pažnju oblikovanju kvalitetne korisničke podrške i efikasnih modela interakcije, a jedan od glavnih imperativa i motivacionih faktora mora biti izgradnja mobilnih aplikacija na visokom nivou upotrebljivosti. Upotrebljivost mobilnih aplikacija još uvek nije dostigla odgovarajući stepen važnosti prilikom uspostavljanja kriterijuma za razvoj aplikacije. Ipak, svetlo na kraju tunela predstavlja činjenica da se sve više posvećuje pažnja tim problemima, kao i to da je uspostavljen neki krajnji cilj, a to je da se podizanjem nivoa upotrebljivosti aplikacija obezbede mobilne usluge jednako ili makar približno dostupne svim grupama korisnika.

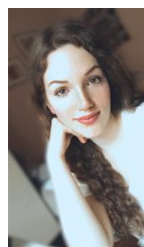
U okviru ovog rada sprovedena je opsežna analiza izazova koji se javljaju prilikom razvoja aplikacija visoke upotrebljivosti za iOS operativni sistem. Akcenat je bio na analizi svih mogućih izazova kako u smislu koraka u razvoju, tako i u smislu eventualnih poteškoća prilikom projektovanja funkcionalnih aplikacija. Jedan deo rada posvećen je ulozi korisnika i njihovom značaju pri samoj izradi određenog programa. Takođe, obrađene su i heuristike na osnovu kojih se proverava upotrebljivost aplikacije. Ovaj deo raden je da bi se utvrdili uslovi i izazovi koji treba da se ispune da bi se kreirala kvalitetna funkcionalnost visoke upotrebljivosti.

Ovaj rad ima široku primenu u realnom sistemu. Kompletan istraživanje može u velikoj meri da pomogne pri izradi kvalitetne iOS aplikacije visoke upotrebljivosti. Na kraju, istraženi su i potencijalni tokovi razvoja iOS aplikacija u budućnosti i jasno je da će fokus biti na augmentovanoj stvarnosti, mašinskom učenju, obezbeđivanju "pametnih kuća" i još jednostavnijem izvršavanju platnih transakcija putem telefona.

8. LITERATURA

- [1] Marcos Antonio Duraes Dourado and Edna Dias Canedo, *Usability Heuristics for Mobile Applications*
- [2] Kristin Lee Wood, *Ideation Methods: A First Study with Measured Outcomes with Personality type*
- [3] Mona Erfani Joorabchi, Ali Meshbah, Philippe Krutchen, *Real Challenges in Mobile App Development*, University of British Columbia
- [4] F.T.C and Vasa, R. 2011. *Toward a Social Media Usage Policy*

Kratka biografija:



Olga Savić rođena je 1996. godine u Novom Sadu. Završila je gimnaziju „Svetozar Marković“ takođe u Novom Sadu 2015. godine. Diplomirala je na Fakultetu tehničkih nauka u Novom Sadu 2019. godine i iste godine je upisala master studije na smeru Softversko inženjerstvo i informacione tehnologije.

kontakt: lola.savi4@gmail.com

PREDIKCIJA BILLBOARD HITOVA NA OSNOVU AUDIO I TEKSTUALNIH OBELEŽJA PESAMA**BILLBOARD HIT SONG PREDICTION BASED ON AUDIO AND TEXT FEATURES**Sandra Rajanović, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu prikazano je kreiranje sistema za predikciju hitova na Billboard top-listama, koristeći audio obeležja i tekstove pesama. Rešenje problema binarne klasifikacije implementirano je uz pomoć više modela mašinskog učenja i dve tehnike obrade prirodnog jezika, a realizovano je u Python programskom jeziku.

Ključne reči: mašinsko učenje, predikcija hit pesama, obrada prirodnih jezika

Abstract – This paper presents creating a system for Billboard hit song prediction using their audio features and lyrics. The binary classification problem solution was implemented using multiple machine learning models and two natural language processing methods, all created in Python programming language.

Keywords: machine learning, hit song prediction, natural language preprocessing

1. UVOD

Hit Song Science se bavi pretpostavkom da se veliki broj popularnih pesama odlikuje istim obeležjima na osnovu kojih se može proceniti da li neka pesma ima šanse da postane hit ili ne [1]. Sa razvojem tehnologije, *state-of-the-art* modeli postaju sve bolji, te ova oblast pronalazi primenu i u komercijalne svrhe. Sa digitalizacijom muzičke industrije i pojavom *streaming* platformi dolazi do promena u načinu konzumiranja muzike, što dovodi do ogromnih količina muzičkog sadržaja i velike konkurencije, gde je jedan od načina izdavanja uspešna hit pesma.

Kreiranjem modela koji bi mogao da predvidi kakve pesme imaju veću šansu da postanu popularne i koje karakteristike na to imaju uticaja, drugim izvođačima bi bilo omogućeno da uključe te karakteristike u svoja dela i povećaju šanse za svoj uspeh. Popularne pesme donose veliku finansijsku dobit, pa bi dovoljno precizan model bio od ogromnog značaja, jer bi omogućio izvođačima i muzičkim kućama da na potencijalne hitove usmere svoje finansijske resurse, i tako ostvare najveću moguću zaradu. Danas se standardom u proceni popularnosti smatraju top-liste gde se pesme rangiraju na osnovu slušanosti, prodaje i puštanja na radiju, a najpoznatije su *Billboard* top-liste. Ovde je najznačajnija *Billboard Hot 100* lista koja predstavlja dosta realnu sliku trenutno aktuelne muzike.

NAPOMENA:

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

Stoga, pesme sa ove liste mogu da predstavljaju reprezentativan uzorak hit pesama za analizu.

Tema ovog rada je predikcija *Billboard* hitova na osnovu obeležja pesama koja čine audio i tekstualna obeležja. Ulazni skup podataka čine pesme nastale u poslednjih deset godina, gde su popularne pesme uzete sa *Billboard Hot 100* listi, a ostatak pesama je izabran nasumično. Dodatni manji podskup podataka je kasnije kreiran, koji sadrži samo hitove sa top 40 pozicija top-listi. Rad se sastoji iz dva različita pristupa treniranju modela, gde se prvi ogleda u korišćenju *topic* modela i četiri različita modela mašinskog učenja, dok drugi koristi BERT model [2] i neuronske mreže. Oba rešenja su evaluirana na isti način koristeći tri mere – tačnost (eng. *accuracy*), preciznost (eng. *precision*) i odziv (eng. *recall*).

2. ARHITEKTURA REŠENJA

Proces modelovanja softverskog rešenja sastoji se iz tri celine. Prva obuhvata prikupljanje podataka, njihovo pretprocesiranje i podelu na trening i test skup. Druga celina se bavi dvema različitim implementacijama rešenja i obuhvata optimizaciju parametara, a zatim i treniranje različitih modela. U trećoj celini testiraju se najbolji modeli. Slika 1 prikazuje šemu celokupnog procesa.

Za prikupljanje podataka korišćena su tri izvora i formirane su dve verzije skupa podataka i njihovih podskupova. Prvoj verziji su pripojena dodatna obeležja dobijena pomoću *topic* modela, a drugoj obeležje teksta pesme. Pretprocesiranje se sastoji se od skaliranja obeležja na opseg između 0 i 1.

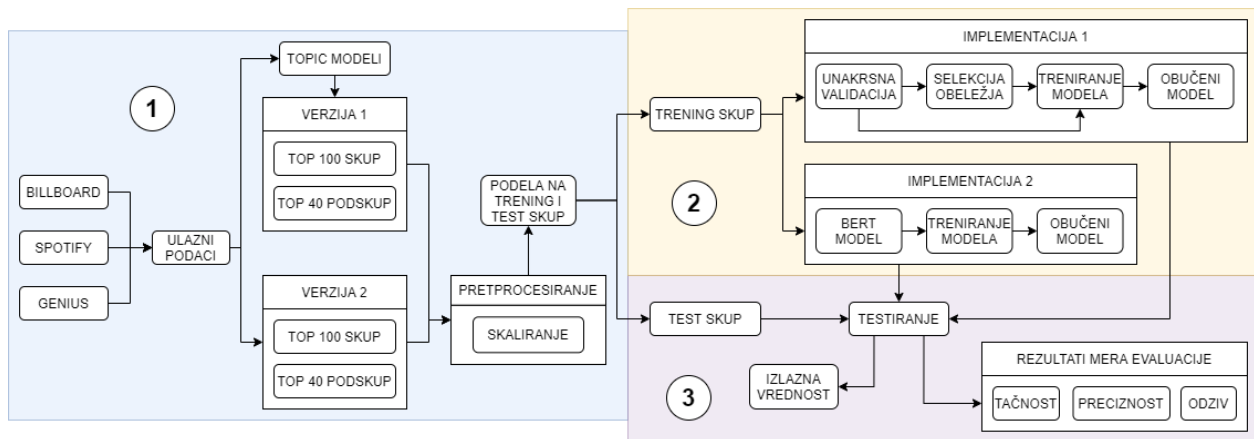
U drugom delu, prvi način implementacije obuhvata optimizaciju parametara pomoću unakrsne validacije za četiri modela mašinskog učenja, a nakon toga i selekciju obeležja. Drugi način implementacije podrazumeva korišćenje BERT modela za procesiranje tekstualnih obeležja i treniranje modela neuronskih mreža.

Poslednji deo sastoji se od evaluacije performansi obučanih modela nad testnim podacima.

3. PRIKUPLJANJE I OBRADA PODATAKA

Za potrebe ovog rada prikupljeni su podaci sa *Billboard Hot 100* top-liste **Error! Reference source not found.**, *Spotify* platforme [4] i *Genius* baze podataka [5], čijim spajanjem je kreiran skup podataka sa 22 obeležja. Kreirane su dve verzije skupa – glavni skup koji sadrži 9126 instanci, i njegov podskup od 3390 numera.

Ciljna labela ima vrednosti – 1 ako je hit, i 0 nije. Oba skupa sadrže približno jednak broj pesama iz obe kategorije radi balasiranosti u skupu podataka.



Slika 1. Šema arhitekture rešenja

3.1. Opis skupa podataka

Prikupljeni podaci su iz vremenskog perioda od 2010. do 2019. godine. Podaci o hit pesmama i njihovim izvođačima su dobavljeni sa *Billboard Hot 100* liste kao u radovima [6] i [7]. Ove liste se ažuriraju jednom nedeljno, pa je na raspolaganju bilo njih 530, tj. 53000 pesama. Nakon uklanjanja duplikata i pesama za koja nisu bila dostupna audio obeležja, dobijen je skup od 4558 pesama. Po ugledu na pristup u radu [8], ne-hitovi su definisani kao sve ostale pesme onih izvođača koji imaju bar jednu pesmu na top-listama. Pomoću *Spotify* API-a dobavljene su sve njihove ostale pesme izdate u posmatranom vremenskom periodu. Ovim procesom prikupljeno je 173227 pesama, od kojih je izdvojen manji podskup od 4568 pesama, kreiran vodeći računa o balansiranosti podataka među klasama. Konačan skup svih podataka brojao je 9126 instanci.

Pomoću *Spotify* API-a dobavljena su audio obeležja svih pesama konačnog skupa, a zatim su koristeći *Genius* API prikupljeni tekstovi ovih pesama. Određena količina pesama u skupu nije na engleskom jeziku, pa je za njih sa interneta ručno preuzet engleski prevod tekstova.

U cilju analiziranja da li je lakše predvideti hitove sa vrha top-lista, kreiran je podskup originalnog skupa za koji se izdvajaju samo pesme koje su se našle u top 40 pozicija. Na ovaj način je dobijeno 1695 hit pesama, a njima je pridruženo isto toliko nasumično izabranih ne-hitova iz prvog skupa, te je konačan broj instanci iznosio 3390.

3.2. Obeležja i labela

Skup podataka ima dve verzije, gde je prva opisana sa 22, a druga sa 19 obeležja, a oba imaju istu izlaznu labelu. Obe verzije skupa imaju obeležja koja mogu biti podeljena u dve grupe: 13 obeležja dobavljenih pomoću *Spotify* API-a i 5 dodatnih boolean vrednosti. Pored toga, obe verzije sadrže i dodatna obeležja koja reprezentuju tekstove pesama.

Audio obeležja pesama, slično kao u radovima [6], [7] i [8], obuhvataju:

- trajanje pesme
- tonalitet
- mod
- promena taktova
- akustičnost
- plesnost
- energija

- instrumentalnost
- živost
- glasnost
- govornost
- valentnost
- tempo

Naredna grupa obeležja obuhvata 5 *boolean* obeležja dodatnih na osnovu uočenih istorijskih trendova kao potencijalni faktori u popularnosti pesama. U nedostatku lakšeg načina za dodelu ispravnih vrednosti ovih obeležja, ceo skup je ručno anotiran. Ovde spadaju sledeći indikatori:

- pesma je originalno na jeziku koji nije engleski
- pesmu izvodi popularan izvođač
- obrada pesme
- božićna pesma
- remiks pesme

Poslednja četiri obeležja u prvoj verziji skupa imaju vrednosti između 0 i 1, a dobijena su kao rezultati *topic* modela, po ugledu na rešenje u radu [9]. Ove vrednosti predstavljaju verovatnoće da numera pripada toj temi na osnovu sadržaja teksta pesama. Druga verzija skupa umesto ovih obeležja koristi samo jedno tekstualno obeležje koje sadrži ceo tekst pesme, i obrađuje se kasnije u toku implementacije.

4. IMPLEMENTACIJA SISTEMA

Rešenje problema implementirano je u programskom jeziku *Python*. Podaci su dobavljeni iz tri izvora koristeći biblioteke *billboard*, *spotipy* i *lyricsgenius*. Pomoću *billboard* biblioteke su dobavljene sve *Billboard Hot 100* top-liste u vremenskom periodu od 2010. do 2019. godine. Biblioteka *spotipy* predstavlja enkapsulaciju *Spotify* API-a, a korišćena je prvo za pronalaženje svih hit pesama i njihovih izvođača radi dobijanja njihovih *Spotify* ID-eva. ID-evi izvođača se zatim koriste za dobavljanje njihovih ostalih pesama, a potom se pomoću ID-eva pesama preuzimaju njihova audio obeležja. Biblioteka *geniuslyrics* koja takođe predstavlja enkapsulaciju *Genius* API-a, koristi se za prikupljanje tekstova pesama.

Topic modeli implementirani su pomoću biblioteka *nlTK*, *gensim* i *scikit-learn*. Biblioteka *nlTK* korišćena je za tokenizaciju tekstova i izbacivanje stop reči, a *gensim* korišćen je za određivanje vrednosti *perplexity* mere koja služi za određivanje optimalnog broja tema, pomoću koje je izabran broj 4 kao rešenje. Izabrani *topic* model je

Latentna Dirihleova alokacija (eng. *Latent Dirichlet allocation*, skraćeno LDA) koji se implementira koristeći *scikit-learn* biblioteku. Za njegovo kreiranje korišćene su klase *TfidfVectorizer* i *LatentDirichletAllocation* pomoću kojih se dobijaju verovatnoće pripadanja svakoj od tema za svaki dokument. Kasnija treniranja modela su izvršena sa i bez ovih obeležja radi analize i poređenja rezultata.

Potom se skaliraju vrednosti određenih obeležja, a zatim se skup deli na trening i test skup u razmeri 80:20.

U okviru prvog načina implementacije realizovana je unakrsna validacija, selekcija obeležja i treniranje četiri modela mašinskog učenja. Unakrsna validacija se koristi radi nalaženja najoptimalnijih parametara za treniranje, a izvedena je pomoću klasa *KFold* i *GridSearchCV* iz *scikit-learn* biblioteke. U ovom procesu trening skup se deli na pet delova i definišu se parametri za koje će svaki model biti treniran. Na taj način su izabrani najpovoljniji parametri za treniranje četiri modela – logistička regresija, *random forest*, metoda potpornih vektora (eng. *Support Vector Machine*, skraćeno SVM) i *gradient boosting*. Za svaki od njih, unakrsna validacija je izvršena po četiri puta, odnosno jednom za svaku verziju skupa podataka – originalni skup, originalni skup bez *topic* obeležja, podskup originalnog skupa i podskup bez *topic* obeležja.

Model logističke regresije je kreiran koristeći klasu *LogisticRegression*, a u procesu unakrsne validacije posmatrani su sledeći parametri: *penalty*, *solver*, *class_weight*, *multi_class* i *max_iter*.

Random forest model je implementiran pomoću klase *RandomForestClassifier*, a optimizovani su mu naredni parametri: *n_estimators*, *criterion* i *max_features*.

Za model SVM, kreiran pomoću klase SVC, sledeći parametri su optimizovani: *C*, *gamma* i *kernel*.

Na kraju, za *gradient boosting* model, implementiran koristeći *GradientBoostingClassifier* klasu, sledeći parametri su trenirani u unakrsnoj validaciji: *loss*, *n_estimators*, *criterion* i *max_features*.

Nakon dobijanja najboljih parametara, vrši se treniranje nad trening i predikcija nad test skupom. Pored ovoga je implementirana i selekcija obeležja koja je primenjena radi optimizacije rešenja, a realizovana je koristeći *SelectFromModel* klasu *scikit-learn* biblioteke. Potom se ponovo vrši treniranje modela sa istim parametrima.

Drugi način implementacije rešenja ostvaren je preko BERT modela i modela neuronskih mreža. BERT je tehnika mašinskog učenja baziranu na transformer modelima dubokog učenja. Biblioteka *transformers* dolazi sa pretreniranim BERT modelima koji su bili upotrebljeni za implementaciju rešenja, zajedno klasama *BertTokenizer* i *BertModel*. *BertTokenizer* služi za tokenizaciju tekstova i pripremu ulaznih informacija za model, a njegova izlazna vrednost predstavlja ulaz za klasu *BertModel* koja dati ulaz transformiše u vektorske reprezentacije reči. Izlazu iz ovog modela se redukuje dimenzionalnost i tako se dobija dvodimenzionalna matrica gde je svaki ulazni podatak predstavljen nizom dužine 768. Nakon toga se ovim podacima dodaje i 18 postojećih audio obeležja.

Modeli neuronskih mreža kreirani su uz pomoć *keras* biblioteke, a treniranje je izvršeno i sa svim obeležjima, i samo sa tekstualnim obeležjima iz BERT modela. Struktura modela, broj slojeva i dimenzionalnost određeni su empirijski. U modelima su korišćeni potpuno povezani

slojevi predstavljeni objektima klase *Dense*, a za svaku verziju skupa podataka, dodate su različite kombinacije slojeva - uglavnom su u pitanju troslojne strukture sa aktivacionim funkcijama *relu* i *sigmoid*. Ulazni podaci su dimenzija 786 za skup sa svim obeležjima i 768 za skup sa samo tekstualnim. Nakon dodavanja slojeva, vrši se kompajliranje svakog modela sa *binary_crossentropy* funkcijom gubitka i optimizacionom funkcijom *adam*, a treniranje se vrši u 150 epoha sa delovima veličine 10.

5. VERIFIKACIJA

Verifikacija obuhvata procenu kvaliteta performansi modela, a sprovodi se uz pomoć tri mere evaluacije nad testnim skupom podataka. Nakon toga su analizirane greške modela i mogući razlozi za njihovo pojavljivanje.

5.1. Rezultati

Mere evaluacije primenjene u radu su tačnost, preciznost i odziv. Za obučavanje i testiranje modela korišćen je skup od 9126 i njegov podskup od 3390 pesama, oba podeljena u razmeri 80:20.

Tabela 1 predstavlja rezultate modela nad celim skupom podataka. Skoro svi modeli imaju dosta slične rezultate i na trening i na test skupu, dok su u nekim slučajevima performanse čak i bolje na testnom nego na trening skupu. Izuzetak je *random forest* model, kod kojeg su rezultati nad trening skupom skoro savršeni, ali su na testnom značajno slabiji, što može ukazivati na *overfitting* modela. I pored ovoga, ova model je dao najbolje rezultate, zajedno sa neuronskim mrežama i *gradient boosting* modelom. Logistička regresija i SVM imaju dosta niže odzive što stvara veću razliku između vrednosti preciznosti i odziva, te može biti znak da model prediktuje veliki broj lažno negativnih instanci. Neuronske mreže imaju primetno bolju meru odziva od ostalih, što smanjuje broj lažnih negativnih instanci.

Za prva četiri modela treniranje je izvršeno i bez *topic* obeležja što je dalo skoro identične rezultate, pa se može pretpostaviti da tekstovi pesama nemaju mnogo uticaja na popularnost pesama, ili da *topic* modeli nisu dobro rešenje ovog problema. Takođe je izvršena i selekcija obeležja nad oba skupa (sa i bez *topic* obeležja), što je opet dalo identične rezultate. Razlog za ovo može biti to da izbačena obeležja svakako nisu imala veliki uticaj na konačni ishod, ili da su trenutni modeli dostigli svoj maksimum, te dodatne optimizacije nemaju efekta. Nad neuronskim mrežama obučavanje je izvršeno i samo sa tekstualnim obeležjima što je dalo dosta lošije rezultate od prethodnih, te je jasno da predikcija samo na osnovu tekstova pesama nije dovoljno dobra.

Tabela 2 prikazuje performanse nad podskupom podataka. Tačnosti svih modela su poboljšane za 3-5% u odnosu na rezultate sa celim skupom podataka. Mera odziva beleži primetno poboljšanje kod svih modela – porast se kreće u opsegu 7-14%, čime se smanjuje velika razlika između odziva i preciznosti, što je prethodno bio potencijalni izvor problema. Modeli *random forest* i neuronske mreže imaju generalno najbolje rezultate, a odmah nakon njih slede logistička regresija i *gradient boosting*. Model logističke regresije ima najveću tačnost i svaka mera evaluacije ima bolji rezultat nad testnim u odnosu na trening skup, dok model neuronskih mreža ima najbolje vrednosti odziva.

Tabela 1. Performanse modela nad celim skupom podataka

	Accuracy		Precision		Recall	
	Trening skup	Test skup	Trening skup	Test skup	Trening skup	Test skup
Logistic Regression	0.69589	0.69769	0.77700	0.78125	0.54854	0.54824
Random Forest	0.99986	0.72562	0.99972	0.75979	1.0	0.65899
SVM	0.69972	0.69989	0.80752	0.80434	0.52358	0.52741
Gradient Boosting	0.72945	0.70974	0.79040	0.76381	0.62369	0.60635
Neural Network	0.82369	0.70920	0.82112	0.71574	0.82720	0.69298

Tabela 2. Performanse modela nad podskupom podataka

	Accuracy		Precision		Recall	
	Trening skup	Test skup	Trening skup	Test skup	Trening skup	Test skup
Logistic Regression	0.73266	0.75516	0.78707	0.79522	0.63790	0.68731
Random Forest	0.99963	0.75073	1.0	0.76234	0.99926	0.72861
SVM	0.73045	0.75073	0.83783	0.82692	0.57153	0.63421
Gradient Boosting	0.79240	0.74778	0.83403	0.76415	0.73008	0.71681
Neural Network	0.92072	0.75516	0.90035	0.73441	0.94616	0.79941

5.2. Analiza grešaka modela

Analiza grešaka izvršena je poređenjem stvarnih i prediktovanih izlaznih vrednosti za svaki od modela. Kreirane su matrice zabuna (eng. *confusion matrix*) na osnovu kojih je ustanovljeno da uglavnom ima više lažno negativnih nego lažno pozitivnih, što je i očekivano s obzirom na to da modeli imaju veće preciznosti, a manje odzive. Izuzetak su modeli neuronskih mreža kod kojih su količine ovih instanci balansirane.

Od svih pogrešnih instanci izdvojene su samo one koje su modeli pogrešno predvideli sa pouzdanosću većom od 80%, gde je situacija sa lažnim instancama bila obrnuta, odnosno više je bilo lažno pozitivnih nego negativnih. Ovo je dosta nepovoljan ishod, jer bi s obzirom na svrhu ovog modela cena bila veća za lažno pozitivne rezultate, tj. za ne-hit pesme koje su pogrešno klasifikovane kao hitovi. Analizirane su vrednosti obeležja na kojima modeli najviše greše i uočeni su određeni trendovi gde većina modela greši na sličnim obeležjima. Računate su srednje vrednosti svakog obeležja za pogrešne u odnosu na tačno klasifikovane instance. Za ceo skup podataka, sledeća obeležja imaju drugačije vrednosti za pogrešne u odnosu na tačno prediktovane:

- hitovi: manja plesnost i valentnost, veća instrumentalnost i živost
- ne-hitovi: manja instrumentalnost

U slučaju podskupa podataka, uočeni su sledeći slučajevi:

- hitovi: manja plesnost, valentnost, energija, glasnost i instrumentalnost, veća živost
- ne-hitovi: veći mod, manja instrumentalnost i valentnost

6. ZAKLJUČAK

Cilj ovog rada bio je kreiranje sistema za predikciju hit pesama na *Billboard* top-listama koji bi bio od velikog značaja muzičkim kućama i izvođačima. U radu su kreirani skup podataka od 9126 i njegov podskup od 3390 pesama opisanih audio i tekstualnim obeležjima.

Implementacija je odrađena u *Python* jeziku pomoću *topic* i BERT modela i pet modela mašinskog učenja. Performanse svih obučanih modela su slične – tačnost za ceo

skup je u rangu 69-72%, a za podskup oko 74-75%. Dodatne optimizacije modela nisu dale bolje rezultate. Ideje za dalja proširenja rada obuhvataju odabir dodatnih obeležja koja tačnije opisuju podatke, kao i poboljšanje BERT modela implementacijom ponovnog treniranja pretreniranih modela uključujući i tekstualna obeležja iz skupa u ovom radu.

7. LITERATURA

- [1] Y. Ni, R. Santos-Rodriguez, M. Mcvicar and T. De Bie, "Hit song science once again a science", *4th International Workshop on Machine Learning and Music*, 2011.
- [2] J. Devlin, M.W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding", *arXiv preprint arXiv:1810.04805*, 2018.
- [3] <https://www.billboard.com/charts/hot-100>
- [4] <https://www.spotify.com/>
- [5] <https://genius.com/>
- [6] K. Middlebrook and K. Sheik, "Song Hit Prediction: Predicting Billboard Hits Using Spotify Data", *arXiv preprint arXiv:1908.08609*, 2019.
- [7] E. Georgieva, M. Suta and N. Burton, "HITPREDICT: PREDICTING HIT SONGS USING SPOTIFY DATA", 2018.
- [8] A. Singhi and D.G. Brown, "Can song lyrics predict hits", *Proceedings of the 11th International Symposium on Computer Music Multidisciplinary Research*, pp. 457-471, 2015.
- [9] R. Dhanaraj and B. Logan, "Automatic Prediction of Hit Songs", *ISMIR*, pp. 488-491, 2005.

Kratka biografija:



Sandra Rajanović rođena je u Beogradu 1995. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnika i računarstvo – Softversko inženjerstvo i informacione tehnologije odbranila je 2020. god.

RAZVOJ MODELA UPRAVLJANJA PMSM ZA PRIMENU U POGONU ELEKTRIČNIH VOZILA U dSPACE/MICROLABBOX OKRUŽENJU

DEVELOPMENT OF A DSPACE/MICROLABBOX BASED ELECTRIC DRIVE SYSTEM

Ivana Santrač, Boris Dumnić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu testirana je aplikacija za upravljanje sinhronom mašinom sa stalnim magnetima u pogonu električnih automobila. Korišćen je sistem hardvera u zatvorenoj petlji, pri čemu se mikrokontroler AURIX iz stvarnog pogona, proizvođača „Infineon Technologies AG“, zamenio dSPACE sistemom, odnosno MicroLabBox hardverskim i softverskim okruženjem.

Ključne reči: sinhrona mašina sa stalnim magnetima, dSPACE, MicroLabBox

Abstract – In this paper the application for controlling a permanent magnet synchronous motor in automotive drive system is tested. Hardware in the loop system is used, whereby the microcontroller AURIX from the real system, manufactured by „Infineon Technologies AG“, is replaced with dSPACE system, MicroLabBox hardware and software environment.

Keywords: permanent magnet synchronous motor, dSPACE, MicroLabBox

1. UVOD

Električni automobili bili su popularni još između kasnih 1800-tih i ranih 1900-ih godina. Oni su u ovom periodu davali više komforosti i jednostavnosti prilikom korišćenja, nego što je to bio slučaj sa automobilima koji su se pokretali fosilnim gorivima. Za njihovo pokretanje bilo je potrebno fizičko okretanje ručice. Međutim, pronalaskom elektromehaničkog pokretača SUS motora, kao i zbog razvoja ove vrste automobila, boljeg dometa i masovne proizvodnje, konvencionalni automobili su postali najzastupljeniji [1].

Ipak, povećavanjem broja automobila sa unutrašnjim sagorevanjem, dolazilo se sve više do njegovih negativnih strana. Prvo i najbitnije, kvalitet vazduha i nivo buke su se znatno pogoršali. Društvena svest o globalnom zagrevanju kao posledici ljudskog dejstva su se iz godine u godinu sve više povećavali. Osim toga, dolazi se i do činjenice da su izvori nafte ograničeni, tako da je ponovo počelo da se traga za alternativom SUS motorima. Upravo zbog toga električna vozila ponovo su dospela u centar pažnje [2].

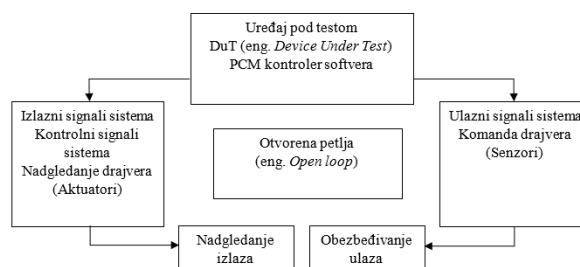
NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Boris Dumnić, van. prof.

2. OPIS HARDVERA I SOFTVERSKOG OKRUŽENJA

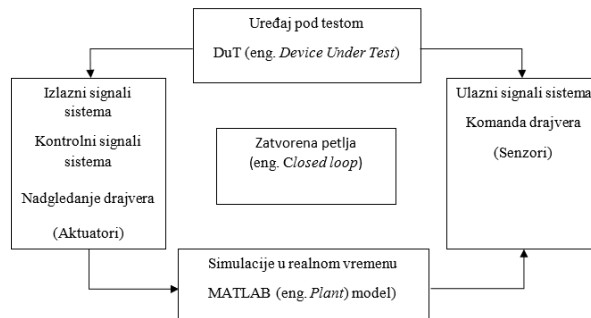
Simulacija sa hardverom u petlji (eng. *Hardware in the loop - HIL*) je tehnika koja se koristi u razvoju i testiranju kompleksnih ugrađenih sistema (eng. *Embedded systems*) u realnom vremenu. Ovaj tip simulacija predstavlja efikasnu platformu zbog toga što se u samo testiranje dodaje kompleksnost postrojenja koje se kontroliše. Sistem koji je ugrađen se ispituje u interakciji upravo sa ovom simulacijom. Prilikom simulacija sa hardverom u petlji koriste se matematički algoritmi koji oponašaju rad postrojenja u realnom vremenu i pri realnim uslovima, sa ciljem predviđanja njegovog ponašanja [3].

HIL aplikacije za testiranje u fazi dizajniranja mogu biti klasifikovane kao testiranje u otvorenoj petlji (eng. *Open Loop Testing*) i testiranje u zatvorenoj petlji (eng. *Closed Loop Testing*). Prilikom testiranja u otvorenoj petlji, koriste se jednostavniji ulazno/izlazni (I/O) model bez povratne sprege, kao što je to prikazano na slici 2. 1.



Slika 2.1 HIL aplikacija u otvorenoj petlji [4]

Sa druge strane, testiranje u zatvorenoj petlji zahteva kompletan model sa odgovarajućim izlaznim odzivima sistema, kao što je prikazano na slici 2.2. Prilikom ovakvog testiranja komandne, odnosno referentne vrednosti se koriste za pokretanje samog modela. Izaz modela jeste mehanička ili električna jedinica i ona je dalje korišćena za proračun ponašanja i dinamičkih odziva senzora [4].



Slika 2.2 HIL aplikacija u zatvorenoj petlji [4]

2.1. Opis dSPACE/MicroLabBox okruženja

DSPACE je sistem za simulacije u realnom vremenu baziran na MATLAB-u/Simulink-u i predstavlja softversku i hardversku platformu za razvoj i testiranje kontrolnih sistema. dSPACE sistem predstavlja platformu za aplikacije bazirane na HIL simulacijama u realnom vremenu, pri čemu su ispunjeni različiti zahtevi kontrole, kao što su podešavanje parametara, radno okruženje samog kontrolera u realnom vremenu, i drugi [5].

MicroLabBox je hardverski uređaj, koji je podržan od strane obimnog dSPACE softverskog paketa, uključujući *Real-Time Interface (RTI)* za Simulink, za modele bazirane na ulazno/izlaznoj integraciji i eksperimentalni softver *ControlDesk*, koji obezbeđuje pristup aplikacijama u realnom vremenu tokom trajanja simulacija putem grafičkih instrumenata [6].

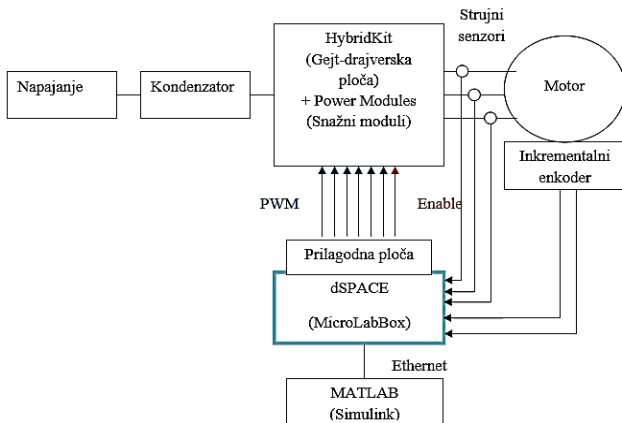
U ovom radu korišćena je *spring-cage* panelna varijanta prikazana na slici 2.3.



Slika 2.3 Izgled MicroLabBox uređaja [6]

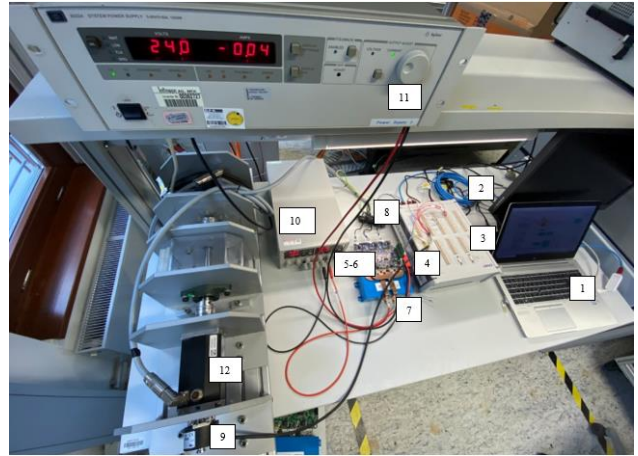
2.2. Opis razvojnog okruženja

Kontrolni sistem sinhronog motora sa stalnim magnetima (eng. *Permanent Magnet Synchronous Machine - PMSM*), baziran na eksperimentu u dSPACE-u, prikazan je na slici 2.4.



Slika 2.4 PMSM kontrolni sistem baziran na dSPACE razvojnom okruženju

Na slici 2.5 prikazana je postavka celokupnog sistema za testiranje u laboratoriji. Komponente koje su prikazane na slici 2.5 su sledeće: 1 - računar, 2 - kabel za uspostavljanje Ethernet veze, 3 - MicroLabBox, prilagoda ploča, 5 i 6 - HybridKit (gejt-drajverska ploča) i HybridPACK™ Drive moduli (snažni moduli), pri čemu su obe ploče proizvođača „Infineon Technologies AG“, 7 - „film kondenzator“ za HybridPACK™ IGBT module, 8 - strujni LEM senzori, oznake HC5F400-S, 9 - Inkrementalni enkoder „Baumer“, oznake GI341, 10 - izvor napajanja prilagodne ploče, 11 - izvor napajanja povezan sa kondenzatorom, 12 - PMSM motor proizvođača „Dunkermotoren“, oznake BG 75x50.



Slika 2.5 Postavka sistema za testiranje u laboratoriji

3. RAZVOJ MATEMATIČKOG MODELA

Prilikom izrade samog modela električne mašine, poželjno je uzeti u obzir sve osobine sistema koje mogu uticati na njegovo ponašanje. Sa druge strane, glavna prednost modelovanja jeste pojednostavljenje celokupnog modela i zanemarivanje elemenata koji nisu bitni za fizički proces [7].

3.1. Matematički model u trofaznom domenu

Jednačine naponske ravnoteže PMSM predstavljene su sledećim relacijama [7]:

$$\begin{bmatrix} u_{sa} \\ u_{sb} \\ u_{sc} \end{bmatrix} = \begin{bmatrix} R_a & 0 & 0 \\ 0 & R_b & 0 \\ 0 & 0 & R_c \end{bmatrix} \begin{bmatrix} i_{sa} \\ i_{sb} \\ i_{sc} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \psi_{sa} \\ \psi_{sb} \\ \psi_{sc} \end{bmatrix} \quad (3.1)$$

Naponi u_{sa} , u_{sb} i u_{sc} predstavljaju fazne napone na priključcima statora, i_{sa} , i_{sb} i i_{sc} struje kroz statorske namotaje, dok R_a , R_b i R_c predstavljaju otpornosti statorskog namotaja i za njih važi da imaju jednake vrednosti pod uslovom da je mašina simetrična.

Jednačine kojima se opisuju fluksni obuhvati statora [7]:

$$\begin{bmatrix} \psi_{sa} \\ \psi_{sb} \\ \psi_{sc} \end{bmatrix} = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \begin{bmatrix} i_{sa} \\ i_{sb} \\ i_{sc} \end{bmatrix} + \begin{bmatrix} \psi_{ma} \\ \psi_{mb} \\ \psi_{mc} \end{bmatrix} \quad (3.2)$$

Jednačina elektromagnetnog momenta [7]:

$$T_e = \frac{1}{2} p [i_s]^T \frac{d[L]}{dt} [i_s] = \frac{p}{2} [i_{sa} i_{sb} i_{sc}] \begin{bmatrix} \frac{L_{aa}}{dt} & \frac{L_{ab}}{dt} & \frac{L_{ac}}{dt} \\ \frac{L_{ba}}{dt} & \frac{L_{bb}}{dt} & \frac{L_{bc}}{dt} \\ \frac{L_{ca}}{dt} & \frac{L_{cb}}{dt} & \frac{L_{cc}}{dt} \end{bmatrix} \quad (3.3)$$

pri čemu je sa p označen broj pari polova. Na kraju se dolazi do Njutnove jednačine rotacionog kretanja [7]:

$$J \frac{d\omega_r}{dt} = T_e - T_m \quad (3.4)$$

pri čemu je sa J označen momenat inercije rotora, sa T ukupni momenat tereta, a sa $\omega_r = d\theta_r/dt$ ugaona brzina rotora.

3.2. Matematički model u dvofaznom stacionarnom, $\alpha\beta$ domenu

Prva transformacija koja se izvršava jeste Klarkina transformacija (eng. *Clarke*), odnosno transformacija raspredznanja. Prilikom ove transformacije dolazi se do prenosa trofaznih veličina iz stacionarnog abc domena u stacionarni dvofazni $\alpha\beta$ domen. Klarkina transformacija se može opisati sledećom relacijom [7]:

$$V_{\alpha\beta} = C \cdot V_{abc} \quad (3.5)$$

Jednačine naponske ravnoteže [7]:

$$\begin{bmatrix} u_{s\alpha} \\ u_{s\beta} \end{bmatrix} = \begin{bmatrix} R_s & 0 \\ 0 & R_s \end{bmatrix} \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \psi_{s\alpha} \\ \psi_{s\beta} \end{bmatrix} \quad (3.6)$$

Jednačine fluksnih obuhvata [7]:

$$\begin{bmatrix} \psi_{s\alpha} \\ \psi_{s\beta} \end{bmatrix} = \begin{bmatrix} L_{sr} & 0 \\ 0 & L_{sr} \end{bmatrix} \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} + \psi_{PM} \begin{bmatrix} \cos(\theta_{PM}) \\ \sin(\theta_{PM}) \end{bmatrix} \quad (3.7)$$

pri čemu je $L_{SR} = \frac{L_d + L_q}{2}$.

3.3. Matematički model u dvofaznom rotacionom, dq domenu

Nakon što je obaveljena Klarkina transformacija, na red dolazi transformacija rotacije, odnosno Parkova transformacija. Prilikom ove transformacije dvofazna veličina iz stacionarnog $\alpha\beta$ domena prenosi se na dvofazni sinhrono rotirajući dq sistem. Osnovni cilj ove transformacije jeste da se obezbedi nezavisnost matrice induktivnosti od ugona položaja rotora θ_R , odnosno vremena. Ova transformacija opisana je sledećom relacijom [7]:

$$V_{dq} = V_{\alpha\beta} e^{-j\theta_{dq}} \quad (3.8)$$

Primenom Parkove transformacije jednačine naponske ravnoteže dobijaju sledeći oblik [7]:

$$\begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} = \begin{bmatrix} R_s & 0 \\ 0 & R_s \end{bmatrix} \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \psi_{sd} \\ \psi_{sq} \end{bmatrix} + \omega_r \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \psi_{sd} \\ \psi_{sq} \end{bmatrix} \quad (3.9)$$

Jednačine fluksnih obuhvata [7]:

$$\psi_{sd} = L_s i_{sd} + \psi_{PM} \quad (3.10)$$

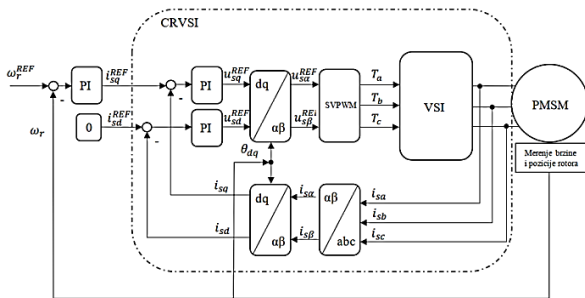
$$\psi_{sq} = L_s i_{sq} + \psi_{PM} \quad (3.11)$$

Jednačina za moment koji proizvodi PMSM [7]:

$$m = \frac{3}{2} P \psi_{PM} i_{sq} \quad (3.12)$$

3.4. Vektorsko upravljanje sinhronom mašinom

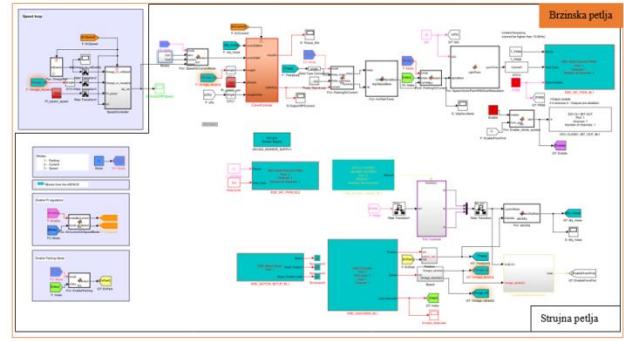
Vektorsko upravljanje (eng. *Field oriented control*) ima cilj nezavisnog upravljanja magnetnim poljem i momentom, kontrolišući d i q komponente struje. Osnova prednost ove tehnike jeste brzi odziv i mala talasnost momenta [8]. Na slici 3.1 prikazan je blok dijagram standardnog upravljanja pogonom sa sinhronim motorom sa stalnim magnetima.



Slika 3.1 Blok dijagram upravljanja PMSM [7]

4. MODEL U MATLAB-U (SIMULINK-U)

Na slici 4.1 prikazan je model vektorskog upravljanja PMSM za izvršavanje u realnom vremenu, realizovan u MATLAB/Simulink-u. Narandžastom bojom označena je brzinska petlja, dok je crnom bojom uokviren deo koji se odnosi samo na strujnu petlju.

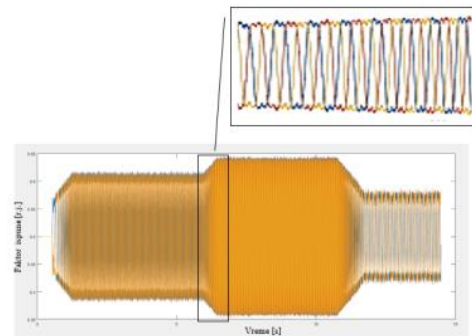


Slika 4.1 Model vektorskog upravljanja sa PMSM u MATLAB-u/Simulink-u

Implementirana su 3 režima rada sa PMSM i to: parking, strujni režim i brzinski režim.

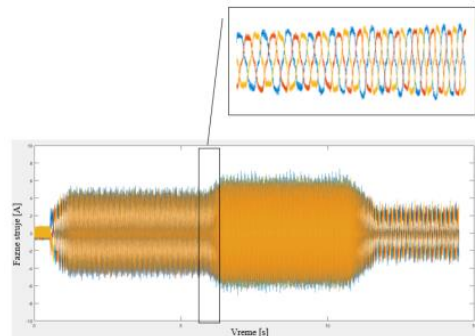
5. REZULTATI EKSPERIMENTA

U okviru ovog rada, na prethodno opisanoj postaci za testiranje PMSM u laboratoriji, izvršeno je niz eksperimenata. Upravljanje tokom eksperimenta, pokretanje kao i zaustavljanje eksperimenta izvršeno je softverskim alatom ControlDesk. U eksperimentu čiji će rezultati biti prikazani u nastavku izvršeno je pokretanje PMSM a nakon toga je izvršena promena mehaničke brzine obrtanja mašine sa 300 o/min na 500 o/min i zatim na 100 o/min. Na slici 5.1 prikazana je vremenska zavisnost faktora ispunje, gde se sa povećanjem brzine povećava i vrednost faktor ispunje.



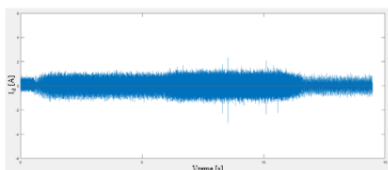
Slika 5.1 Vremenska zavisnost faktora ispunje

Na slici 5.2 prikazane su struje sve tri faze tokom ovog eksperimenta. Može se videti da su fazne struje PMSM pomerene za 120°. Takođe, kao što je i očekivano sa povećanjem brzine mašine povećava se i amplituda struje koju mašina vuče iz izvora napajanja.

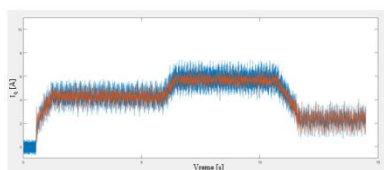


Slika 5.2 Vremenska zavisnost faznih struja

Na slici 5.3 prikazana je podužna komponenta struje statora (I_d), dok je na slici 5.4 prikazana poprečna komponenta (I_q). Može se videti da je podužna komponenta tokom čitavog eksperimenta ostala na nultoj vrednosti, dok se poprečna komponenta menjala u skladu sa promenom brzine.

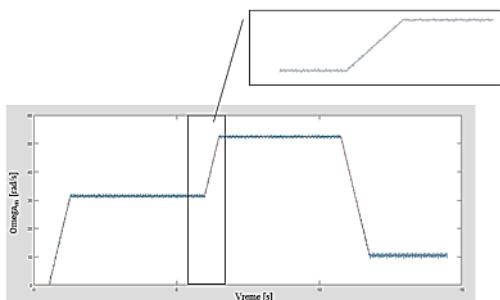


Slika 5.3 Vremenska zavisnost podužne (I_d) komponente struje statora

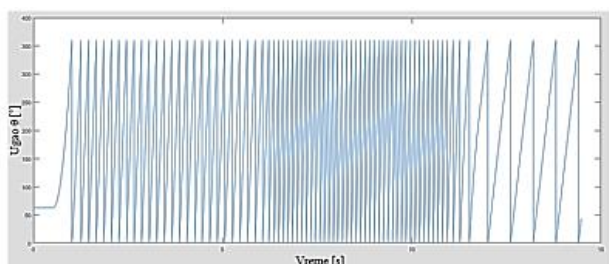


Slika 5.4 Vremenska zavisnost poprečne (I_q) komponente struje statora

Na slici 5.5 prikazana je mehanička brzina u o/min, dok je na poslednjoj slici prikazan ugao θ . Povećanjem brzine povećava se frekvencija, pa je samim time i učestalost ugla θ veća.



Slika 5.5 Vremenska zavisnost mehaničke brzine PMSM



Slika 5.6 Vremenska zavisnost ugla θ

6. ZAKLJUČAK

Razvoj i testiranje kontrolnih algoritama za upravljanje elektromotornim pogonima zahteva upotrebu savremenih softverskih rešenja i emulacionih kao i eksperimentalnih tehnika. Celokupan sistem hardvera u petlji, prikazan u ovom radu, praktično je implementiran i testiran u laboratoriji tima za istraživanje i razvoj, u automobilskom sektoru, kompanije „Infineon Technologies AG“. Rezultati dobijeni simulacijama odgovaraju osobinama koje su karakteristične za upravljanje sinhronim motorom sa stalnim magnetima i samim time potvrđuju tačnost

algoritma upravljanja koji je razvijen u ovom radu i testiran. Iako se u rezultatima vide šumovi u merenim veličinama struje i brzine obrtanja, ostvaren je sasvim zadovoljavajući kvalitet upravljanja PMSM dok su ovi šumovi direktna posledica korišćenja strujnih senzora za prilično veliki opseg struja od ± 400 A, dok su vrednosti merenih struja svega par A.

7. LITERATURA

- [1] „Global EV Outlook”, <https://www.iea.org/reports>
- [2] „Istorija hibridnih i električnih vozila od prvih automobila do danas”, <https://www.energetskiportal.rs>
- [3] “Hardware-in-the-Loop” <https://www.ni.com/en-us/>
- [4] S. Nabi i M. Balike, Visteon Corporation, J. Allen i K. Rzemien, dSPACE Inc, “An overview of hardware-in-the-Loop testing systems at Visteon”, 2014
- [5] Q. Yang i Z. Kang, “Study on space vector PWM technology of PMSM control system based on dSPACE”, *IEEE International Conference on Information and Automation (ICIA)*, str. 579
- [6] “dSPACE web page”, <https://www.dspace.com/>
- [7] M. Gecić, „Energetski efikasno digitalno upravljanje sinhronim motorom sa stalnim magnetima pri velikim brzinama obrtanja“, doktorska disertacija, Fakultet tehničkih nauka, Novi Sad, 2016.
- [8] D. Vindel, S. Haghbin, A. Rabiei, O. Carlson, R. Ghorbani, “Field oriented control of a PMSM drive system using the dSPACE controller”, *IEEE International Electric Vehicle Conference (IEVC)*

Kratka biografija:



Ivana Santrač rođena je u Vršcu 1995. godine. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – energetska elektronika i električne mašine odbranila je 2021. godine.
kontakt: santracivana@gmail.com



Boris Dumnić rođen je u Bileći 1976. god. Diplomirao je 2003. godine na Fakultetu tehničkih nauka u Novom Sadu – Elektrotehnika i računarstvo. Od 2004. godine zaposlen je na Fakultetu tehničkih nauka u Novom Sadu. Magistarski rad je odbranio 2007. godine, a 2013. godine je odbranio i doktorsku tezu. Oblast interesovanja su mu električne mašine, pogoni, energetska elektronika i obnovljivi izvori električne energije.

ISPITIVANJE RELEJNE ZAŠTITE SREDNJENAPONSKOG IZVODA MEDIUM VOLTAGE FEEDER RELAY PROTECTION TESTING

Dimitrije Mijić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je podešena i ispitana zaštita sredjenaponskog izvoda, odnosno prekostrujna, kratkospojna i usmerena zemljospojna zaštita. Nakon svakog od ispitivanja, generisan je deo izveštaja o ispitivanju zaštite. Opisano je kako radi kofer za ispitivanje relejne zaštite.

Ključne reči: releji, relejna zaštita, kofer za ispitivanje zaštite.

Abstract – The main focus of the paper is to set and test medium voltage feeder protection, time overcurrent, instantaneous overcurrent and directional ground time overcurrent protection. After each test of one of the protections, a part of the report that is generated, after the protection test, will be displayed. It will be described how relay test unit works.

Keywords: relays, power system protection, relay testing unit.

1. UVOD

U radu je objašnjen način funkcionisanja zaštite i opisan način funkcionisanja kofera za ispitivanje relejne zaštite. Nakon toga je izvršeno podešavanje i ispitivanje zaštite jednog sredjenaponskog izvoda (10kV) kog štiti mikroprocesorska zaštita pomoću prekostrujnog, kratkospojnog i usmereno – zemljospojnog člana. Nakon svakog ispitivanja jedne od zaštita, prikazan je deo izveštaja koji se generiše nakon ispitivanja zaštite.

U drugoj glavi su dati osnovni pojmovi relejne zaštite, kao i ostali pojmovi potrebni za razumevanje izložene materije.

Treća glava je posvećena samom podešavanju i ispitivanju zaštite, kao i rezultatima dobijenim tokom ispitivanja.

Četvrta glava je posvećena zaključku za zadati problem.

U petoj glavi je navedena korišćena literatura.

2. OSNOVE RELEJNE ZAŠTITE

U ovoj glavi će biti objašnjeni elektromehanički i mikroprocesorski zaštitni uređaji, kao i način funkcionisanja kofera za ispitivanje relejne zaštite.

2.1. Elektromehanički zaštitni uređaji

Tehnika relejne zaštitne električnih mreža i postrojenja razvijala se paralelno s opštim razvojem elektroenergetike, koje pred nju postavlja sve složenije zadatke [1]. Nagli razvoj niskonaponskih (NN) i visokonaponskih (VN) mreža, izgradnja velikih elektrana i transformatorskih stanica te razvoj industrije osetljive na prekide u snabdevanju električnom energijom, postavili su pred tehniku relejne zaštite zadatak da izradi uređaje koji će momentalno lokalizovati i selektivno isključiti svaki kvar. U naporima da se što više poboljša kvalitet električne energije i da se osigura neprekidnost u snabdevanju potrošača električne energije, relejna zaštita dobila je centralno mesto u kompleksu automatizacije elektroenergetskih postrojenja.

U vreme dok su električne mreže bile sasvim male i jednostavne osigurači su bili sasvim dovoljna zaštita i to samo na generatorima. S porastom opterećenja i razvojem mreža rasle su i struje kratkog spoja, tako da se je došlo do snaga koje su postale prevelike za osigurače (izrada ovakvih osigurača bi bila ekstremno skupa). Zbog toga se došlo do rešenja da okidač, elektromagnet, koji kod nedozvoljenog porasta struje privlači kvotu i na taj način deluje na isključenje prekidača. Kasnije je izrađen prvi relej, tj. elektromagnet koji je električno delovao na isključenje prekidača. Dalji razvoj kretao se u smeru izrade sve preciznijih, osetljivijih i sigurnijih releja.

Prvi kriterijum merodavan za prorađivanje releja bila je struja (prekostrujni releji). Kasnije su izrađeni releji koji su delovali u međusobnoj zavisnosti napona i struje. To su bili releji snage koji su, osim na veličinu struje reaguvali i na njen smer. Daljim razvojem električnih mreža visokog napona ni ovakvi releji nisu mogli zadovoljiti sve zahteve. Razvijeni su razni otpori ili distantni releji, priključeni na struje i napone. Vreme delovanja ovih releja zavisilo je o udaljenosti kvara od mesta ugradnje releja. Što je kvar bliži, relej je delovao brže. Razvijen je osim toga niz specijalnih releja za zaštitu svih elemenata elektroenergetskog sistema. Paralelno sa razvojem zaštite električnih mreža tekao je i razvoj relejne zaštite generatora, transformatora, motora i dr.

Osnovni element zaštitnih uređaja je relej. To je uređaj koji kod unapred određene nadgledane veličine izaziva naglu promenu u jednom ili više signalnih ili komandnih strujnih krugova. Nadgledana veličina može da bude električna struja, napon, snaga, otpor, frekvencija ili neelektrična temperatura, pritisak. Nagla promena u komandnim ili signalnim krugovima, kad nadgledana veličina premaši određenu vrednost, odražava se u zatvaranju ili otvaranju kontakata, putem kojih se daju potrebne komande ili se uključuju signalni releji. Kod

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je dr Duško Bekut, red. prof.

beskontaktnih releja transduktori, tranzistori, elektronske cevi efekat je isti kao i kod releja s kontaktima, tj. kad nadgledana veličina premaši podešenu vrednost dešava se bitna promena stanja u krugu na koji relej deluje.

Posebnu vrstu releja predstavljaju vremenski i pomoćni releji. Oni ne deluju na pojedine električne ili mehaničke veličine koje se pojavljuju kod kvarova, nego služe kao pomoćni elementi u relejnim uređajima. Vremenski releji imaju zadatak da uspore delovanje zaštitnog uređaja kad je to potrebno, zbog postizanja selektivnosti ili izbegavanja nepotrebnog delovanja kod kratkotrajnih porasta kontrolisane veličine. Pomoćni releji upotrebljavaju se za pojačanje slabih impulsa koje daju precizni releji koji nisu u stanju da direktno deluju na izvršenje potrebnih komandi, za istovremeno komandovanje sa više nezavisnih strujnih krugova i za druge pomoćne svrhe.

Osnovni zahtevi koji se postavljaju pred relejnu zaštitu su: selektivnost, brzina, visoka osetljivost i sigurnost u radu.

2.2. Mikroprocesorski zaštitni uređaji

Do sada je opisan razvoj zaštite i osnovni pojmovi i shvatanja kako bi se mogla razumeti tema ovog rada [2]. Opisivani su elektromehanički releji koji su i dalje prisutni na području Srbije. Njih prate osobine vrlo visoke pouzdanosti, robusnosti i jednostavne izrade. Njih, u poslednje vreme, počinju da menjaju uređaji mikroprocesorske zaštite. Ovo omogućuje realizaciju niza novih funkcija. Za razliku od klasičnih releja (elektromehaničkih i statičkih) mikroprocesorske digitalne zaštite pored jedne, osnovne zaštitne funkcije, imaju i druge zaštitne i kontrolne funkcije. Jednostavno rečeno, ovi uređaji predstavljaju računare i granice za rad sa tim uređajima uveliko prevazilaze mogućnosti elektromehaničkih releja.

Složeniji digitalni releji imaju priključke protokola za merenje i komunikaciju, omogućavajući releju da postane element u SCADA (supervisory control and data acquisition) sistemu – slika 1.



Slika 1. Prednja strana mikroprocesorskog releja

Relaj može imati široku lepezu postavki, osim onoga što se može uneti pomoću tastera na prednjoj ploči, a ta podešavanja se prenose u relej preko interfejsa sa računarom (lični računar), a isti taj interfejs računara može se koristiti za sakupljanje izveštaja o događajima iz releja.

2.2. Kofer za ispitivanje relejne zaštite

Nakon svakog prepodešavanja parametara zaštite, zaštitu je potrebno ponovo ispitati. Takođe, zaštita treba periodično da se ispituje svake dve godine. Tokom ispitivanja se na proveravaju samo strujna ili naponska podešenja, već i same veze (kola za isključenje) i ostali podaci koji bi mogli da ukažu na kvar u kolima i preduprede otkaze zaštite u slučaju kvarova.

Uređaji za ispitivanje zaštite se nazivaju koferima za ispitivanje zaštite. Oni su namenjeni i za primarna i za sekundarna ispitivanja. Koferi „injektuju” struje u kolo i ako se radi o primernim vezama, ti koferi moraju biti velike snage usled velikih izlaznih struja. U ovom radu će biti korišćen kofer za sekundarno ispitivanje, međutim uz pravilno rukovanje, on može ispitivati i primarne veze. U radu je korišćen kofer pod oznakom Sverker 780 – slika 2.



Slika 2. Izgled korišćenog kofera za ispitivanje zaštite pod oznakom Sverker 780

3. ISPITIVANJE ZAŠTITE IZVODA

Za zaštitu kablovskih i nadzemnih vodova 10 kV i 20 kV u transformatorskim stanicama (TS) 110/10 kV/kV, 110/20 kV/kV, 110/35/10 kV/kV/kV i 35/10 kV/kV primenjuje se prekostrujna zaštita i zemljospojna zaštita [3]. Pošto se u radu ispituje 10 kV vod, napajan iz TS 35/10 kV/kV, to će se upravo pomenute zaštite i ispitivati.

3.1. Prekostrujna zaštita

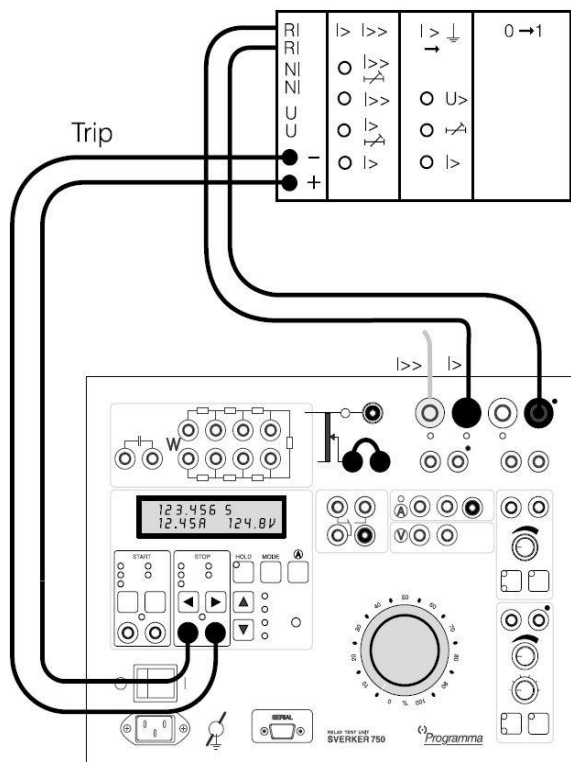
Prekostrujna zaštita je dvofazna ili trofazna maksimalna strujna vremenski nezavisna zaštita, koja reaguje:

- sa vremenskom zadržkom pri strujnim opterećenjima koja prelaze vrednosti dozvoljenih strujnih opterećenja voda, kao i pri udaljenim kratkim spojevima na vodu – I>;
- trenutno pri bliskim kratkim spojevima – I>>.

Potrebno je napomenuti da prekostrujna zaštita nije, po pravilu, zaštita od termičkog preopterećenja voda [3]. Zato se koristi "preventivna zaštita" od preopterećenja, koja se ostvaruje redovnim praćenjem i prognoziranjem opterećenja konzuma koji se napaja preko štićenog voda, analizom mogućeg opterećenja voda u normalnim i havarijskim uslovima s obzirom na odabranu koncepciju mreže i uslova okruženja.

Ispitivani izvod je podešen na 240 A i zategnut na 1 s. Vreme od 1 s je dovoljno jer se na izvodu ne nalazi nikakva oprema sa zaštitom (riklozeri) te nema potrebe za stepenovanjem. Ukoliko na nekom izvodu postoje riklozeri, onda se vreme prorade prekostrujne zaštite podešava na 1,5 – 2 s.

Nakon podešavanja zaštite, potrebno ju je ispitati. Na slici 3. je principijelno dat način vezivanja kofera kako bi se ispitala prekostrujna zaštita.



Slika 3. Način vezivanja kofera radi ispitivanja zaštite

Na slici se mogu uočiti dva strujna kruga. Prvi strujni krug predstavlja odavanje struje u relej i njega je potrebno povezati ili direktno na relej ili na izvučene strujne kontakte unutar ćelije. Drugi strujni krug predstavlja signal koji se vraća nazad u kofer i automatski ga gasi nakon što relej odradi. Na ovaj način je moguće meriti vreme prorade releja, pomoćnih releja i prekidača. Ovaj način ispitivanja je, takođe, dobar kako ne bi došlo do pregrevanja kofera i oštećenja žica i releja.

Nakon povezivanja, kofer se pali i meri se vrednost na kojoj relej prorađuje kao i proverava za koje vreme prorađuje. Za prekostrujnu zaštitu potrebno je ispitivati sve tri faze sa zemljom kao i međusobno.

3.2. Kratkospojna zaštita

Kratkospojna ($I >>$) je posebna vrsta prekostrujne zaštite ali namenjena za struje vrlo velikih intenziteta. Obično se vremenski podešava na 0s ukoliko ne postoji potreba za zatezanjem. Ukoliko na izvodu (npr. 35 kV) postoji više prekidača (i zaštite) u nizu postoji problem sa zatezanjem. Uzima se da zaštititi da registruje i prekidaču da isključi struju kvara treba gotovo 0,1 s što znači da se kratkospojna zaštita može stepenovati sa korakom od 0,1 s. Međutim, kako $I >>$ zaštita treba da isključi struje velikog intenziteta, to se te struje moraju isključiti što pre. Praksa je da kratkospojna zaštita može da traje najviše 0,2

s. Zbog ovoga postoji problem sa stepenovanjem, ukoliko postoji više prekidača u nizu.

Strujno podešenje kratkospojne zaštite $I_{>>}$ na izvodima 10 kV treba da bude izvedeno tako da se ova zaštita ne pobuđuje pri kratkim spojevima na sabirnicama 0,4 kV u TS 10/0,4 kV/kV [3]. Kako u ED Srbije snage trolnih kratkih spojeva na sabirnicama 0,4 kV ne prelaze 20 MVA ako TS 10/0,4 kV/kV radi u kablovskoj mreži 10 kV ili 15 MVA ako je TS priključena na nadzemni vod. Preporučuje se podešenje struje reagovanja kratkospojne zaštite I_{pks} na sledeće vrednosti (primarno):

- $I_{pk} = 1200$ A za kablovske vodove (podzemni kabl ili SKS);
- $I_{pk} = 900$ A za nadzemne vodove sa golim Al/Č užadima.

Pošto je ispitivani izvod realizovan kao podzemni kablovski, primarno strujno podešenje iznosi 1200 A, a vremensko zatezanje je 0s jer ne postoje potrebe za stepenovanjem na samom izvodu. Kako je ovaj tip zaštite podvrsta prekostrujne zaštite, gotovo sve što važi za prekostrujnu važi i ovde što se tiče vezivanja kofera i samog ispitivanja.

3.3. Zemljospojna zaštita

Kako bi bilo objašnjeno zašto se koristi upravo usmerena zemljospojna zaštita, moraju se objasniti načini uzemljenja transformatora SN/SN kV/kV odnosno distributivne mreže.

Istorijski gledano, metod uzemljenja sistema za različite postavke električnog sistema, npr. industrijska, komercijalna itd., potiče iz ranog perioda ovog veka, kada su samo dve metode razmatrane: čvrsto uzemljene i neuzemljene [4].

U nastavku su dati najčešće sretani tipovi uzemljenja neutralne tačke u Srbiji i njihova definicija.

- Izolovana (neuzemljena) mreža – sistem, kolo ili uređaj koji ne poseduje namerno povezanu metalnu tačku sa zemljom, osim indikacije koja se ostvaruje potencijalnim razdelnicima. Ipak, veza sa zemljom postoji, ali kroz kapacitivnost i provodnost rasklopne opreme i vodova.
- Uzemljena (direktno) mreža – sistem, kolo ili uređaj koji poseduje bar jednu tačku (neutralna tačka transformatora) koja je namerno kratko spojena sa zemljom.
- Uzemljena (indirektno) mreža – sistem, kolo ili uređaj koji poseduje bar jednu tačku (neutralna tačka transformatora) koja je spojena sa zemljom preko čisto rezistivne impedanse, čisto induktivne impedanse ili kombinacije. Otpornik služi za ograničavanje struja zemljospojeva na određene vrednosti dok induktivnost služi da kompenzuje kapacitivnu komponentu struje kvara kako bi struja kvara bila čisto aktivna.

Razlog za korišćenje izolovane mreže je da je nekim kritičnim potrošačima bilo potrebno omogućiti besprekidno napajanje, a izolovana (neuzemljena) mreža je upravo pružala ovu mogućnost. Izolovana mreža odlikuje veliku otpornost između zvezdišta i zemlje te zbog toga ograničava struju zemljospoja na vrednosti koje su vrlo niske. Ove vrednosti su toliko niske, da je dozvoljeno, po tehničkim preporukama raditi po nekoliko sati sa kvarom (zemljospojem) u mreži. Međutim, kod izolovanih mreža

se javlja jedna opasnost. Tokom jednog zemljospoja, fazni naponi nepogođenih faza dolaze na međufazne vrednosti napona, dok napon faze sa kvarom pada na nulu. Stoga, usled povišenog napona, može doći do preskoka između dve zdrave faze ili između zdravih faza i zemlje. Zbog ovoga je potrebno što pre moguće otkloniti zemljospoj u izolovanu mrežu.

Zaštita koja isključuje kvarove sa zemljom se naziva zemljospojna zaštita. Ona može biti usmerena ili neusmerena. Kada je usmerena, ona nadgleda, osim intenziteta struje kvara i napon i ugao između struje i napona, dok neusmerena zaštita nadgleda samo intenzitet struje.

Mreža u kojoj se ispituje izvod u ovom radu je izolovana (10 kV). Struje zemljospojeva su prvenstveno ograničene deonicama koje mogu biti vazdušne ili podzemne. U tehničkim preporukama propisanim od strane EPS Distribucije TP 4a, poduzna struja zemljospoja za vazdušni vod iznosi 0,03 A/km, a za podzemni 1,4 A/km [6]. Kako je poznato da su podzemni vodovi višestruko pouzdaniji od nadzemnih vodova (usled smanjenih atmosferskih uticaja), to se i teži zameniti sve vazdušne vodove podzemnim ili SKS (samonosivi kablovski snop), kao i sve nove vodove izgraditi kao podzemne ili SKS.

3.4. Ostale zaštite

U ovom poglavlju će biti spomenute ostale zaštite.

3.4.1. Zaštita od otkaza prekidača (ZOP)

ZOP koristi SMT od ZS (zaštite sabirnica) u transformatorskim ćelijama. U ispitivanom postrojenju, ZOP je podešen na 0,5 s. Ukoliko u TS postoje dva transformatora, svaki ima svoj ZOP koji reaguje posebno za svaku sekciju sabirnica. Ukoliko je spojno polje uključeno, prvo se isključuje prekidač spojnog polja, a nakon toga i prekidač u odgovarajućoj transformatorskoj ćeliji.

Ako dođe do kvara na izvodu koji pobuđuje neku od zaštita i ako je nalog za isključenje izvodnog prekidača poslat, međutim prekidač se ne isključuje iz nekog razloga, onda se pobuđuje signal otkaza prekidača i šalje se komanda prekidaču u transformatorskoj ćeliji da isključi prekidač.

3.4.2. Zaštita sabirnica (ZS) i blokada ZS

ZS se nalazi na zaštiti transformatorske ćelije. U suštini je to kratkospojna zaštita koja služi da isključi samo kvarove na sabirnicama (u ovom slučaju 10 kV). Ako je kvar na izvodu i ako zaštita na izvodu vidi kvar, ona treba da blokira ZS.

Treba još istaći da da bi se pobudile ove zaštite, potrebno je da kolica prekidača izvoda koji se ispituje budu u radnom položaju. Tokom test položaja se ove zaštite ne mogu pobuditi. Tokom ispitivanja izvoda TS koje se nalaze u pogonu, dešavalo se da reaguje ZOP ili ZS i izbaci celu TS iz pogona. Ovaj problem je rešen tako da se tokom ispitivanja zaštite izvoda TS koja je u pogonu, kolica izvodnog prekidača izvuku i automatski se prelazi u test položaj. U test položaju ZS i ZOP ne reaguju tako da se ne može izbaciti cela TS iz pogona.

4. ZAKLJUČAK

Za odabrani izvod su ispitane sve zaštite koje on poseduje i napravljen je izveštaj. Relejni zaštitu je potrebno redovno ispitivati kako bi se predupredili kvarovi i otkrili kvarovi koji su u normalnom pogonu prikriveni. Takođe, ispitivanjem zaštite se proveravaju i signali i upozorenja koja dobija dispečer, ukoliko je TS u daljinskom pogonu i vrši se provera lokalne signalizacije.

Preporuka je da se vrši ispitivanje zaštite nakon svakog prepodešavanja ili periodično na dve godine. Ovim je ispitivanje završeno.

5. LITERATURA

- [1] Numerical relay, Dostupno na: https://en.wikipedia.org/wiki/Numerical_relay (Pristupljeno: 2020).
- [2] Relejna zaštita, Dostupno na: <https://elraj.rs/relejna.html> (Pristupljeno: 2020).
- [3] EPS Distribucija – Tehnička preporuka TP 4a1: Zaštita elektrodistributivnih vodova 10 kV, 20 kV i 35 kV, Dostupno na: https://www.epsdistribucija.rs/interni_standardi/preporuke/TP%204a1.pdf (Pristupljeno: 2020).
- [4] David Shipp, Frank Angelini – Characteristics of different power systems neutral grounding techniques: Fact and fiction, Dostupno na: <https://www.eaton.com/content/dam/eaton/products/medium-voltage-power-distribution-control-systems/vacclad-w-pages/characteristics-power%20systems-neutral-grounding-techniques-fact-fiction-ap083005en.pdf> (Pristupljeno: 2020).

Kratka biografija:



Dimitrije Mijić rođen je u Loznici 1996. god. Osnovne akademske studije završio je na Fakultetu tehničkih nauka 2019. god. iz oblasti Elektrotehnike i računarstva, smer Elektroenergetski sistemi. Master rad, na istom fakultetu, na smeru Elektroenergetski sistemi odbranio je 2021. god. kontakt: mail.dmijic1996@gmail.com

PRIMENA IMPEDANSNE SPEKTROSKOPIJE ZA ANALIZU UTICAJA ETARSKIH ULJA NA PROVODNE STRUKTURE FABRIKOVANE NA PAPIRNOJ OSNOVI**IMPEDANCE SPECTROSCOPY APPLICATION FOR ANALYZING INFLUENCE OF ESSENTIAL OILS ON CONDUCTIVE MATERIALS FABRICATED ON PAPER**Stefan Švenderman, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu opisana je fabrikacija mikrofluidnih čipova koji sadrže provodnu strukturu od različitih provodnih materijala fabrikovanih na osnovi od papira. Upotrebljeni provodni materijali su zlato, srebro, aluminijum i provodna električna boja. U čipove su injektovane različite koncentracije rastvora etarskih ulja eukaliptusa i peperminta kako bi se ispitalo njihov abrazivni uticaj na različite provodne materijale. Karakterizacija uzoraka i dokumentovanje promena električnih veličina obavljena je primenom impedansne spektroskopije u različitim, ali ekvidistantnim vremenskim intervalima. Rezultati karakterizacije prikazani su u grafičkoj formi.

Ključne reči: mikrofluidika, etarska ulja, provodni materijali, impedansna spektroskopija

Abstract – In this work the fabrication of microfluid chips that contains electrode built from different conductive materials on paper surface was described. Used conductive materials are gold, silver, aluminium, conductive electric paint. Different mixture of eucalyptus and peppermint essential oils were injected in chips to provide results of degraded impact on conductive materials fabricated on the paper substrate. Chips characterization and examination of variation in electrical parameters were measured by impedance spectroscopy technique. Measurements were done in equidistant time steps. Characterization results were documented in numerous pictures and graphs.

Keywords: microfluidics, essential oils, conductive materials, impedance spectroscopy

1. UVOD

Osnovni zadatak ovog rada predstavlja analiza i dokumentovanje promena električnih parametara mikrofluidnog čipa koji kao jedan od svojih središnjih slojeva sadrži provodnu strukturu na osnovi od papira. Provodna struktura izložena je uticajima etarskih ulja različitih koncentracija. Pažljivom selekcijom odabrana su četiri provodna materijala različitih mehaničkih, električnih i hemijskih karakteristika.

Injekcijom rastvora etarskog ulja u kanal čipa stvara se abrazivna sredina za jedan od pomenutih provodnih materijala. Krajnji cilj i suština rada nalaze se u tome da se što tačnije okarakterise uticaj etarskih ulja na provodnu strukturu kako bi se verno prikazala promena mehaničkih, električnih i hemijskih karakteristika svakog provodnog materijala.

NAPOMENA:

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

2. MIKROFLUIDIKA

Mikrofluidika kao vrlo mlada nauka može se uvrstiti u posebnu granu elektrotehničke oblasti, konkretno širok spektar primene našla je u posebnoj primeni organske mikroelektronike.

Pogodnosti u primeni ogledaju se u veoma brzim analizama kojima su potrebne male količine fluida dok se ujedno ne narušava procenat pouzdanosti analiza i portabilnost uzoraka.

Posebna oblast mikrofluidike je mikrofluidika bazirana na papiru. Mikrofluidni uređaji koji su bazirani na papiru su u poslednjih nekoliko godina doživeli svoju široku rasprostranjenost. Uređaji se konstantno unapređuju i dalje usavršavaju kako bi obavljali sve sofisticiranije i veoma kompleksnije funkcije, a da se pritom zadrži preciznost i konzistentnost [1].

Glavni razlozi zašto je papir našao veoma široku primenu u mikrofluidici su višestruki. Prvi razlog je taj da je papir opšte sveprisutan celulozni materijal koji je veoma lako izraditi u odnosu na neke druge materijale. Drugi razlog ogleda se u njegovoj kompatibilnosti u mnogim hemijskim, biohemijskim i medicinskim aplikacijama. Treći razlog predstavlja kontrolisani prenos fluida uz pomoć kapilarnih sila bez potrebnog konstantnog prisustva strane spoljašnje sile.

U ovom trenutku razvoja, mikrofluidni čipovi bazirani na papiru smatraju se za komponente niske cene, lake i jednostavne upotrebe, a da pritom i dalje zauzimaju popriličnu relevantnost u unapređenju zdravstvenog sistema i brzom detekciji bolesti [2].

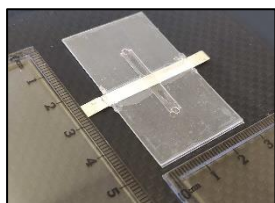
3. DIZAJN MIKROFLUIDNOG ČIPA

Postoje mnogobrojne tehnike za izradu mikrofluidnih komponenti i pogodan odabir tehnike vrši se na osnovu materijala od kojeg je potrebno fabrikovati komponentu.

Za potrebe izrade slojeva mikrofluidnog čipa u ovom radu upotrebljene su plastične folije debljine 120 µm i 80 µm.

Mikrofluidna komponenta sastoji se od ukupno 4 sloja plastične folije debljine 80 µm, 2 sloja plastične folije debljine 120 µm, 2 zaštitna sloja od plastične folije debljine 80 µm i centralnog sloja od papira debljine 125 µm na kojem je fabrikovan sloj od različitih provodnih materijala.

Na slici 1 prikazan je konačan izgled čipa.



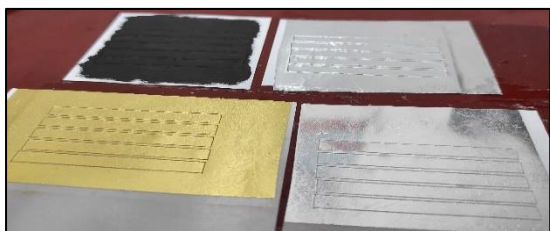
Slika 1. Konačan izgled čipa

4. SELEKCIJA PROVODNIH MATERIJALA

Odabir provodnih materijala izvršen je na osnovu dva kriterijuma. Prvi, provodni materijali koji se uzimaju u obzir moraju imati primenu u mikroelektronici. Drugi, provodni materijali moraju biti jednostavni za fabrikciju na papirnoj osnovi. U obzir su uzeti zlato, srebro, aluminijum i električna provodna boja.

Zlato i srebro koje je upotrebljeno u ovom radu nalazi se u obliku zlatnih listova. Listovi su sačinjeni od 99% čistog, 24-karatnog zlata i srebra. Debljina zlata je 10 μm , a srebra 32 μm . Aluminijum upotrebljen u ovom radu nalazi se u traci. Debljina sloja aluminijuma je 70 μm . Boja je rastvorljiva u vodi i veoma se lako nanosi četkicom ili sunderom na raznovrsne podloge. Namena boje je za izradu nestandardnih električnih kola i eksperimenata.

Provodni slojevi fabrikovani su na papiru za organsku i štampanu elektroniku. Debljina papira iznosi 125 μm . Konačan izgled provodnih slojeva prikazan je na slici 2.



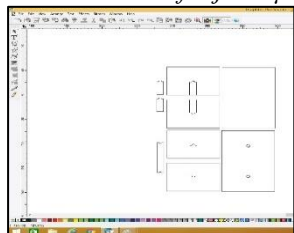
Slika 2 Konačan izgled provodnih slojeva

5. FABRIKACIJA SLOJEVA ČIPA

Cutter plotter je uređaj koji omogućava rezanje i graviranje željenih struktura od različitih materijala. Za pripremu fabrikcije slojeva na cutter plotter-u potrebno je imati spremnu kapton podlogu za cutter plotter, dvostrano lepljivu traku, valjak i plastičnu foliju debljine 120 μm i 80 μm . Montaža plastične folije debljine 80 μm prikazana je na slici 3. Nakon montaže folije potrebno je pokrenuti softver *GraphtecPro Studio* i učitati fajl sa ekstenzijom *.dxf* prethodno nacrtan u *AutoCad* alatu koji sadrži nacрте slojeva kao na slici 4.



Slika 3. Montaža folije 80 μm



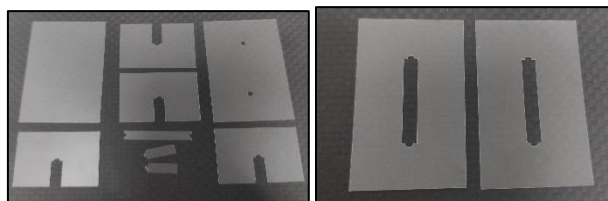
Slika 4. Učitani fajl u softveru

Nakon što je uzorak spreman potrebno ga je postaviti na radnu površinu cutter plotter-a i pritisnuti taster broj tri (Sheet) koji ima cilj da cutter plotter obavi sopstvenu kalibraciju. Postavka materijala prikazana je na slici 5.



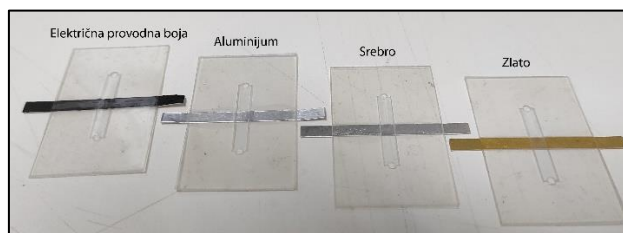
Slika 5. Postavka materijala na radnu površinu cutter-a

Poslednji korak predstavlja podešavanje brzine sečenja i vrednosti sile koju aktuator primenjuje na nož cutter plotter-a. Vrednosti koje se unose u ove parametre utiču na kvalitet izrade slojeva. Na slici 6a su prikazani slojevi debljine 80 μm , a na slici 6b slojevi debljine 120 μm .



Slika 6a. Slojevi od 80 μm Slika 6b. Slojevi od 120 μm

Poslednji korak fabrikcije čipa predstavlja laminacija izrađenih slojeva. Temperatura pri kojoj se vrši laminacija podešena je na 160°C. Konačan izgled fabrikovanih čipova sa provodnim materijalima prikazan je na slici 7.



Slika 7. Konačan izgled fabrikovanih čipova

6. ETARSKA ULJA

Etarska ulja imaju veoma široku primenu u različitim oblastima prehrambene, modne, tekstilne industrije. Primer primene predstavlja ulje sa aromom eukaliptusa koji je sastavni deo sredstva za čišćenje, različitih rastvarača, antiseptika, dezodoransa, sirupa za kašalj, pasta za zube.

Ulje sa aromom eukaliptusa u medicini ima primenu kao alternativa antibiotiku u lečenju određenih specifičnih bolesti, destruktivno utiče na bakterije i neke mikroorganizme u adekvatnoj koncentraciji [3].

U ovom radu upotrebljene su dve arome etarskog ulja, ulje sa aromom eukaliptusa i aromom pepermint. Oba ulja su brend proizvođača *Shadhi*. Pomenuta etarska ulja prepoznata su kao pogodna za upotrebu u ovom radu iz razloga što imaju veoma široku primenu u prehrambenoj, farmaceutskoj i medicinskoj industriji.

Zbog posedovanja karakteristike abrazivnog dejstva, ulja su rastvorena u tri različite koncentracije. Rastvarač je destilovana voda. Pomenuta tri rastvora predstavljaju različite odnose etarskog ulja i vode u razmeri 1:10, 1:50 i

1:100. Na slici 8 nalazi se izgled neophodne opreme za fabrikaciju rastvora različitih koncentracija.



Slika 8. Neophodna oprema za fabrikaciju rastvora

7. PRIMENA IMPEDANSNE SPEKTROSKOPIJE

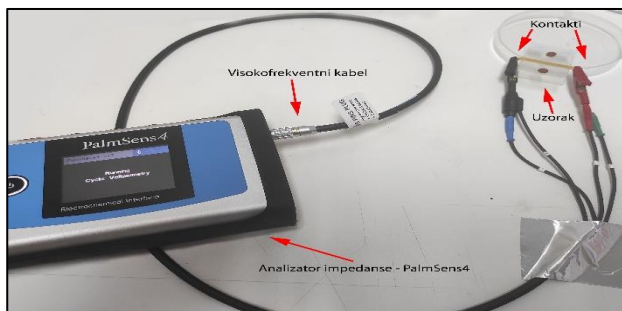
Za potrebe karakterizacije čipova korišćene su dve merne metode, elektrohemijska impedansna spektroskopija (EIS) i ciklična voltametrija (CV).

Ukoliko se na apscisu grafika postave vrednosti pobudnog naponskog signala, a na ordinati vrednosti odziva sinusnog strujnog signala, dobija se grafik pod nazivom CV grafik [4].

Ako se na grafiku na apscisi postave vrednosti realnog dela, a na ordinati vrednosti imaginarnog dela dobija se Nikvistov dijagram.

Jedini nedostatak kod Nikvistovog dijagrama predstavlja to da kada se pogleda tačka krive, odnosno njena vrednost, ne postoji mogućnost da se ustanovi vrednost frekvencije na kojoj je ona zabeležena.

Ovaj nedostatak otklonjen je upotrebom dijagrama koji prikazuje vrednosti realnog dela impedanse u funkciji promene frekvencije. Merenja su obavljena upotrebom spektralnog analizatora PalmSens4. Postavka eksperimenta prikazana je na slici 9.

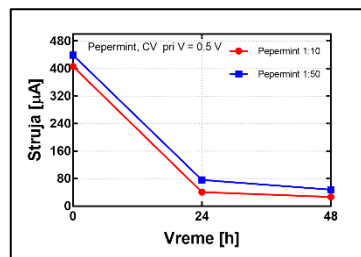


Slika 9. Postavka eksperimenta

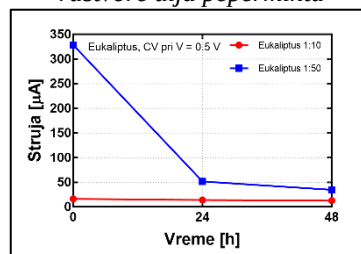
8. DISKUSIJA DOBIJENIH REZULTATA

Najveće promene vrednosti CV dijagrama mogu se zapaziti na uzorcima koji u sebi sadrže električnu provodnu boju. Kod 3 od 6 CV dijagrama provodne boje primetan je histerezis krive. Pojava histerezisa uzrokovana je rastvorima etarskih ulja. Za veoma kratko vreme, tokom merenja, dovoljno utiču na promenu impedanse provodnog materijala, električne provodne boje.

Očekivano je da se sa svakom novom injekcijom rastvora poveća otpornost provodnog materijala. Povećanje otpornosti prati se smanjivanjem vrednosti struje detektovane CV metodom. Na slici 10a i 10b prikazane su vrednosti struje CV dijagrama obe vrste ulja, za tačke čiji je napon u vrednosti od 0.5 V za sva merenja.



Slika 10a. Vrednosti struja uzoraka provodne boje za rastvore ulja peperminta



Slika 10b. Vrednosti struja uzoraka provodne boje za rastvore ulja eukaliptusa

Promene otpornosti provodnih materijala, realnog dela impedanse, prikazani su u naredne tri tabele. Sve vrednosti otpornosti u tabeli odgovaraju frekvenciji vrednosti 10 kHz. Notacija $Z'_{poč}$ označava vrednost otpornosti za prvo merenje, a Z'_{kraj} označava vrednost otpornosti poslednjeg merenja (48 časova od prvog merenja). U tabeli 1 nalazi se prikaz vrednosti otpornosti za koncentracije 1:10, u tabeli 2 koncentracije 1:50 i tabeli 3 koncentracije 1:100.

Tabela 1 Vrednosti otpornosti provodnih materijala za koncentracije rastvora 1:10

	Eukaliptus 1:10 [Ω]	Pepermint 1:10 [Ω]
Aluminijum	$Z'_{poč} = 0.069$ $Z'_{kraj} = 0.061$	$Z'_{poč} = 0.077$ $Z'_{kraj} = 0.067$
Boja	$Z'_{poč} = 8033$ $Z'_{kraj} = 18006$	$Z'_{poč} = 954$ $Z'_{kraj} = 12496$
Srebro	$Z'_{poč} = 0.59$ $Z'_{kraj} = 0.75$	$Z'_{poč} = 0.71$ $Z'_{kraj} = 0.94$
Zlato	$Z'_{poč} = 2.32$ $Z'_{kraj} = 2.17$	$Z'_{poč} = 2.04$ $Z'_{kraj} = 2.02$

Tabela 2 Vrednosti otpornosti provodnih materijala za koncentracije rastvora 1:50

	Eukaliptus 1:50 [Ω]	Pepermint 1:50 [Ω]
Aluminijum	$Z'_{poč} = 0.077$ $Z'_{kraj} = 0.069$	$Z'_{poč} = 0.066$ $Z'_{kraj} = 0.067$
Boja	$Z'_{poč} = 1411$ $Z'_{kraj} = 11081$	$Z'_{poč} = 1255$ $Z'_{kraj} = 9146$
Srebro	$Z'_{poč} = 0.59$ $Z'_{kraj} = 1.099$	$Z'_{poč} = 0.64$ $Z'_{kraj} = 0.91$
Zlato	$Z'_{poč} = 2.1$ $Z'_{kraj} = 2.1$	$Z'_{poč} = 1.78$ $Z'_{kraj} = 1.97$

Tabela 3 Vrednosti otpornosti provodnih materijala za koncentracije rastvora 1:100

	Eukaliptus 1:100 [Ω]	Pepermint 1:100 [Ω]
Aluminijum	$Z'_{poč} = 0.083$ $Z'_{kraj} = 0.070$	$Z'_{poč} = 0.068$ $Z'_{kraj} = 0.066$
Boja	$Z'_{poč} = 15224$ $Z'_{kraj} = 37897$	$Z'_{poč} = 6161$ $Z'_{kraj} = 12877$
Srebro	$Z'_{poč} = 0.54$ $Z'_{kraj} = 1.18$	$Z'_{poč} = 0.53$ $Z'_{kraj} = 0.74$
Zlato	$Z'_{poč} = 2.2$ $Z'_{kraj} = 2.37$	$Z'_{poč} = 1.88$ $Z'_{kraj} = 2.29$

Izučavanjem vrednosti otpornosti iz gore navedenih tabela na prvi pogled može se doneti nekoliko zaključaka. Najveće oscilacije u promeni vrednosti otpornosti prisutne su kod uzoraka koji sadrže provodu električnu boju. Rezultat pomenutih promena primarna je posledica nehomogenosti debljine sloja boje između uzoraka. Promene otpornosti uzoraka koji sadrže sloj sa električnom provodnom bojom u opsegu od 2 do 12 puta je i očekivan.

Uzorci sa provodnim slojem od aluminijuma su se pokazali kao najnepouzdaniji. Očekivano je da se sa svakom injekcijom rastvora u čip vrednost otpornosti povećava, a ne da se smanjuje. Kod uzoraka sa provodnim slojem od aluminijuma prisutno je najveće bočno razlivanje fluida van kanala unutar samog čipa. Ovim razlivanjem nastaju neželjeni parazitni elementi koji utiču u velikoj meri na pouzdanost merenja. Vrlo je verovatno da rastvor svojim postojanjem u čipu kreira parazitnu impedansu paralelnu sa horizontalnom osom provodnog materijala. Vrednost impedanse provodnog sloja i vrednost parazitne impedanse u paraleli formiraju ekvivalentnu impedansu manje vrednosti pa iz tog razloga je krajnja otpornost manja od početne.

Uzorci sa provodnim slojem od zlata znatno su pouzdaniji. Uticaj parazita kod uzoraka sa zlatom je primetan i unosi grešku u određena merenja. Sa druge strane neophodno je pomenuti kreirano parazitno serijsko LC kolo od kontakta uređaja i samog provodnog materijala čija impedansa teži minimalnoj vrednosti u blizini rezonantne učestanosti. Postoji mogućnost da frekvencija u vrednosti od 10 kHz predstavlja okolinu rezonantne učestanosti određenih parazitnih struktura.

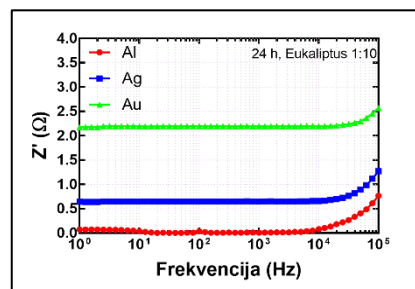
Kao najpouzdaniji materijal u ovom eksperimentu pokazali su se uzorci koji u sebi sadrže provodan sloj od srebra. Kod ovih uzoraka promene vrednosti otpornosti su u potpunosti ispunila očekivanja. Vrednosti krajnje otpornosti u odnosu na početnu veće su za najmanje 30 %. Jedino što dobijene vrednosti otpornosti ne mogu sa sigurnošću da potvrde jeste koje ulje je abrazivnije. Zbog prethodno pomenutih odstupanja u merenjima uzorci su naknadno podvrgnuti skenirajućoj elektronskoj mikroskopiji (SEM) kako bi se utvrdila tačna debljina provodnih slojeva.

Vrednost realnog dela impedanse u teoriji izračunava se na osnovu formule (1) koja zavisi od vrednosti specifične električne otpornosti, dužine i vrednosti površine poprečnog preseka provodnog materijala.

$$R = \frac{\rho L}{A} \quad (1)$$

Najveću vrednost specifične električne otpornosti ima aluminijum, potom zlato i najmanju srebro. Na osnovu različitih vrednosti ovih parametara očekivalo bi se da srebro ima najmanju otpornost. Međutim, ključnu vrednost otpornosti u izrazu (1) određuje vrednost površine poprečnog preseka provodnog materijala.

Kada se u izrazu simboli zamene vrednostima dolazi se do zaključka da najmanju otpornost poseduje aluminijum, potom srebro i na kraju zlato. Rezultati dobijeni izrazom (1) potvrđeni su na slici 11, prikazom realnog dela otpornosti u funkciji promene frekvencije za koncentraciju rastvora eukaliptusa 1:10.



Slika 11. Potvrda rezultata očekivanih izrazom (1)

9. ZAKLJUČAK

Sveobuhvatnim gledanjem na rezultate realizovanog eksperimenta sa sigurnošću se može potvrditi da je eksperiment uspešno realizovan. Generalno, cilj ovog rada je bio da se realizuju i ispituju uticaji etarskih ulja na provodne strukture primenom impedansne spektroskopije i pruženi su adekvatni rezultati koji su materijali najpogodniji i najnepogodniji za ovu primenu.

Ono što je nesvesno urađeno ovim radom je čak i najvažnije, a to je da su eksperiment i njegovi rezultati obogatili mikrofluidiku kao mladu nauku.

10. LITERATURA

- [1] Carlos Castro, Cindy Rosillo, Hideaki Tsutsui - Characterizing effects of humidity and channel size on imbibition in paper-based microfluidic channels
- [2] Xu Li, David R. Ballerini, Wei Shen - A perspective on paper-based microfluidics: Current status and future trends
- [3] Asaad Rehman Saeed Al-Hilphy - A practical study of new design of essential oils extraction apparatus using ohmic heating
- [4] David Loveday, Pete Peterson, and Bob Rodgers - Evaluation of Organic Coatings with Electrochemical Impedance Spectroscopy (Part 2)

Kratka biografija:



Stefan Švenderman rođen je u Zrenjaninu 1996. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnika i računarstva-Elektronika, odbranio je 2019. god. na temu "Karakterizacija biorazgradive fleksibilne podloge i njene senzorske primene."

REKONFIGURACIJA DISTRIBUTIVNIH MREŽA PRIMENOM MATEMATIČKIH I HEURISTIČKIH OPTIMIZACIONIH METODA

RECONFIGURATION OF DISTRIBUTION NETWORKS USING MATHEMATICAL AND HEURISTIC OPTIMIZATION METHODS

Dario Kanjo, Neven Kovački, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Cilj ovog rada jeste da prikaže i upoređi tri različita pristupa za rešavanje problema rekonfiguracije distributivnih mreža, a koji su zasnovani na matematičkim i heurističkim optimizacionim metodama. Najpre je prikazan matematički optimizacioni model za rekonfiguraciju distributivnih mreža zasnovan na mešovito celobrojnom programiranju. Zatim su prikazane metoda izmene grana i metoda simuliranog kaljenja, koje spadaju redom u heurističke i meta-heurističke optimizacione metode za rekonfiguraciju distributivnih mreža. Sve prikazane metode su testirane na primeru realne distributivne mreže, a rezultati su prikazani i komentarisani u radu.

Ključne reči: rekonfiguracija distributivne mreže, smanjenje gubitaka aktivne snage, mešovito celobrojno programiranje

Abstract – The aim of this paper is to present and compare three approaches to solve the problem of distribution network reconfiguration, which are based on mathematical and heuristic optimization methods. First, this paper presents mathematical optimization model for the distribution network reconfiguration which is based on the mixed integer programming. Then, the paper presents branch exchange method and the method based on simulated annealing, which represent heuristic and meta-heuristic methods used for the distribution network reconfiguration respectively. All presented methods are verified on the example of real distribution network and the results are presented and discussed in the paper.

Keywords: distribution network reconfiguration, reduction of active power losses, mixed integer programming, branch exchange, simulated annealing

1. UVOD

Rekonfiguracija distributivnih mreža (DM) predstavlja proces izmene njihove topološke strukture (konfiguracije), koji se postiže zatvaranjem/otvaranjem prekidačke opreme u mreži. Rekonfiguracija ima veliku primenu u upravljanju DM, gde se najčešće koristi za: smanjenje gubitaka aktivne snage [1], ravnomernu raspodelu opterećenja na napojnim transformatorima i izvodima, smanjenje padova napona (kvalitetnije naponske prilike), povećanje pouzdanosti napajanja, kao i restauraciju napajanja nakon kvarova u mreži [2].

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio doc. dr Neven Kovački.

U praksi se koristi veliki broj metoda za rekonfiguraciju distributivnih mreža, koje su zasnovane na matematičkim, heurističkim i meta-heurističkim optimizacionim metodama. Cilj ovog rada jeste da prikaže po jednu iz navedenih grupa metoda i da se one uporede na primeru realne distributivne mreže.

U svim prikazanim metodama kao funkcija cilja postavljena je minimizacija gubitaka aktivne snage, dok su kao ograničenja razmatrana pogonska ograničenja i radijalna konfiguracija distributivne mreže. Takođe, u svim prikazanim metodama optimalna konfiguracija se određuje za jedan fiksni nivo opterećenja distributivne mreže, na primer za vršno opterećenje. Dakle, pristup određivanju optimalne rekonfiguracije DM u ovom radu je statički.

2. MATEMATIČKI OPTIMIZACIONI MODEL ZA REKONFIGURACIJU DISTRIBUTIVNE MREŽE

U ovoj glavi je prikazan optimizacioni model za rekonfiguraciju DM koji je zasnovan na mešovito-celobrojnom programiranju. U prikazanom optimizacionom modelu statusi grana (l) (otvorena /zatvorena) predstavljeni su binarnim (0/1) promenljivama x_l , dok je režim DM predstavljen modulima i faznim stavovima (V_i, θ_i) fazora napona u čvorovima (i) DM.

Kao funkcija cilja postavljena je minimizacija gubitaka aktivne snage P_{Loss} , što se zapisuje na sledeći način:

$$\min_{x_l, V_i, \theta_i} \left(P_{Loss} = \sum_{l=1}^{N_L} P_{Loss, l} = \sum_{l=1}^{N_L} (P_{ij} + P_{ji}) \right), \quad (1)$$

gde su sa P_{Loss} označeni ukupni gubici aktivne snage, sa $P_{Loss, l}$ gubici aktivne snage u grani l , N_L predstavlja ukupan broj grana u DM, dok $P_{ij}(P_{ji})$ predstavlja tok aktivne snage u grani l od čvora i (j) ka čvoru j (i).

U datom optimizacionom modelu važe sledeća ograničenja:

1. Zavisnost tokova snaga (P_{ij}, Q_{ij}) u granama DM $l=1, 2, \dots, N_L$ od promenljivih odlučivanja (x_l, V_i, θ_i):

$$P_{ij} = x_l \{ G_l (V_i)^2 - V_i V_j [G_l \cos(\theta_i - \theta_j) + B_l \sin(\theta_i - \theta_j)] \} \quad (2)$$

$$Q_{ij} = x_l \{ -B_l (V_i)^2 - V_i V_j [B_l \cos(\theta_i - \theta_j) - G_l \sin(\theta_i - \theta_j)] \} \quad (3)$$

gde su G_l i B_l konduktansa i susceptansa grane l , respektivno.

2. Bilansi aktivnih i reaktivnih snaga u čvorovima distributivne mreže $i=1, 2, \dots, N_N$:

$$P_{G,i} - P_{L,i} - \sum_{j \in \Omega_i} P_{ij} = 0, \quad (4)$$

$$Q_{G,i} - Q_{L,i} - \sum_{j \in \Omega_i} Q_{ij} = 0. \quad (5)$$

gde su sa $P_{G,i}$, $P_{L,i}$ označeni proizvodnja i potrošnja aktivne snage u čvoru i , sa $Q_{G,i}$, $Q_{L,i}$ označeni su proizvodnja i potrošnja reaktivne snage u čvoru i , a sa Ω_i skup čvorova j koji su direktno povezani sa čvorom i ,

3. Specificirani moduo i fazni stav napona u napojnom čvoru ($i=0$):

$$V_0 = V_s, \quad (6)$$

$$\theta_0 = 0, \quad (7)$$

gde je V_s specificirana vrednost modula fazora napona napojnog čvora.

4. Ograničenje modula napona u čvorovima DM $i=1,2,\dots,N_N$:

$$V_i^{\min} \leq V_i \leq V_i^{\max}, \quad (8)$$

gde su $V_i^{\min}$ i $V_i^{\max}$ minimalni i maksimalni dozvoljeni napon u čvoru i , respektivno.

5. Ograničenje tokova snaga u granama distributivne mreže $l=1,2,\dots,N_L$:

$$(P_{ij})^2 + (Q_{ij})^2 \leq (S_l^{\max})^2, \quad (9)$$

gde je $S_l^{\max}$ maksimalna dozvoljena snaga grane l ,

6. Radijalna konfiguracija razmatrane DM:

$$\sum_{l=1}^{N_l} x_l = N_N - N_{SP}, \quad (10)$$

gde je N_{SP} broj napojnih čvorova u DM.

Izrazima (1)-(10) definisan je mešovito-celobrojni nelinearni optimizacioni model za rekonfiguraciju DM. Najveći problemi navedenog optimizacionog modela jesu: 1) nelinearnost i 2) izuzetno velika dimenzionalnost. Naime, u slučaju primene na realne DM koje sadrže više hiljada čvorova, opisani model sadrži više hiljada kontinualnih i binarnih promenljivih. Ovakve optimizacione probleme je veoma teško rešiti primenom dostupnih softverskih paketa za optimizaciju. Zato se u praksi za rešavanje rekonfiguracije DM mnogo češće koriste heurističke i meta-heurističke optimizacione metode, koje su prikazane u nastavku.

3. METODA IZMENE GRANA

Algoritam izmene grana (eng. *Branch Exchange*) se koristi tako da se na brz i jednostavan način ispitaju promene vrednosti zadate kriterijumske funkcije usled promene mesta normalno otvorenog (NO) rasklopnog uređaja u mreži. Cilj heurističkih metoda jeste da se poznajući fiziku određenog problema u što kraćem roku dođe do kvalitetnih praktičnih, ali suboptimalnih (lokalnih) rešenja [3].

Izmena grana (IG) zapravo znači izmenu mesta NO rasklopnog uređaja sa njime spregnutim normalno zatvorenim (NZ) rasklopnim uređajem. Pod spregnutim rasklopnim uređajima podrazumeva se par NO/NZ čijom

se izmenom mesta održava napajanje svih čvorova u mreži, ali i radijalnost mreže.

Velika prednost ovog metoda je da se promena mesta NO i NZ rasklopnih uređaja proračunava na bazi relacije u kojoj figurišu samo vrednosti iz baznog režima mreže. Na taj način, izbegnuta je potreba za proračunom tokova snaga (proračunom kompletnog režima) za novu konfiguraciju mreže. Dakle, ispituje se vrednost kriterijumske funkcije za NO uređaj i svaki njemu spregnuti NZ rasklopni uređaj, nakon čega se prihvata onaj slučaj gde je ostvareno poboljšanje kriterijumske funkcije. Tek nakon tog koraka, vrši se proračun tokova snaga koji predstavlja bazni režim za sledeću iteraciju [4].

Metoda izmene grana može se predstaviti sledećim pseudo kodom:

Ulaz: Inicijalizacija algoritma: /*Podaci o DM*/

$s = s_0$; /*Generisanje početnog dopustivog rešenja, npr. početna konfiguracija DM*/

$n_{NO} = 1$; /*Postavljanje brojača NO rasklopnih uređaja na 1*/

$k = 1$;

Ponavljati /*Globalna iteracija k*/

Ponavljati /*Za trenutno obrađivani NO rasklopni uređaj, odnosno u globalnoj iteraciji k*/

Generisati susedno rešenje (izvršiti zamenu mesta NO rasklopnog uređaja sa njemu spregnutim NZ rasklopnim uređajem), tj. konfiguraciju s' ;

$\Delta f = f(s) - f(s')$; /*Razlika ciljnih funkcija, tj. gubitaka u konfiguracijama s i s' */

Ako je $\Delta f > 0$ **Onda** $s = s'$; /*Prihvati s' kao tekuće rešenje*/

Dok /*Uslov zaustavljanja */**Dok** se ne obrade svi NZ rasklopni uređaji koji su u sprezi sa trenutno obrađivanim NO rasklopnim uređajem */

Dok Kriterijum zaustavljanja /*Dok se ne obrade svi otvoreni rasklopni uređaji u mreži*/

Izlaz: Najbolje pronađeno rešenje (s)

4. METODA SIMULIRANOG KALJENJA

Simulirano kaljenje (eng. *Simulated Annealing*) je metoda zasnovana na lokalnom pretraživanju, uz mehanizam inspirisan procesom kaljenja čelika koji omogućuje izlazak iz lokalnog optimuma [5].

U svojoj originalnoj formi metod simuliranog kaljenja (SK) baziran je na analogiji između fizičkog procesa kaljenja metala i kombinatorne optimizacije. U fizici čvrstih materijala, metalurško kaljenje materijala predstavlja postupak u kojem se materijal zagreva do maksimalne temperature (temperatura kaljenja) na kojoj je unutrašnja struktura materijala stohastički organizovana. Postepeno hlađenje ovako zagrejanog materijala ima za posledicu da se raspored unutrašnjih elementarnih čestica materijala menja, tako da one zauzimaju manja energetska stanja, prilagođavajući se temperaturi (unutrašnja energija elementarnih čestica je proporcionalna temperaturi) [5,6].

Polazeći od početnog dopustivog rešenja metoda SK se izvršava iterativno. U svakoj iteraciji se na slučaj način generišu rešenja koja su susedna (bliska) tekućem rešenju. Koraci (rešenja) koji unapređuju kriterijumsku funkciju se uvek prihvataju. U suprotnom rešenje se prihvata sa

nekom verovatnoćom koja zavisi od tekućeg parametra koji se naziva temperatura (T) (po analogiji sa procesom kaljenja metala) i od veličine degradacije (ΔE) ciljne funkcije. Degradacija (ΔE) predstavlja razliku u vrednosti ciljne funkcije (energije) tekućeg rešenja ($f(s)$) i novog (susednog) rešenja ($f(s')$). Kako se povećava broj iteracija, tj. kako se smanjuje temperatura T , verovatnoća prihvatanja lošijih rešenja se smanjuje. Ideja ovog postupka je da se prihvatanjem lošijih rešenja izbegne konvergiranje ka lokalnom optimumu, što je bio slučaj u metodi izmene grana.

Metoda simuliranog kaljenja može se predstaviti sledećim pseudo kodom:

Ulaz: Program (raspored) hlađenja: /*Početna temperatura(T_{max}), Krajnja temperatura(T_{min}), Tempo (brzina) hlađenja, Broj iteracija na određenoj temperaturi*/.

$s = s_0$; /*Generisanje početnog dopustivog rešenja, tj. početne konfiguracije*/

$T_k = T_{max}$; /* Početna temperatura*/

$k=1$;

Ponavljati /*Globalna iteracija, k */

Ponavljati /*Na određenoj temperaturi T_k , odnosno u globalnoj iteraciji k */

Na osnovu tekućeg rešenja s generisati susedno (blisko) rešenje, tj. konfiguraciju DM s' , npr. izmenom statusa jednog NO rasklopnog uređaja sa spregnutim NZ rasklopnim uređajem;

$\Delta E = f(s') - f(s)$; /*Razlika ciljnih funkcija, tj. gubitaka u ove dve konfiguracije */

Ako je $E \leq 0$ **Onda** $s = s'$ /*Prihvati s' kao tekuće rešenje*/

U suprotnom Prihvati s' sa verovatnoćom $e^{-\Delta E/T_k}$, odnosno prihvati s' ako važi: $e^{-\Delta E/T_k} > \text{rand}[0,1]$;

Ako je $f(s') - f(s_{best}) < 0$ **Onda** $s_{best} = s'$; /*Prihvati s' kao najbolje rešenje do sada (s_{best})*/

Dok Uslov zaustavljanja /*Dok se ne obrade svi otvoreni rasklopni uređaji u mreži */

$T_{k+1} = \alpha \cdot T_k$; /* Tempo (brzina) hlađenja, $\alpha \in [0.5 - 0.99]$ */

$k=k+1$;

Dok Kriterijum zaustavljanja /* $T_k < T_{min}$ */

Izlaz: Najbolje pronađeno rešenje (s_{best})

5. VERIFIKACIJA ALGORITMA OPTIMALNE REKONFIGURACIJE DM

Rekonfiguracija razmatrane test DM izvršena je primenom metode izmene grana, metode simuliranog kaljenja, kao i rešavanjem optimizacionog modela na bazi mešovito-celobrojnog programiranja. Na osnovu dobijenih rezultata ispitani su benefiti primene prikazanih postupaka za rešavanje problema rekonfiguracije.

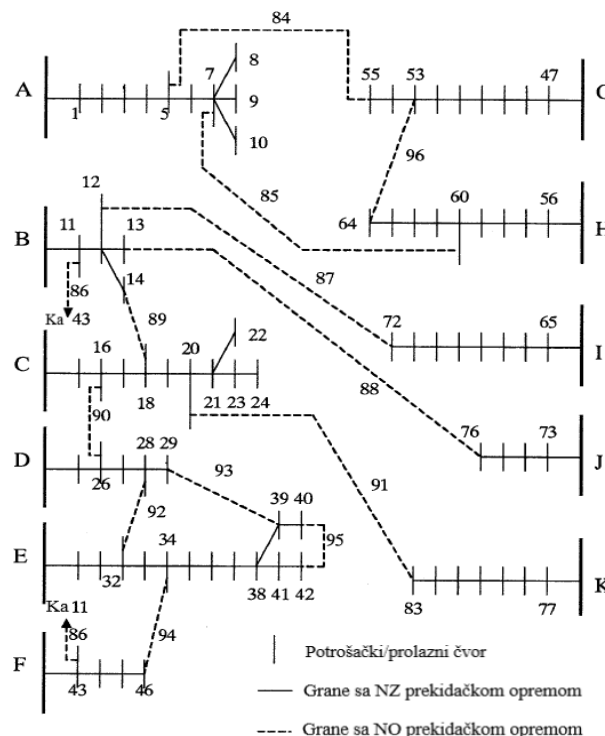
5.1. Opis test DM

Proračun se vrši na distributivnoj test mreži prikazanoj na slici 1. DM se sastoji od 11 sredjenaponskih izvoda označenih slovima od A do K. Nominalni napon DM mreže iznosi 11.4 kV. Parametri svih vodova i kablova dati su u [7].

Ukupan broj zatvorenih grana u mreži iznosi 83, dok je broj otvorenih grana 13. Svaka grana DM poseduje

rasklopni uređaj čime se omogućuje promena konfiguracije DM.

Rasklopni uređaji nisu posebno prikazani na slici 1 kako ne bi opterećivali sliku. Na slici 1 prikazani su početni i krajnji indeks čvora na svakom izvodu, kao i indeksi onih čvorova u kojima se stiče veći broj grana.



Slika 1. Prikaz Test DM [7]

5.2 Rezultati proračuna

Rezultati proračuna optimalne konfiguracije mreže primenom prethodno opisanih metoda prikazani su u tabeli 1.

Kao posledica inicijalno vrlo velikog opterećenja na izvodu A, dešava se da vrednosti napona čvorova DM padnu ispod dozvoljenih minimalnih vrednosti. U slučaju naponskog nivoa od 11.4 kV, dozvoljena donja granica napona bilo kog čvora u DM je $V_{min} = 0.95 \cdot 11.4 \text{ kV}$, odnosno $V_{min} = 10.83 \text{ kV}$.

Vrednosti napona svih čvorova koji u inicijalnoj konfiguraciji imaju manju vrednost od dozvoljene, određene su pokretanjem algoritma za proračun tokova snaga i prikazane u tabeli 1. Rezultati proračuna tokova snaga odgovaraju onima za početnu konfiguraciju (531.75 kW) datim u literaturi [7].

Rezultati prikazani u tabeli 1 pokazuju da proračun optimalne konfiguracije DM utiče i na naponske prilike u mreži. Pre samog proračuna naponi deset čvorova u mreži imali su vrednosti manje od donje naponske granice. Nakon završetka proračuna i preraspodele opterećenja po izvodima mreže postiže se i to da su naponi svih čvorova u okvirima prihvatljivih pogonskih vrednosti.

Ovi rezultati postižu se kako primenom metode IG, tako i metode SK. Naravno, postiže se i smanjenje pogonskih gubitaka aktivne snage sa inicijalnih 531.75 kW na 475.91 kW, tj. za 10.5 % u odnosu na inicijalno pogonsko stanje za slučaj kada se koristi metod IG.

Tabela 1. Rezultati proračuna

	Početna konfiguracija	Metoda IG	Metoda SK
Otvoreni rasklopni uređaji	84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	7, 13, 32, 35, 41, 61, 72, 82, 84, 86, 89, 90, 92	7, 13, 34, 62, 72, 83, 84, 86, 89, 90, 92, 93, 95
Gubici aktivne snage [kW]	531.75	475.91	470.69
Naponi čvorova u kojima su vrednosti napona narušene, tj. manje od $V_{\min}$	$V_{\min}$ [kV]	10.830	10.830
	V_4	10.773	11.044
	V_5	10.643	10.983
	V_6	10.620	10.973
	V_7	10.602	10.912
	V_8	10.595	10.905
	V_9	10.582	10.895
	V_{10}	10.594	10.904
	V_{71}	10.818	10.860
	V_{72}	10.816	11.194
	V_{83}	10.806	10.944
Broj promenjenih rasklopnih uređaja	-	12	12
Smanjenje gubitaka [%]	-	10.5 %	11.48 %

Sa druge strane primenom metode SK postiže se još bolji rezultat tako da po završetku proračuna ukupni gubici iznose 470.69 kW, što je za 11.48 % manje u odnosu na inicijalno pogonsko stanje. Ukupna injektirana aktivna snaga u mrežu se menja, njena vrednost primenom optimizacionih metoda IG i SK sa inicijalnih 28921.75 kW pada na 28859.91 kW odnosno 28820.69 kW. Na kraju, rešavanjem optimizacionog modela na bazi mešovito-celobrojnog programiranja koji je prikazan u sekciji 2 rada, nisu dobijeni očekivani rezultati. Naime, navedeni optimizacioni model je implementiran u programskom jeziku AMPL i rešavan na on-line NEOS serveru [8]. Međutim primenom dostupnih softverskih alata (eng. *solvers*) nisu dobijeni rezultati ni nakon nekoliko sati izvršavanja.

6. ZAKLJUČAK

U ovom radu je opisano rešavanje rekonfiguracije DM primenom heurističkih i meta-heurističkih optimizacionih metoda, kao i metoda zasnovanih na mešovito-celobrojnog programiranju. Cilj je bio da se primenom meta-heurističkog pristupa dođe do boljih rezultata, za razliku od heurističkih metoda koje se često sreću u praksi, ali omogućavaju postizanje samo lokalnih optimuma. Na osnovu rezultata koji su prikazani u sekciji 5 vidi se da je to i postignuto, s obzirom da su ukupni gubici dobijeni primenom metode SK manji od onih koji su dobijeni metodom IG. Vrednosti ukupnih gubitaka ne razlikuju se previše, ali dobijeni rezultati su očekivani s obzirom da se metodom izmene grana dobijaju lokalni optimumi, dok je cilj metode simuliranog kaljenja da se dodje do globalnog optimuma. Da bi se na još bolji način sagledali prethodno opisani efekti ove metode trebalo bi primeniti na mrežu čija je dimenzionalnost

višestruko veća od mreže koja je korišćena pri izradi ovog rada.

Optimizacioni modeli za rekonfiguraciju distributivnih mreža zasnovani na mešovito-celobrojnog programiranju pokazali su se kao praktično neupotrebljivi čak i u slučaju manjih DM, odnosno rezultati se dobijaju nakon isuviše dugo vremena. Zbog toga ovakav pristup rekonfiguraciji distributivnih mreža potpuno gubi na smislu kada su u pitanju realne DM velikih dimenzija.

7. LITERATURA

- [1] D.Popović, D.Bekut, V.Dabić: *Specijalizovani DMS Algoritmi*; Prosveta, Novi Sad, 2011.
- [2] V.Fathi, H.Seyedi, B.M.Ivatloo: *Reconfiguration of distribution systems in the presence of distributed generation considering protective constraints and uncertainties*; Int. Trans Electr Energ Systems; February 2020;pp. 1-25; DOI: 10.1002/2050-7038.12346
- [3] N.Kovački: *Operativno planiranje rekonfiguracije distributivnih mreža primenom višekriterijumske optimizacije*; Doktorska disertacija, Fakultet tehničkih nauka, Novi Sad, 2017.
- [4] D.Babić: *Optimalna rekonfiguracija distributivne mreže*; Master rad, Fakultet tehničkih nauka, Novi Sad, 2016.
- [5] Ž.Popović: *Planiranje razvoja distributivnih mreža u prisustvu neizvesnosti*; Doktorska disertacija, Fakultet tehničkih nauka, Novi Sad, 2011.
- [6] M.Stojkov, A.Softić, M.Atić: *Optimization of switching conditions in distribution power subsystem*; Tehnički Vjesnik, 2015; 22(5): 1297-1303. 10.17559/TV-20141211120022.
- [7] Ching-Tyong Su, Chu-Sheng Lee: *Network reconfiguration of distribution systems using improved mixed-integer hybrid differential evolution*; Power Delivery, IEEE Transactions on. 18. 2003; 1022 - 1027. 10.1109/TPWRD.2003.813641.
- [8] <https://neos-server.org>

Kratka biografija



Dario Kanjo rođen je u Somboru 1996. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi odbranio je 2019. godine.



Neven Kovački rođen u Zrenjaninu, 1987. god. Diplomirao je, masterirao i doktorirao na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi redom 2010, 2011. i 2018. godine.

SERVIS ZA IZRADU PODLOGE ZA AKADEMSKO PRIZNAVANJE PERIODA MOBILNOSTI**SERVICE FOR CREATING A BASIS FOR ACADEMIC RECOGNITION OF MOBILITY PERIODS**

Vuk Bošković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Ovaj rad opisuje proces implementacije sistema za olakšavanje procesa izrade dokumenta za priznavanje perioda mobilnosti, kao i stvari koje su prethodile implementaciji.*

Ključne reči: *Studentska mobilnost, poslovni procesi, veb servisi.*

Abstract – *This paper describes the process of implementing a system to facilitate the process of drafting a document for recognizing mobility periods, as well as things that preceded implementation.*

Keywords: *Student mobility, business processes, web services.*

1. UVOD

Kontekst vremena u kojem živimo, jasno nam govori da ljudi imaju sve veću želju za putovanjima i upoznavanjem novih kultura. Ova potreba, naročito se izražava kod mladih ljudi, a studenti, kao većina ove grupe, često imaju problem da pronađu vremena i sredstva kako bi je zadovoljili.

Jedan od najboljih načina da se ovo reši jeste odlazak na studentsku mobilnost. Studentska mobilnost se može realizovati na više načina: studentskom razmenom, stručnom praksom u inostranstvu, studijama na više univerziteta i druge. Razmene, kao najčešći oblik mobilnosti sve su popularnije među studentima koji žele da probaju život u inostranstvu i sve je više programa preko kojih se mogu realizovati. Ovi programi, često pružaju finansijsku pomoć studentima, kako bi razmene bile pristupačne za sve i kako materijalna sredstva ne bi predstavljala prepreku za odlazak na razmenu.

Problem vremena koje će biti utrošeno na odlazak na mobilnost, odnosno problem produktivnosti na razmeni i stručnoj praksi rešava se različito, na različitim fakultetima. Fakultet tehničkih nauka, Univerziteta u Novom Sadu ovo pitanje rešava sastavljanjem Podloge za akademsko priznavanje perioda mobilnosti.

Ovaj dokument sadrži osnovne podatke o studentu koji ide na razmenu, osnov mobilnosti, u kom periodu će se mobilnost odviti i na kojoj instituciji. Ukoliko se radi o studentskoj razmeni, podloga sadrži dve tabele sa informacijama o ispitima koje će student slušati na

razmeni. Prva tabela sadrži ispite koji, ukoliko budu položeni, biće uvaženi za isti ili sličan predmet sa fakulteta sa kog student ide na razmenu. Da bi se predmet našao u ovoj tabeli potrebna je saglasnost predmetnog profesora sa matičnog fakulteta. Druga tabela sadrži one ispite koji će, ukoliko budu položeni, biti priznati na matičnom fakultetu kao dodatak diplomu. Popunjena podloga mora biti potpisana od strane šefa studijskog programa na kojem je student. Šef pregleda popunjenu podlogu i ukoliko ona zadovoljava sve uslove potpisuje je, kako bi podloga bila važeća po povratku studenta sa razmene.

Proces pravljenja dokumenta za priznavanje, traženja saglasnosti profesora, traženja potpisa šefa studijskog programa zahteva dosta vremena. Potrebno je da svaki pojedinac izdvoji dodatno vreme kako bi učestvovao u ovom procesu. Cilj ovog rada je olakšavanje procesa za izradu podloge, gde bi se učesnici procesa povezali preko mrežnog servisa, postavljenog kao poslovni proces koji na kraju sam generiše podlogu u PDF formatu i pruža mogućnost preuzimanja za potpis

2. TEORIJSKE OSNOVE

Studentske razmene, kao najčešći oblik studentske mobilnosti, mogu se realizovati preko više programa razmene. Univerziteti i fakulteti putem programa razmene stvaraju međusobnu saradnju. Ovim putem, studentima jednog univerziteta ili fakulteta je omogućeno da se prijave na konkurs za razmenu na instituciji sa kojom je ostvarena saradnja. Osim same mogućnosti da deo studija provede na stranoj instituciji, program razmene studentu često nudi i finansijsku podršku, odnosno određenu stipendiju tokom trajanja razmene, kako finansije ne bi predstavljale prepreku u odluci studenta da se prijavi na neki od aktivnih konkursa. Ovo je posebno bitno za studente iz zemalja koje nisu ekonomski razvijene i znatno povećava broj zainteresovanih za razmene.

Proces prijave na konkurs za studentsku razmenu predstavlja složen proces i još jednu prepreku u prijavljivanju na konkurs. Da bi se student prijavio na konkurs, potrebno je da sakupi svu neophodnu dokumentaciju i usmena odobrenja aktera procesa. Takođe, želja studenta je da, pored upoznavanja kulture i sticanja novih poznanstava, razmenu provede efikasno i sa akademske strane, odnosno da predmeti koje bude položio na stranoj instituciji budu priznati kao zamena za neke od predmeta na instituciji sa koje ide na razmenu. Ovo bi značilo da je potrebna neka vrsta garancije za

NAPOMENA:

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

studenta da će njegov trud i položeni predmeti na razmeni biti vrednovani kad se razmena završi. Rešenje ovog problema Fakultet tehničkih nauka, Univerziteta u Novom Sadu, pronalazi u potpisivanju posebnog dokumenta pre početka razmene.

3. SPECIFIKACIJA ZAHTEVA

Ovo poglavlje će se baviti specifikacijom zahteva koje je potrebno ispuniti kako bi sistem bio što efikasniji.

3.1. Prikupljanje zahteva

Za svaki inženjerski problem, čije rešenje predstavlja neki softverski proizvod, neophodno je da se na početku rešavanja, jasno definišu zahtevi koje bi taj proizvod trebalo da ispunjava.

Kako bi se zahtevi definisali, potrebno je da se pribegne nekom od načina prikupljanja zahteva. Autor ovog rada izabrao je intervjuisanje za metodu prikupljanja zahteva. Ovo je najčešća metoda prikupljanja zahteva i podrazumeva anketiranje korisnika sistema za koji se softver pravi.

3.2. Definisanje poslovnog procesa

Sve prikupljene informacije su obrađene u cilju što boljeg shvatanja izvršavanja procesa. Iz svih informacija se jasno može videti da je ceo proces vrlo složen, sa posebnim naglaskom na proces izrade Podloge.

Proces za izradu ovog dokumenta uključuje više učesnika, odnosno više tipova učesnika koji imaju različite zadatke ili aktivnosti koje izvršavaju u okviru procesa. Svi ovi zadaci su međusobno koordinisani i vode do jednog zajedničkog cilja – izrade

Podloge za akademsko priznavanje perioda mobilnosti. Pošto proces sadrži sve navedene karakteristike, zaključeno je da može biti posmatran kao poslovni proces [2].

3.3. Slučajevi korišćenja

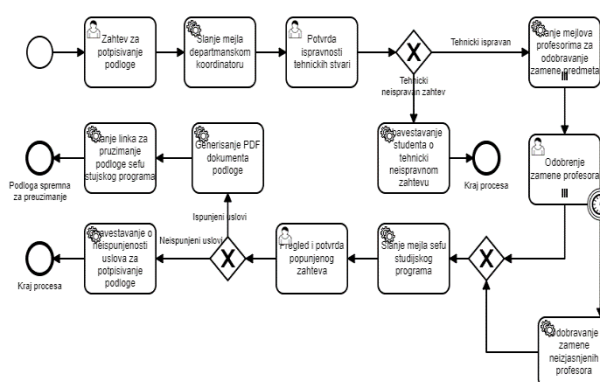
Za dobro razumevanje problema izrade Podloge za akademsko priznavanje perioda mobilnosti definisani su slučajevi korišćenja kroz tabelarni prikaz. Svaki slučaj je detaljno prikazan u tabeli, kroz opis, učesnike, uslove koje je potrebno ispuniti pre, kao i posledicama korišćenja.

4. IMPLEMENTACIJA

U ovom poglavlju biće omogućen pregled implementacije ranije definisanog sistema i procesi koji su joj prethodili, kao što su modeliranje poslovnog procesa implementiranja sistema i modelovanje podataka.

4.1. Poslovni proces i modelovanje podataka

S obzirom na prirodu procesa izrade podloge za akademsko priznavanje, ranije tokom ovog rada zaključeno je da će se gledati kao poslovni proces. Kao takav, imaće najveći uticaj na funkcionisanje sistema. Znajući ove činjenice, pre implementacije je definisan precizan model poslovnog procesa kojim će se serverska aplikacija voditi, a upravljanje njime Camunda framework omogućiti [10]. Model poslovnog procesa za funkcionisanje se može videti na slici 4.1.



Slika 4.1. Model poslovnog procesa sistema

Još jedna stvar koja prethodi implementaciji je modelovanje podataka. Neophodno je napraviti dobar model podataka kako bi baza podataka mogla efikasno da ih skladišti i time poboljša performanse sistema.

4.2. Implementacija funkcionalnih zahteva

Prvi zahtev koji je bilo potrebno ispuniti je omogućavanje studentu da podnese novi zahtev za izradu Podloge za akademsko priznavanje perioda mobilnosti. U svakom momentu, na svakoj stranici, pri vrhu prozora prisutan je navigacioni bar čija je uloga da omogući pristup raznim aktivnostima koje je moguće sprovoditi. Nakon pristupa aplikaciji, klikom na obeleženu komponentu navigacionog bara „Podnesi zahtev“, korisnik odlazi na stranicu na kojoj mu je ponuđen formular za prijavu. Formular je potrebno prvo popuniti osnovnim podacima o studentu kao što su: ime, prezime, broj indeksa, smer i osnovne informacije o periodu koji će provesti razmeni. Deo formulara u koji se tiče osnovnih informacija se može videti na slici 4.2.

Podnesi zahtev Saznaj stanje zahteva Recenzije stranih univerziteta

Ime

Ime

Prezime

Prezime

E-mail

Email

Smer

Broj indeksa

ra-77-2014

Naziv institucije gde se obavlja strucna praksa

Naziv institucije

Link institucije gde se obavlja strucna praksa

Link institucije

Godina studija na strucnoj praksi

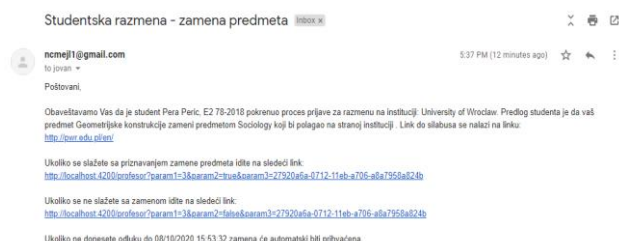
1. (osnovne)

Slika 4.2. Deo formulara za podnošenje zahteva

Drugi deo formulara tiče se predmeta koji će biti slušani na stranoj instituciji. S tim ciljem, korisniku je ponuđeno da u drugi deo forme ubaci podatke o predmetu koji će slušati i da odabere opciju da li taj predmet ima zamenu na njegovom fakultetu..

Nakon uspešno podnetog zahteva studenta, izvršava se prvi zadatak u modelu poslovnog procesa, a nakon njega se automatski izvršava servisni zadatak – „Slanje mejla departmanskom koordinatorskom i aktivira sledeći korisnički zadatak „Potvrda tehničke ispravnosti zahteva“ koji se dodeljuje departmanskom koordinatorskom. Isti svom zadatku pristupa putem linka dobijenog elektronskom poštom, sa objašnjenjem šta je potrebno da uradi kao učesnik procesa. Link u sebi, pored putanje do stranice sadrži i jedan parametar. Vrednost tog parametra predstavlja ID instance poslovnog procesa i omogućava pregled tačno određenog podnetog zahteva. Nakon pregleda, departmanski koordinatorski donosi odluku o tehničkoj ispravnosti. Ukoliko je odluka da zahtev nije tehnički ispravan, izvršiće se automatski servisni zadatak koji studentu šalje mejl sa obaveštenjem o tome i komentarom ukoliko ga je koordinatorski ostavio, a izvršavanje procesa će se prekinuti. Sa druge strane, ukoliko je odluka potvrđna, nastaviće se sa izvršavanjem i pokrenuće se servisni zadatak za slanje mejla profesorima čiji su predmeti predloženi za zamenu.

Profesori moraju imati mogućnost da potvrde da li se slažu sa zamenom njihovog predmeta, nekim drugim predmetom sa strane institucije. Ovaj zahtev im je ispunjen veoma prostim rešenjem. Nakon potvrđne odluke koordinatorskog, svi profesori dobijaju mejl koji sadrži objašnjenje o tome šta je predlog studenta, svim podacima o tome i koji je njihov zadatak u celom procesu. Izvršavanje tog zadatka podrazumeva donošenje odluke o slaganju sa zamenom njihovog predmeta i rešava se klikom na jedan od dva linka iz mejla. Odlaskom na prvi link, profesor se izjašnjava potvrđno sa zamenom predmeta, a klikom na drugi odrično. Izgled strukture jednog ovakvog mejla može se pogledati na slici 4.3.



Slika 4.3. Primer mejla za profesora

Znajući da su profesori često zauzeti i da zbog obimnosti posla, puno studenata ili drugih obaveza, nekad nemaju vremena da donose ovakve odluke u kratkom vremenskom intervalu, bilo je potrebno vremenski ograničiti izvršavanje njihovog zadatka. Ovaj zahtev se rešio dodavanjem tzv. Boundary timer event-a na zadatak profesora. Time je omogućeno vremensko ograničenje, koje, ukoliko se ne ispoštuje, preusmerava izvršavanje procesa na alternativni tok izvršavanja. Model poslovnog procesa sada preusmerava tok na servis koji automatski prihvata zamene predmeta svih profesora koji se nisu izjasnili do tada. Nakon ovog zadatka, alternativni tok se vraća prvobitnom toku, zamišljenom u slučaju da se svi profesori na vreme izjasne.

Skoro isto kao departmanskom koordinatorskom, šefu studijskog programa je omogućeno da izvrši svoj zadatak. Nakon dobijenog mejla u kojem je obavešten da su

profesori doneli odluku i linka sa parametrima koji ih vodi do stranice za izvršavanje zadatka, pregledaju zahtev i ispunjenost uslova. Zatim donose odluku i ostavljaju komentar ukoliko smatraju da ima potrebe za tim. Ukoliko je odluka negativna, studentu se šalje mejl sa razlogom odbijanja njegovog zahteva za izradu podloge, sa komentarom ili savetom šefa, ukoliko ga je ostavio. Sa druge strane, ukoliko je odluka pozitivna, proces pokreće servisni zadatak za izradu PDF dokumenta. Za implementaciju servisa je korišćena biblioteka iText PDF koja omogućava rad sa PDF dokumentima. Primer jedne ovako izgenerisane podloge može se pogledati na slici 4.4.

UNIVERZITET SLOBODNE DRŽAVE SLOVENIJE
Ljubljana, 021 6350-413; 021 458-810; Centralni: 021 485 2000
Faks: 021 458-133; e-mail: fmd@univ.si

INSTITUCIJA
SISTEM
IMPLEMENTACIJE
CERTIFIKOVANOSTI

Datum: _____
Broj: _____

Student: Pera Peric (Br. indeksa: E2 78-2018)
Studijski program: Arhitektura
Godina studija na kojoj će biti provedena razmena/stručna praksa: 1. (master)
Semestar/semestri koje će biti provedena razmena/stručna praksa: 2. (zimski)
Semestar akademske godine: 2022/23
Period mobilnosti (za mobilnost krađu od jednog semestra): od _____ do _____
Ustanova-primalac: Institucija
Osnov razmene/stručne prakse: Erasmus+

PODLOGA ZA AKADEMSKO PRIZNAVANJE PERIODA MOBILNOSTI

kojom se utvrđuje sledeće:

- I) Predmeti koje će student slušati na ustanovi-primalcu, a ispunjavaju uslove da u potpunosti budu priznati i zamenjeni relevantnim predmetom Fakulteta tehničkih nauka:

Predmet koji će biti slušan na ustanovi-primalcu	Broj ECTS bodova koje će student ostvariti na ustanovi-primalcu	Predmet matične ustanove sa kojim se ispitni položaj na razmeni/zamenuje	Broj ECTS koji će se priznati
1. Neural Networks	5	Neuronske mreže	5
2. Databases	7	Baze podataka	6

- II) Predmeti koje će student slušati na ustanovi-primalcu, a koji se ne mogu zameniti srodnim predmetom Fakulteta tehničkih nauka, ali će se naznačiti u Dodatku diplomi:

Predmet koji će biti slušan na ustanovi-primalcu (na izvornom)	Broj ECTS koje će student ostvariti na ustanovi-primalcu	Fond časova na ustanovi-primalcu
1. New Subject 1	6	
2. New Subject	7	

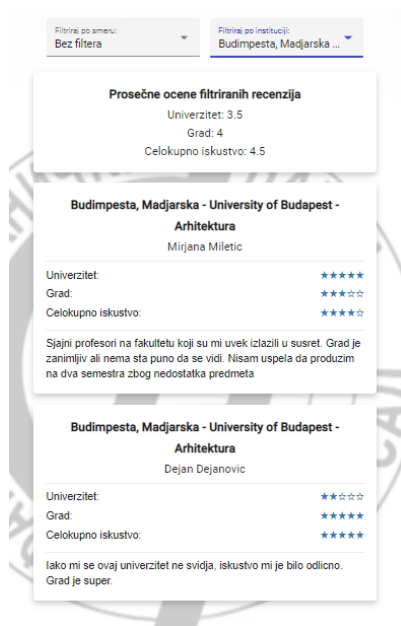
Slika 4.4. Primer izgenerisane podloge

Nakon servisa za generisanje podloge, shodno toku procesa, izvršava se i servis za slanje elektronske pošte. Šefu studijskog programa se šalje mejl sa linkom na kom može izvršiti preuzimanje podloge. Za dobavljanje PDF fajla sa serverske aplikacije i omogućavanje njegovog preuzimanja sa klijentske aplikacije korišćena je biblioteka File-saver.

4.3. Implementacija nefunkcionalnih zahteva

Problem motivacije studenta za odlazak na razmenu i odabir univerziteta, odnosno lokacije na kojoj će sprovedi razmenu, može se rešiti na više načina. Postoje različiti pristupi za rešavanje ovog problema. Pristup koji je korišćen u ovom radu je izrada sistema za ocenjivanje perioda provedenog na razmeni.

Ovo podrazumeva pružanje studentu mogućnost da u svakom momentu može da pogleda recenzije univerziteta, lokacija i iskustava drugih studenata, koji su tamo ranije sprovedi svoju razmenu. Slika 4.5. sadrži prikaz slučaja korišćenja funkcionalnosti izradene sa ciljem rešavanja ovog problema.



Slika 4.5. Primer korišćenja servisa za pregled recenzija

Da bi se recenzije mogle prikazivati, potrebno je da budu unete u bazu podataka. S toga je dodata još jedna funkcionalnost u sistem – dodavanje recenzije. Kako bi se implementirala ova funkcionalnost, u bazu podataka su uneti test podaci o studentima koji su bili na razmeni. Nakon završene razmene, studentima se šalje generički mejl sa molbom da napišu recenziju svog perioda provedenog na razmeni i tako pomognu ostalim studentima da steknu neku okvirnu sliku o njima. Uz molbu, u poruci se nalazi i link, na kom student može obaviti zadatak dodavanja recenzije. Link u sebi sadrži parametar čija vrednost predstavlja identifikacioni broj tog studenta pod kojim se čuva u bazi podataka. Na osnovu ove vrednosti, sistem prepoznaje koji student je napisao recenziju.

5. ZAKLJUČAK

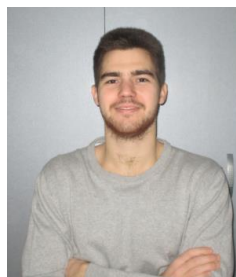
Inženjerski sistemi imaju za cilj olakšavanje posla njihovim korisnicima, a to je slučaj i sa sistemom opisanim u ovom radu. Cilj spajanja učesnika na brz način, ispunjen je u potpunosti. Učesnici ovog procesa mogu uz par klikova mišem da izvrše svoje zadatke, što govori o efikasnosti načina na koji je ovaj problem rešen. Camunda framework pokazao se kao dobar izbor za alat koji će upravljati poslovnim procesom i time se preporučio za druge radove koji rade sa poslovnim procesima. Sa druge strane, jasno je da ovaj sistem ima mnogo potencijala za proširenje u različitim pravcima. Prvo što se nameće kao moguće proširenje je dodatak koji bi se bavio poslovnim procesom cele prijave na konkurs za razmenu, a ne samo izradom Podloge.

Ovo rešenje bi moglo da pomogne pri prikupljanju dokumentacije, njenom organizovanju i olakša prikupljanje potpisa neophodnih za ostalu dokumentaciju. Još jedno potencijalno proširenje bi mogao predstavljati dodatak za prijavu stranih studenata koji bi aplicirali za dolazak na Fakultet tehničkih nauka. Ovim bi se i broj dolaznih studenata na Fakultet tehničkih nauka povećao kao i ukupan broj studenata koji odlaze na razmenu, što je jedan od ciljeva ovog rada.

6. LITERATURA

- [1] Z. Gračak and L. Brkić, "Software supporting international student exchange program in Higher Education," 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, 2017, pp. 1528-1533, doi: 10.23919/MIPRO.2017.7973664.
- [2] Brumec, J., & Brumec, S. (2011). Modeliranje poslovnih procesa. Varaždin: Koris.
- [3] Bjelica, D. (2020). SISTEM ZA AUTOMATIZACIJU PROCESA IZDAVANJA LICENCI ZA FIZIČKO-TEHNIČKO OBEZBEĐENJE. Zbornik radova Fakulteta tehničkih nauka u Novom Sadu, 35(04), 805-808.
- [4] Suwanto, N. A. (2015). Student Exchange Program Web Based Application System (Outbound). IRC.
- [5] Mustafar, S. N. (2015). INBOUND STUDENT EXCHANGE PROGRAM WEB-BASED SYSTEM. IRC.
- [6] Pettersen, T. M., & Andersen, L. L. (2017). Developing a Web Application and a Case-based Reasoning Recommender System to Improve Students' Motivation for Exchange Programs (Master's thesis, NTNU).
- [7] Ramlan, M. I. I. (2014). UTP Exchange Student Program System (Outbound). IRC.
- [8] MAZALAN, M. A. (2014). Web Based Application: An Online Management System for the Student Exchange Program in Universiti Teknologi PETRONAS. IRC.
- [9] Dirks, R. (2010). The experiences of student exchange program participants: a phenomenological study.
- [10] The Camunda BPM Manual. (2020). Preuzeto 5. avgusta 2020., sa <https://docs.camunda.org/>
- [11] Rodriguez A., (2008). "RESTful Web services: The basics", IBM.
- [12] Angular. Preuzeto 10. septembra 2020., sa <https://angular.io/>
- [13] Bruegge B. & Dutoit A. (2009). Object-Oriented Software Engineering Using UML, Patterns, and Java (3. izd.). Pearson.
- [14] Ilić, J. (2020). APLIKACIJA ZA UPRAVLJANJE RAZMENOM STUDENATA PUTEM ERASMUS+ PROGRAMA. Zbornik radova Fakulteta tehničkih nauka u Novom Sadu, 35(03), 474-477.
- [15] Portal mobilnosti studenata, (2020). Preuzeto 5. jula 2020., sa <http://mobility.ftn.uns.ac.rs/>

Kratka biografija:



Vuk Bošković rođen je 1994. god. u Novom Sadu. Završio je gimnaziju „Laza Kostić“ u Novom Sadu 2013. godine. Fakultet tehničkih nauka je upisao iste godine. Odranio je diplomski rad 2018. godine na temu „Programski jezik Kotlin“, nakon čega nastavlja sa master studijama na istom fakultetu.
kontakt: vukboskovic3@gmail.com

INTEGRACIJA FIREBASE CLOUD MESSAGING SISTEMA U VEB I MOBILNE APLIKACIJE**INTEGRATION OF FIREBASE CLOUD MESSAGING SYSTEM IN WEB AND MOBILE APPLICATIONS**

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

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – Zadatak rada predstavlja razvoj veb i mobilne aplikacije uz korišćenje Firebase Cloud Messaging (FCM), koji omogućava slanje poruka bez ikakvih troškova. Kao primer kreirana je aplikacija za teretane, koja nudi svojim klijentima mogućnost zakazivanje grupnih treninga uz stizanje notifikacija sat vremena pre zakazanog treninga. Klijentski deo veb aplikacije implementiran je pomoću Angular Framework-a, dok je serverski deo razvijen pomoću Express Framework-a. Mobilna aplikacija napravljena je isključivo za Android operativni sistem. Svi podaci se čuvaju u MongoDB.

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

Abstract – The aim of this paper is to develop a web and mobile application using the Firebase Cloud Messaging (FCM, which allows sending messages at no cost. As an example, an gym application is created, which offers its clients the possibility of scheduling group trainings with receiving notifications an hour before the scheduled training. The client part of the web application is implemented using the Angular Framework, while the server part is developed using the Express Framework. The mobile application is made exclusively for the Android operating system. All data is stored in MongoDB.

Key words: Angular, Express, Android, MongoDB, Mongoose, FCM

1. UVOD

Kada je reč o tehnologijama koje su korišćenje prilikom razvijanja aplikacije, serverski deo, koji je zajednički za veb i mobilnu aplikaciju, razvijen je upotrebom *Express Framework-a*. *Angular Framework*, kao jedan od najpopularnijih okruženja za razvoj veb aplikacija, korišćen je za izradu klijentskog dela veb aplikacije.

Mobilna aplikacija napravljena je isključivo za *Android* mobilni operativni sistem, za uređaje sa *Android 4.1* ili novijom verzijom zbog potrebe FCM klijenata kako bi uspostavili slanje poruka.

Svi podaci veb i mobilne aplikacije čuvaju se u *MongoDB*. Mapiranje objekata između *MongoDB-a* i serverskog dela ostvaruje se putem *Mongoose-a*.

NAPOMENA:

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

2. TEHNOLOGIJE

U ovom poglavlju osvrt je dat na tehnologije pomoću kojih su realizovane veb i mobilna aplikacija. Za svaku tehnologiju stavljen je fokus na bitne karakteristike koje su potrebne za dalje razumevanje implementirane aplikacije.

2.1. Angular

Angular 2+ predstavlja okvir i razvojnu platformu za stvaranje efikasnih klijentskih aplikacija na jednoj stranici (SPA - Single Page Apps) koristeći HTML i TypeScript [1]. Komponente predstavljaju osnovni gradivni element angular aplikacije. Svaka angular aplikacije ima bar jednu komponentu, takozvanu root komponentu (listing 1).

```
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent {
}
```

Listing 1 – Primer root komponente

Svaka komponenta sastoji se od HTML šablona koji definiše šta se prikazuje na stranici, TypeScript klase koja definiše ponašanje, CSS selektora koji definiše kako se komponenta koristi u šablonu i CSS fajla u kom su definisani stilovi primenjeni na HTML šablonu.

2.2. Express

Express je najpopularniji *Node.js* okvir, koji je posebno dizajniran za izgradnju veb i mobilnih aplikacija, predstavlja osnovnu biblioteku za brojne druge *Node.js* veb okvire. Izgrađen je na vrhu *Node.js* http modula i dodaje podršku za rutiranje. Omogućava programerima da kreiraju i održavaju server. Dolazi sa mnogim ugrađenim funkcijama, takođe ima i ogroman paket nezavisnih dodataka koje programeri mogu koristiti za pružanje bolje funkcionalnosti, poboljšanje brzine i povećavanje sigurnosti [2]. Programeri koriste Express okvir za pisanje logike na strani servera u JavaScript-u za veb i mobilne aplikacije.

2.3. Firebase Cloud Messaging

Firebase Cloud Messaging (FCM) predstavlja rešenje za razmenu poruka na više platformi koje omogućava pouzdano slanje poruka bez ikakvih troškova. Koristeći FCM možete da šaljete poruke obaveštenja da biste podstakli ponovno angažovanje i zadržavanje korisnika

[3]. Pored toga FCM pruža mogućnost obaveštavanja klijentske aplikacije da su drugi podaci dostupni za sinhronizaciju, kao i razmenu trenutnih poruka pri čemu poruka može da sadrži teret veličine 4KB.

FCM pruža pouzdanu i efikasnu vezu između servera i uređaja koja omogućava da poruke i obaveštenja budu primljena i isporučena na *IOS* ili *Android* uređajima kao i na veb aplikaciji bez troškova.

2.4. Android

Android je mobilni operativni sistem kompanije Google, dizajniran za mobilne uređaje sa ekranima osljetivim na dodir kao što su pametni telefoni i tablet uređaji [4]. Android predstavlja softver otvorenog koda zasnovan na Linux jezgri, razvijen za širok spektar uređaja.

Android aplikacije mogu biti napisane pomoću jezika Java, C++ i Kotlin. Osnovni gradivni elementi Android aplikacije su komponente. Komponenta predstavlja ulaznu tačku kroz koju korisnik ili sistem mogu da pristupe aplikaciji. Postoje različite komponente kao što su aktivnosti, usluge i dobavljači sadržaja.

2.5. MongoDB

MongoDB je vodeća nerelaciona tj. *NoSQL* baza podataka. U skladu sa interesovanjem za alternativne sisteme za upravljanje bazama podataka, koji se razlikuju od relacionih baza podataka, pojavio se koncept takozvanih *NoSQL* baza podataka, koje ne koriste SQL (*Structured Query Language*) za povezivanje. *NoSQL* baze podataka su nerelacione, distribuirane, otvorenog koda i horizontalno skalabilne [5].

MongoDB 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. Dokumenti su organizovani unutar kolekcija pri čemu jedna kolekcija obično sadrži dokumente sa sličnom namenom. Dokument pored primitivnih tipova podržava nizove i ugneždene objekte kao vrednosti što poboljšava performanse prilikom čitanja i pisanja dokumenata.

MongoDB i Node.js se često koriste zajedno zbog zajedničke upotrebe JavaScripta-a. Mapiranje objekata između MongoDB-a i Node.js-a ostvaruje se putem Mongoose-a. Na listingu 2 dat je prikaz Mongoose šeme.

```
const mongoose = require('mongoose');
const classSchema = mongoose.Schema({
  name: { type: String, required: true},
  instructor: { type: String, required: true},
  location: { type: String, required: true},
  timeStart: { type: String, required: true},
  timeEnd: { type: String, required: true},
  description: { type: String, required: true}
});
module.exports = mongoose.model('class', classSchema);
```

Listing 2 – Primer Mongoose šeme

3. SPECIFIKACIJA ZADATAKA

Zadatak rada obuhvata kreiranje veb i mobilne aplikacije za zakazivanje grupnih treninga u datoj teretani. Aplikacija je razvijena tako da podržava komunikaciju sa

FCM sistemom koji služi za slanje obaveštenja korisnicima koji su zakazali trening.

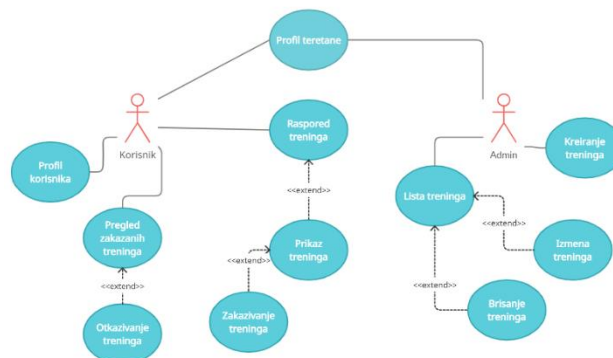
Aplikacija podržava dva tipa korisnika, a to su korisnici usluga koje teretana pruža i administrator sistema, koji održava aplikaciju. Mobilna aplikacija razvijena je samo za potrebe korisnika teretane, dok veb aplikaciji mogu da pristupe i administratori.

Prilikom dolaska na veb stranicu aplikacije korisnicima je omogućena prijava na sistem. Prijava se vrši putem email adrese i lozinke. Ukoliko neko od korisnika nema nalog nudi mu se opcija registracije na sistem.

Nakon uspešne autentifikacije, običnom korisniku se prikazuje raspored treninga za narednih sedam dana, pregled svih zakazanih treninga kao i istorija svih prethodnih zakazanih treninga koji su prošli, profil korisnika i profil teretane koji sadrži informacije o datoj teretani.

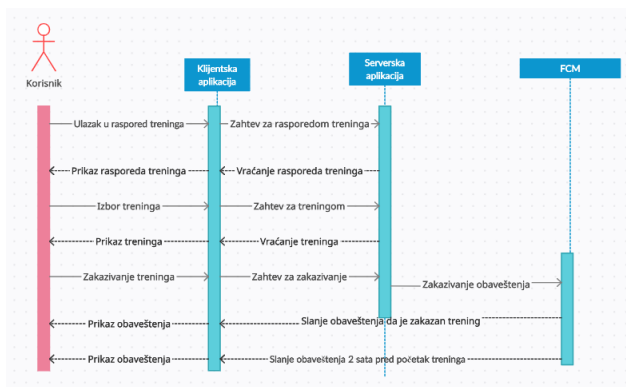
Korisnik ima mogućnost detaljnog prikaza svakog treninga koji se nalazi na rasporedu za narednih sedam dana, kao i mogućnost rezervisanja datog treninga. Korisnik pored rezervacije ima i opciju otkazivanja treninga.

Administratoru se nakon uspešne prijave na sistem, prikazuje lista svih treninga koji su kreirani, profil teretane koji može da uređuje i opcija za kreiranje novih treninga. Pored toga, administrator može da menja i uklanja postojeće treninge. Na slici 1 dat je prikaz dijagrama slučajeva korišćenja prijavljenog korisnika.



Slika 1. Dijagram slučajeva korišćenja prijavljenog korisnika

Na slici 2 dat je prikaz dijagrama sekvence za zakazivanje treninga. Sekvenca događaja počinje korisnikovim ulaskom u raspored treninga za narednih sedam dana. Nakon što serverska aplikacija vrati klijentskoj aplikaciji raspored treninga, korisnik bira trening koji želi da zakaže. Prikazuje mu se detaljan opis treninga i opcija da zakaže. Nakon što korisnik odabere da zakaže trening, klijentska aplikacija šalje zahtev za zakazivanje treninga serverskoj aplikaciji, koja zakazuje obaveštenja kod FCM. FCM šalje prvo obaveštenje klijentskoj aplikaciji da je trening uspešno zakazan i prikazuje to korisniku u vidu obaveštenja u Toolbar-u. FCM šalje drugo obaveštenje klijentskoj aplikaciji sat vremena pre zakazanog treninga kako bi podsetila korisnika na trening, obaveštenje je takođe prikazano u Toolbar-u



Slika 2. Dijagram sekvence

4. IMPLEMENTACIJA

Za potrebe rada razvijene su veb i mobilna aplikacija. Obe aplikacije sastoje se iz dva dela, klijentskog i serverskog. Serverski deo je zajednički za obe aplikacije. U daljem nastavku teksta biće detaljnije objašnjene implementacije obe aplikacije.

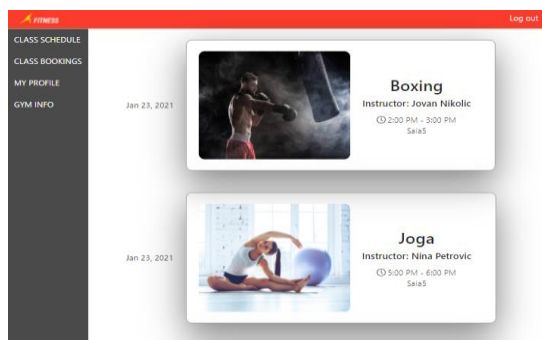
4.1. Klijentski deo veb aplikacije

Klijentski deo veb aplikacije, razvijen korišćenjem Angular Framework-a, podeljen je u tri modula. Prvi modul koji je takođe i podrazumevani modul predstavlja autentifikacioni modul. Rute iz ovog modula su dostupne prilikom dolaska na veb sajt, dostupne su samo neprijavljenom korisniku. Rute koje se nalaze u ovom modulu su rute za prijavu i registraciju na sistem. Prijavljen korisnik u zavisnosti od uloge može da pristupi ostalim modulima. Ukoliko je prijavljen običan korisnik, rutira se na modul app, a ukoliko je prijavljen administrator, rutira se na admin modul.

Nakon uspešne autentifikacije podaci o prijavljenom korisniku se čuvaju u *Local Storage*-u. Pored podataka o korisniku, čuva se i token koji se kreira od strane servera za svakog korisnika kada se uloguje, koji u sebi sadrži i rok važenja.

Kako ne bi moglo da se pristupi modulima koji dozvoljavaju pristup samo prijavljenim korisnicima, koriste se Angular-ovi zaštitnici ruta. U aplikaciji je za tu svrhu kreirana *AuthGuard* klasa koja implementira interfejs *CanActivate*. Svrha ove klase je da obavesti rutera na osnovu povratne vrednosti da li može da se pristupi traženoj ruti.

Ukoliko je prijavljen običan korisnik, nakon prijave prikazuje mu se početna strana, na kojoj je prikazan raspored treninga za narednih sedam dana (slika 3).



Slika 3. Prikaz početne strane korisnika

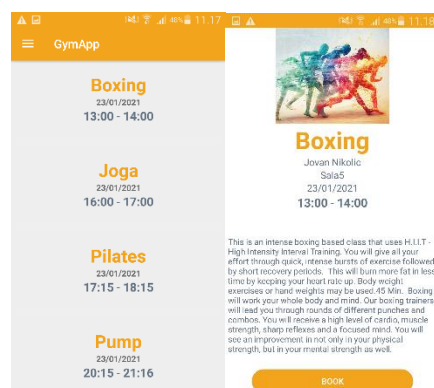
Korisniku se nudi mogućnost detaljnog prikaza treninga, klikom na karticu treninga u rasporedu treninga.. Detaljan prikaz sadrži informaciju o instruktoru, mestu održavanja treninga (broj sale), vreme održavanja i kratak opis treninga.

Ukoliko je korisnik zainteresovan za dati trening nudi mu se opcija zakazivanja treninga (*Book*).

4.2. Klijentski deo mobilne aplikacije

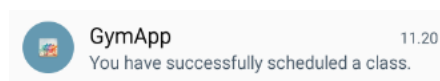
Klijentski deo mobilne aplikacije realizovan je kao Android aplikacija, za uređaje sa Android 4.1 ili novijom verzijom zbog potreba FCM klijenta. Android aplikacija namenjena je isključivo korisnicima teretane. Pokretanjem aplikacije korisniku se prikazuje forma za prijavu na sistem.

Unošenjem email adrese i lozinke korisniku se dozvoljava pristup njegovom nalogu i prelazak na početnu stranu (slika 4 - levo).



Slika 4 Prikaz početne strane (levo) i detaljan prikaz treninga (desno)

Selektovanjem konkretnog treninga, poziva se aktivnost za detaljan prikaz treninga (slika 4 - desno). Korisniku se nudi mogućnost zakazivanja treninga klikom na dugme *Book*. Nakon što korisnik zakaže trening prikazuje mu se obaveštenje u *Toolbar*-u, u vidu potvrde da je uspešno zakazao trening (slika 5). Slanje obaveštenja realizovano je uz pomoć FCM klijenta.



Slika 5. Prikaz obaveštenja u Toolbar-u

Da bi se kreirala klijentska aplikacija sa FCM za Android, potrebno je koristiti *FirebaseMessaging API* i *Android Studio 1.4* ili većom verzijom sa *Gradle*-om.

FCM pruža pouzdanu i efikasnu vezu između servera i uređaja koja omogućava besplatnu isporuku i primanje poruka i obaveštenja. Pre nego što se doda *Firebase* u projekat, potrebno je registrovati projekat na *Firebasu*-u.

4.3. Serverski deo

Serverski deo aplikacije kao što je već navedeno, implementiran je korišćenjem Express Framework-a. Serverska strana zadužena je za uspostavljanje komunikacije sa MongoDB preko Mongoose-a, koji predstavlja biblioteku za objektno modeliranja podataka između MnogoDB i Node.js. Nakon kreiranja Express aplikacije, uspostavlja se konekcija sa bazom podataka preko Mongoose-a, postavljaju se rute koje su vidljive klijentskim aplikacijama na podrazumevanom portu 3000. Rute su podeljene u tri grupe: autentifikacione rute, rute vezane za trening i rute vezane za teretanu. Takođe podešen je pristup rutama samo ukoliko se u zaglavlju zahteva koji stiže sa klijentske strane nalazi token, osim za autentifikacione rute kojima je dozvoljen pristup bez tokena.

Mongoose-ova šema definiše strukturu dokumenta, podrazumevane vrednosti i validatore. Sve u Mongoose-u počinje sa šemom, svaka šema se preslikava na MongoDB kolekciju i definiše oblik dokumenta u kolekciji.

Sve funkcionalnosti vezane za trening nalaze se u zasebnom kontroleru. Samo prijavljeni korisnici mogu da pristupe rutama kontrolera. Kontroler iz svakog HTTP zahteva preuzima parametre zahteva i prosleđuje ih servisu, u kome je izmeštena sva logika. Servis zatim vraća rezultat kontroleru, u vidu statusa i vrednosti samog rezultata nakon čega kontroler šalje tu povratnu vrednost klijentskoj aplikaciji. Sve funkcije u servisu su asihrone i kao povratnu vrednost vraćaju *Promise*. Pošto su funkcije asihrone, dozvoljeno je korišćenje *await* izraza. *Await* izrazi se stavljaju na pozive upućene ka bazi, kako bi se zaustavio napredak kroz funkciju.

Prilikom zakazivanja treninga, pored ažuriranja korisnikove liste zakazanih treninga u bazi podataka, šalje se prvo obaveštenje da je uspešno zakazan trening i vrši se zakazivanje obaveštenja koje treba da se pošalje sat vremena pre početka zakazanog treninga. Implementacija metode za slanje obaveštenja prikazana je na listingu 3. Kako bi se obaveštenje poslalo na određeni uređaj, potrebno je proslediti registracioni token.

```
firebaseAdmin.sendNotification = function (payload) {
  const message = {
    notification: {
      title: payload.title,
      body: payload.body,
    },
    token: payload.token,
  };
  return admin.messaging().send(message);
};
```

Listing 3. Implementacija metode za slanje obaveštenja

Obaveštenje koje treba da stigne sat vremena pre početka treninga zakazuje se pomoću *node-schedule*. Node Schedule omogućava da se zakazuju obaveštenja za određenje datume i vreme. Svako zakazano obaveštenje u rasporedu predstavljeno je objektom.

5. ZAKLJUČAK

Osnovni zadatak ovog rada bio je kreiranje veb i mobilne aplikacije uz korišćenje FCM za slanje obaveštenja nakon zakazivanja treninga kao i obaveštenja sat vremena pre početka zakazanog treninga. Rad obuhvata osnovne funkcionalnosti koje teretana nudi svojim korisnicima poput pregleda rasporeda treninga za narednih sedam dana, zakazivanje i otkazivanje treninga, kao i pregled liste zakazanih treninga. Dalja proširivost ovog zadatka, podrazumevala bi proširivost osnovnih funkcionalnosti koje se nude korisniku kako za veb tako i za mobilnu aplikaciju.

Prilikom implementacije zadatka korišćene su najnovije tehnologije koje se koriste za kreiranje veb i mobilnih aplikacija, sa ciljem lakog održavanja, i jednostavne i efikasne nadogradnje. FCM koji je korišćen u izradi ovog zadatka, predstavlja trenutno najefikasnije rešenje za razmenu poruka na više platformi bez ikakvih troškova. FCM je konstantno u procesu razvoja što pruža mogućnost proširenja u budućnosti. Takođe korišćenjem savremenih i trenutno aktuelnih tehnologija, stvaraju se mogućnosti za nadogradnju aplikacije u budućnosti kako bi se pružila još lakša i brža interakcija između korisnika i teretane.

6. LITERATURA

- [1] Angular, <https://angular.io/>
- [2] Express, <https://expressjs.com/>
- [3] Firebase Cloud Messaging, <https://firebase.google.com/docs/cloud-messaging>
- [4] Android, <https://developer.android.com/>
- [5] MongoDB, <https://www.mongodb.com/>

Kratka biografija:

Ilinka Kovačević rođena je 14.11.1996. godine u Novom Sadu. Završila gimnaziju "Svetozar Marković" u Novom Sadu. Godine 2015 upisuje Fakultet tehničkih nauka, smer Računarstvo i automatika. Na trećoj godini studija opredeljuje se za smer Primenjene računarske nauke i informatiku. Diplomirala je 2019. godine te je upisala master akademske studije na istom fakultetu, smer Računarstvo i automatika, modul Softversko inženjerstvo. 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*.

PROVERA IDENTITETA KORISNIKA KORIŠĆENJEM QR KODA**USERS AUTHENTICATION USING QR CODE**David Vuletaš, *Fakultet tehničkih nauka, Novi Sad***Oblast – SOFTVERSKO INŽENJERSTVO I INFORMACIONE TEHNOLOGIJE**

Kratak sadržaj – U radu je opisan sistem za dvostruku verifikaciju identiteta korisnika skeniranjem QR koda, arhitektura aplikacije, komunikacija, model komponenti.

Ključne reči: QR kod, bezbednost, 2FA, Dvostruka verifikacija identiteta, C4 model

Abstract – The paper describes a system for two factor verification of user identity by scanning a QR code, application architecture, communication, component model..

Keywords: QR code, security, 2FA, Two factor verification of identity, C4 model

1. UVOD

Dvostruka verifikacija identiteta korisnika rešava neke od problema u očuvanju bezbednosti naloga dodavanjem dodatnog sloja bezbednosti. To zahteva izvršavanje još jednog dodatnog koraka od strane korisnika neposredno pre pristupanja resursu [1].

Koncept verifikacije identiteta korisnika u ovom radu ogleda se kroz dva faktora. Prvi faktor predstavlja uobičajeni postupak unošenja korisničkog imena i lozinke putem forme u okviru aplikacije. Kao dodatni odnosno drugi faktor predstavlja skeniranje QR (Quick Response) koda putem mobilne aplikacije uz pomoć koje se potvrđuje identitet korisnika [1].

U ovom radu će biti predstavljen sistem za generisanje QR koda kao i očitavanje pomenutog koda u cilju uspešnog omogućavanja koncepta dvostruke verifikacije identiteta korisnika, prikaz arhitekture sistema, odnosno od čega se sastoji ovaj sistem. S obzirom da je bezbednost veoma bitna kod ovakvih sistema, biće prikazano na koji način se generiše bezbedan QR kod i kako je jedino moguće pročitati vrednost tog koda.

2. TEORIJSKE OSNOVE

Za razumevanje rada potrebno je poznavanje pojmova kao što su QR kod, i dvostruka verifikacija korisničkog identiteta (2FA).

2.1. QR kod

Kao i kod razvoja mnogih tehnologija, QR kodovi su stvoreni iz nužde. QR kodovi su zapravo počeli kao bar kodovi sa svojom namenom u supermarketima.

Pre nego što su postojali bar kodovi, blagajnici su morali ručno da unose pojedinačne predmete, s obzirom da je to uzimalo dosta vremena poslodavci su shvatili da im je potreban način da sve prate [2].

QR kodovi predstavljaju dvodimenzionalne (matrice) kodove koji su razvijeni na način da u njih mogu da se jednostavno smeste informacije kao što su web adrese ili lokacije na mapi koji mobilni uređaj može brzo i jednostavno da skenira. Struktura QR koda je prikazana na slici u nastavku (Slika 2.1).



Slika 2.1. Struktura QR koda [3]

2.1.1. Generisanje QR koda

Proces kreiranja QR koda se svodi na 7 značajnih koraka: Analiza podataka, kodiranje podataka, kodiranje ispravke greške, strukturiranje završne poruke, smeštanje modula u matricu, maskiranje podataka i dodavanje informacija o formatu i verziji [4].

2.2. Dvostruka verifikacija korisničkog identiteta (2FA)

Dvostruka verifikacija korisničkog identiteta vezuje se za moderne tehnologije, takođe se naziva i više faktorska verifikacija korisničkog identiteta. Tradicionalni postupak verifikacije korisničkog identiteta pruža samo jedan faktor, obično nešto što korisnik može veoma lako da zapamti, lični brojevi (PIN) i lozinke su tipični primeri ove vrste verifikacije [5].

Načini upotrebe dvostruke verifikacije identiteta korisnika koji se često koriste su putem SMS poruke, mobilne aplikacije, upotrebe USB uređaja za dodatnu potvrdu identiteta korisnika.

Dvostruka verifikacija korisničkog identiteta predstavlja veći izazov za korisnika zbog toga što je potrebno da se koriste dva od tri faktora verifikacije identiteta (Slika 2.2).



Slika 2.2. Pristup servisu korišćenjem višestruke verifikacije identiteta korisnika [5]

NAPOMENA:

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

2.2.1. Google-ova implementacija dvostruke verifikacije identiteta

U cilju očuvanja bezbednosti naloga korisnika, Google podržava više opcija za dvostruku verifikaciju identiteta. Jedna opcija predstavljena je korišćenjem mobilne aplikacije Google Authenticator, dok se druga opcija ogleda u korišćenju servisa notifikacija Google-a [6].

2.2.2. Microsoft-ova implementacija dvostruke verifikacije identiteta

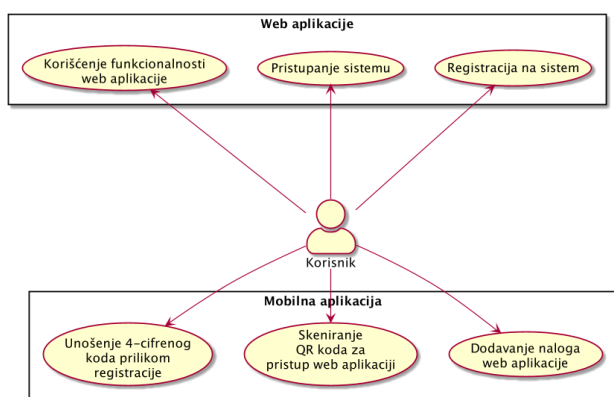
Microsoft omogućava svojim korisnicima pristup resursima na siguran način. Microsoft Authenticator se često koristi u većim kompanijama gde je potrebno postići dodatni nivo bezbednosti postavljanjem dvostruke verifikacije identiteta [7].

2.2.3. Okta Verify aplikacija

Kompanija Okta je razvila platformu za skladištenje aplikacija na jedan nalog, gde se jednostavno prilikom pristupanja sistemu korisniku omogućuje da pristupi bilo kojoj aplikaciji sa kojom je integracija izvršena. Okta Verify je aplikacija koja je podržana za iOS kao i za Android platformu [8].

3. MODEL SISTEMA

Broj funkcionalnosti koje korisnik može da izvrši korišćenjem ovog sistema nije velik, zbog toga što je glavni fokus sistema na bezbednom generisanju QR koda, kao i očitavanju. Dijagram korišćenja prikazan je u nastavku (Slika 3.1).

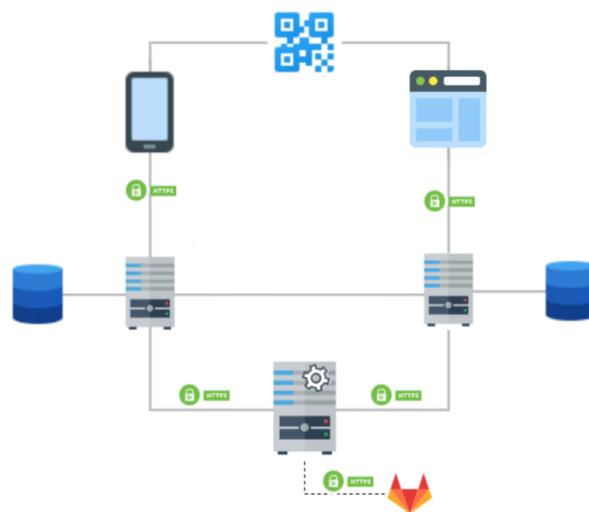


Slika 3.1. Dijagram korišćenja sistema

3.1. Arhitektura sistema

Glavne komponente koje čine ovakav sistem su: web aplikacija, mobilna aplikacija, konfiguracioni server kao i baze podataka. Pomenute komponente su prikazane na slici u nastavku (Slika 3.2).

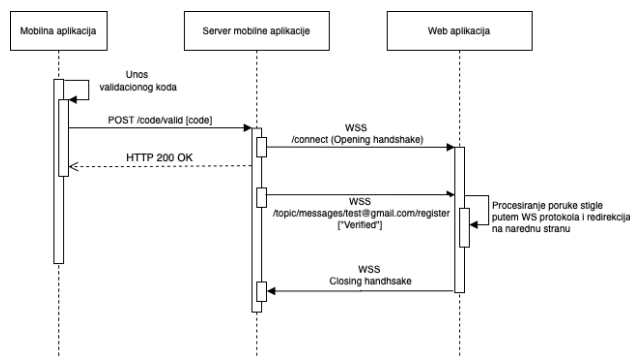
Korisnik web aplikaciju koristi za obavljanje funkcionalnosti koje pripadaju sklopu oblasti za koje je ona namenjena. Da bi korisnik uspeo uspešno da pristupi web aplikaciji potrebno da je potvrdi identitet putem mobilne aplikacije. Korisnik pomoću aplikacije skenira QR kod i na taj način verifikuje svoj identitet. Postojanje konfiguracionog servera ogleda se u smeštanju ključeva koji se koriste za kodiranje podataka prilikom kreiranja QR koda, takođe kao što je potrebno postojanje ključeva za kodiranje tako je i potrebno imati i ključeve za dekodiranje podataka koji se nalaze u QR kodu.



Slika 3.2. Komponente sistema

Baza podataka je zadužena za čuvanje podataka web aplikacije i mobilne aplikacije. Podaci koji se najčešće čuvaju su vezani za informacije o QR kodovima i korisnicima.

Prilikom uspostavljanja komunikacije između komponenti sistema koriste se dva načina uspostavljanja komunikacije. Komunikacija između klijenta i servera web aplikacija kao i mobilne aplikacije ostvaruje se putem HTTP protokola odnosno njegovo korišćenje je uspostavljeno putem RESTful servisa. Komunikacija između mobilne aplikacije i web aplikacije se ostvaruje korišćenjem Web socket-a protokola. Na slici u nastavku je prikazan primer komunikacije između mobilne i web aplikacije korišćenjem Web socket protokola.



Slika 3.3. Komunikacija putem Web socket protokola

4. IMPLEMENTACIJA

Da bi na bezbedan način sistem funkcionisao potrebno je ispravno izvršiti generisanje QR koda. Jedan od izazova prilikom izrade aplikacije bio je izabrati najbezbedniji način za generisanja QR koda. Potrebno je bilo odabrati da li će se kod generisati na strani web aplikacije odnosno na klijentskom delu korišćenjem biblioteka za generisanje slika koji će momentalno generisati korisniku QR kod ili je to bolje uraditi na strani web servera pa na neki način dostaviti to na klijentsku aplikaciju koja će samo prikazati podatke u vidu QR koda.

Otvoreni tekst QR koda u sistemu za verifikaciju identiteta predstavljen je strukturom JSON objekta sa poljima koja su prikazana u nastavku (slika 4.1)

```

{
    "email": "master@gmail.com",
    "generatedTime": "01-11-2020 20:30",
    "validTimeUntil": "01-11-2020 20:35"
}

```

Slika 4.1. QR kod otvorenog teksta

4.1.1. Šifrovanje podataka

Za šifrovanje podataka iz otvorenog teksta korišćen je AES algoritam. AES algoritam predstavlja bezbedni algoritam simetričnog šifrovanja, odnosno to znači da pripada grupi algoritama gde se sa istim ključem može izvršiti šifrovanje i dešifrovanje te je veoma bitan način na koji se vrši dostavljanje ključeva ovim stranama [9]

Kod implementiran u programskom jeziku Java prikazan je u nastavku (Listing 4.1).

```

public static String encryptData(String data) {
    try {
        IvParameterSpec iv = new IvParameterSpec(
            secretInitialFromProperties.
                getBytes(StandardCharsets.UTF_8));
        SecretKeySpec keySpec = new SecretKeySpec(
            secretKeyFromProperties.
                getBytes(StandardCharsets.UTF_8), "AES");

        Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
        cipher.init(Cipher.ENCRYPT_MODE, keySpec, iv);
        byte[] encrypted = cipher.doFinal(data.getBytes());
        return Base64.getEncoder().encodeToString(encrypted);
    } catch (Exception e) {
        log.error("Error while encrypting: " + e.toString());
    }
    return null;
}

```

Listing 4.1. Šifrovanje korišćenjem AES algoritma

4.1.2. Kreiranje sadržaja QR koda

Nakon što su podaci šifrovani i više se ne nalaze u obliku otvorenog teksta, na kraj niza podataka se dodaje tekst otvorenog oblika koji daje informaciju kojoj aplikaciji je namenjen QR kod tako da bi aplikacija znala da čita isključivo kodove koji su namenjeni njoj.

Generisanje QR koda se vrši korišćenjem Google-ove biblioteke ZXing [10] implementirane u Javi koja je namenjena za obradu slika u jednoj ili dve dimenzije za rad sa kodovima. Implementacija generisanja koda prikazana je u nastavku (Listing 4.2)

```

public String generateQRCode(String email) {
    QRCode valueOfQRCode = QRCode.builder()
        .email(email).build();
    valueOfQRCode.generateNewTime();
    String data = EncryptionAlgorithm
        .encryptData(valueOfQRCode.asString());
    data += ";applicationName: finance";
    BitMatrix matrix = null;
    ByteArrayOutputStream outputStream =
        new ByteArrayOutputStream();
    try {
        matrix = qrCodeWriter.encode(data,
            BarcodeFormat.QR_CODE, 350, 350);
        MatrixToImageWriter
            .writeToStream(matrix,
                "PNG", outputStream);
    } catch (WriterException e) {
        log.error("Error while
            generating QR code
            {} ", e.getMessage());
    }
    String base64bytes = Base64.getEncoder()
        .encodeToString(
            outputStream.toByteArray());
    return "data:image/png;base64," + base64bytes;
}

```

Listing 4.2. Generisanje QR koda

4.1.3. Prikazivanje QR koda

Nakon što web server izvrši generisanje QR koda i dostavi ga web aplikaciji, potrebno je da aplikacija prikaže korisniku QR kod kako bi on mogao da ga očitava, i time nastavi korišćenje sistema.

QR kod je ograničen vremenom, odnosno čitanje i verifikacija koda moguća je isključivo u određenom vremenskom opsegu koji se definiše prilikom generisanja kao jedan od parametara.

Zbog prethodno navedenog ograničenja, korisniku je potrebno na neki način predstaviti informaciju o vremenskom opsegu u kome može da izvrši očitavanje koda.

Ukoliko korisnik ne izvrši skeniranje u vremenu određenom za to, odnosno ako tajmer prikaže da je vreme za skeniranje QR koda isteklo, korisniku će biti prikazan taster za ponovno generisanje QR koda. Nakon što korisnik klikne na dugme za ponovno generisanje QR koda, web aplikacija će pozvati server da generiše kod koji se sada sastoji od istih podataka dok će jedino vreme važenja i generisanja biti drugačije.

4.1.4. Očitavanje i prikaz informacija QR koda

Generisani QR kod koji je prikazan korisniku putem web aplikacije u web pretraživaču sadrži informacije na osnovu kojih korisnik može da potvrdi svoj identitet. Da bi korisnik potvrdio svoj identitet potrebno je da poseduje mobilnu aplikaciju koja poseduje logiku za očitavanje podataka i verifikaciju identiteta korisnika.

S obzirom da su podaci iz QR koda zaštićeni šifrovanjem podataka, ukoliko bilo koji korisnik pokuša da očitava kod korišćenjem raznih aplikacija za skeniranje kodova, dobiće podatke ali mu oni ništa neće značiti zbog toga što su zaštićeni i predstavljaju tekst koji nema nikakvo značenje, odnosno neće znati kojem korisniku su namenjeni kao i druge informacije koje kod sadrži.

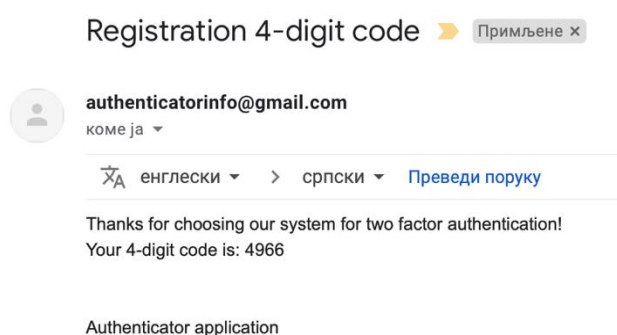
4.1.5. Komuniciranje mobilne i web aplikacije nakon uspešnog skeniranja koda

Nakon što korisnik putem mobilne aplikacije uspešno izvrši očitavanje QR koda, potrebno je obavestiti korisnika u web aplikaciji da je njegov QR kod uspešno očitav i da može nastaviti korišćenje web aplikacije.

S obzirom da mobilna aplikacija nema direktnu vezu sa web aplikacijom putem HTTP-a, komunikacija se ostvaruje korišćenjem WS (WebSocket) protokola. Komunikacija korišćenjem web socket-a ostvarena je između servera mobilne aplikacije i korisničkog dela web aplikacije.

4.1.6 Generisanje koda za potvrdu identiteta prilikom registracije

Prilikom procesa registracije na sistem posle korisnikovog skeniranja QR koda putem mobilne aplikacije potrebno je da se izvrši još jedan korak koji predstavlja dodatni nivo bezbednosti prilikom registracije, a to je unos jedinstvenog koda koji je prilikom registracije poslat korisniku putem mail-a. (primer mail-a koji stiže korisniku prilikom registracije, Slika 4.2)



Slika 4.2. Unos koda putem mobilne aplikacije

5. ZAKLJUČAK

Glavni fokus rada bio je opisati kompletan proces generisanja bezbednog QR koda, počev od podataka koje QR kod sadrži te opisa algoritma za šifrovanje pomenutih podataka, smeštanje šifrovanih podataka u QR, prikazivanja koda korisniku putem web aplikacije kao i međusobno komuniciranje između web aplikacije i mobilne aplikacije nakon uspešnog očitavanja koda. S obzirom da je jedinstvenost QR koda veoma bitna, objašnjeno je čemu služi jedinstveni kod prilikom registracije kao i na koji način se sprečava ponovno očitavanje već generisanog QR koda

Dodatna unapređenja u sistemu za dvostruku verifikaciju identiteta koja se mogu dodati su vezana najviše za distribuciju ključeva između web aplikacija i mobilne aplikacije, a takođe i jedna veoma bitna stvar je da se omogućiti postavljanje servera mobilne i web aplikacije na neki od Cloud provajdera (Google, Amazon, itd.) da bi se aplikaciji moglo pristupiti i van mreža na kojima se nalaze serveri. Nakon što bi se servisi izvršavali na platformama nekih od Cloud provajdera, bilo bi potrebno i omogućiti da se mobilne aplikacije mogu preuzeti preko nekih od poznatih IOS i Android prodavnica kao što su Google Play i App Store.

6. LITERATURA

- [1] Brian P. Sutton "The Effects of Technology in Society and Education,, 2013
https://digitalcommons.brockport.edu/cgi/viewcontent.cgi?article=1196&context=ehd_theses (pristupljeno u januaru 2021.)
- [2] Ako Muhammad Abdullah, Roza Hikmat Hama Aziz "Evaluating the Use of Quick Response (QR) Code at Sulaimani University Libraries" 2014
<https://www.researchgate.net/publication/270879583> (pristupljeno u januaru 2021.)

- [3] Zhongpai Gao, Guangtao Zhai and Chunjia Hu "The Invisible QR Code" 2015
<https://www.researchgate.net/publication/309350866> (pristupljeno u januaru 2021.)
- [4] Shuai Chen, "Evaluation of Two-Dimensional Codes for Digital Information Security in Physical Documents",
https://scholarworks.umass.edu/cgi/viewcontent.cgi?article=1235&context=masters_theses_2 (pristupljeno u januaru 2021.)
- [5] Jori Kymäläinen, "IMPLEMENTING TWO-FACTOR AUTHENTICATION", 2018
https://www.theseus.fi/bitstream/handle/10024/145670/Thesis_Jori_Kymalainen.pdf?sequence=1&isAllowed=y (pristupljeno u januaru 2021.)
- [6] Google, American company
<https://www.britannica.com/topic/Google-Inc> (pristupljeno u januaru 2021.)
- [7] About Microsoft, <https://www.microsoft.com/en-us/about> (pristupljeno u januaru 2021.)
- [8] Okta Verify,
<https://help.okta.com/en/prod/Content/Topics/Mobile/okta-verify-overview.htm> (pristupljeno u januaru 2021.)
- [9] Mahmoud A. eltatar, "Modified Advanced Encryption Standard Algorithm for Reliable Real-Time Communications" Dec/2017
<https://library.iugaza.edu.ps/thesis/123123.pdf> (pristupljeno u januaru 2021.)
- [10] ZXing image processing library, Javadoc
<https://zxing.github.io/zxing/apidocs/> (pristupljeno u januaru 2021.)

Kratka biografija:



David Vuletaš je rođen 25. oktobra 1995. godine u Zrenjaninu, Republika Srbija. Godine 2014. upisao je Fakultet Tehničkih nauka u Novom Sadu smer Softversko inženjerstvo i informacione tehnologije. 2018. godine diplomirao je na osnovnim studijama na Fakultetu Tehničkih nauka, iste godine potom upisuje Master studije na smeru Softversko inženjerstvo i informacione tehnologije.

MIKROSERVIS ZA EKSTRAKCIJU TEKSTA IZ WORD I PDF DOKUMENATA MICROSERVICE FOR TEXT EXTRACTION FROM WORD AND PDF DOCUMENTS

Dejan Bešić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – U ovom radu će biti opisano rešenje ekstrakcije teksta iz dokumenata u Word i PDF formatu. Pored same implementacije rešenja, diskutovaće se biblioteke koje su potrebne za ekstrakciju teksta, kao i za konverziju jednog formata dokumenta u drugi. Opisat će se struktura PDF dokumenta, zbog čega su u upotrebi kao i koji su problemi prilikom ekstraktovanja teksta. Problem ekstraktovanja teksta iz Word i PDF dokumenata se svodi na problem ekstraktovanja teksta iz PDF dokumenata.

Ključne reči: Mikroservis, ekstrakcija, tekst, Word, PDF, konverzija

Abstract – This paper will describe the solution of text extraction from documents in Word and PDF format. The solution will be created as a microservice architecture. In addition to the implementation of the solution itself, the libraries required for extraction as well as conversion will be discussed. This paper will describe the structure of PDF documents, why they are in use and what are possible disadvantages of text extraction from this type of documents. The problem of extracting text from PDF and Word documents, will be reduced to the problem of extracting text from PDF documents.

Keywords: Microservice, extraction, text, Word, PDF, conversion

1. UVOD

U ovom radu se prolazi kroz Servisno orijentisanu arhitekturu kao i Mikroservisnu arhitekturu. Zatim, nepohodno je objasniti šta su PDF dokumenti, kakva je struktura ovakvih dokumenata i kakvi problemi mogu da nastanu prilikom ekstrakcije teksta baš zbog takve strukture. Opisano je kako se problem ekstrakcije teksta iz Word i PDF dokumenata se prebacuje na problem ekstrakcije teksta iz samo PDF dokumenata. Prolazi se kroz biblioteke za konverziju datoteka i zbog čega neke od tih biblioteka ne zadovoljavaju potrebe ovog rada. Zatim, prolazi se kroz biblioteke za ekstrakciju teksta, za čime sledi implementacija rešenja. Na kraju rada se nalazi zaključak u kojem se spominju nedostaci ovog rešenja i moguća unapređenja.

2. MIKROSERVISI

Postoje mnogi dizajn šabloni za implementaciju i dizajniranje programskog rešenja, zavisno od toga koje

poslovne zahteve treba programsko rešenje da pokrije i koliko se planira unapred i ostavlja mogućnost za dalja unapređenja.

Ovo je neizbežno pitanje pri razvoju programskog rešenja, koliko god malo ili veliko po pitanju domena i opsega.

2.1. Servisno orijentisana arhitektura (SOA)

Osnovna jedinica SOA sistema jeste servis [2], dakle SOA se može definisati kao skup servisa, najčešće u distribuiranom računarskom sistemu, povezanih servisnim interfejsima koji komuniciraju pomoću zajedničkog komunikacionog kanala po predefinisanim pravilima. Ako odemo korak dalje, servisno orijentisana arhitektura predstavlja oblik organizacije integrisanog informacionog okruženja jednog sistema. Njega karakteriše ponuda i korišćenje njegovih distribuiranih poslovnih funkcija, grupisanih u različitim vidovima servisa. [1] Svaki servis u ovakvom sistemu sadrži u sebi kod (implementaciju, nezavisnu od konkretnog programskog jezika) i integraciju podataka neophodnih da se izvrši jedna kompletna, diskretna poslovna funkcija. Npr: kalkulacija mesečne rate za kredit, ili provera kreditnog statusa klijenta.

Dalje apstraktnu definiciju ovakve arhitekture čini skup principa pomoću kojih se definiše novi koncept informacionih sistema, od kojih su najznačajniji:

1. granulacija poslovnih funkcija,
2. obezbeđenje jedinstvenog pogleda na poslovanje kroz izgradnju arhitekture poslovnog sistema,
3. nezavisnost od tehnologije implementacije (fleksibilnost).

2.2. Mikroservisna arhitektura

Mikroservisna arhitektura je arhitektonski šablon koji je takoreći „nikao“ iz sveta domenski-vođenog dizajna (domain-driven design), kontinualnog isporučivanja softverskog rešenja, automatizacije kako platformi na koji je hostovan, tako i infrastrukture korišćena za hostovanje, i skalabilnih sistema. Mikroservisna arhitektura je zasnovana na SRP principu tj na principu da komponenta (u ovom slučaju servis) treba da ima jednu odgovornost

Ključna jedinica mikroservisne arhitekture leži u njenom nazivu: niz malih servisa, zvanih mikroservisi, tj. slobodnih programskih komponenti. Ovi servisi pružaju uslugu drugim servisima, sa kojima su povezani i sarađuju. Cilj jeste da servisi sadrže isključivo one funkcije koje su neophodne za izvršavanje jedne odgovornosti, i da se servisi kao takvi mogu nezavisno testirati i kombinovati.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Branko Milosavljević, red. prof.

Ovakva struktura servisa omogućava brži razvoj, i lakše delegiranje resursa tima ka održavanju i puštanju novijih verzija servisa u produkciju [3].

3. PDF DOKUMENTI

Kao najčešći tipovi dokumenata u sadašnjoj upotrebi računara, nije na odmetu reći da su PDF i Word dokumenta prva asocijacija kada korisnik treba da podeli neke informacije ili prebaci ručno napisane formule u digitalan ekvivalent. U ovom poglavlju se zalazi u strukturu PDF dokumenta, pošto su oba zasnovana ispod haube na XML notaciji.

3.1. Struktura

PDF je format dokumenta koji omogućava prezentovanje dokumenata sa tekstualnim sadržajem, multimedijalnim elementima, slikama i slično, razvije od strane Adobe korporacije, 1993. godine. Ovaj format je napravljen da bude agnostičan u odnosu na konkretne operativne sisteme, aplikacije i hardver. PDF fajl sadrži opis konkretnog rasporeda sadržaja dokumenta, kao i opis svakog elementa neophodnog za njegovo prikazivanje, bilo to fontovi, vektorske slike ili čist tekst.

4. PROBLEM EKSTRAKCIJE TEKSTA IZ PDF DOKUMENATA

Kada se posmatra ekstrakcija teksta iz dokumenta u PDF formatu, često se postavlja pitanje šta je tu zaista problem jer je tekst lako vidljiv ljudskom oku. Rad sa dokumentima u PDF formatu je težak zbog same ekstremne fleksibilnosti PDF-a. Najveći problem je što PDF format nikada nije stvarno dizajniran kao format unosa podataka, već je dizajniran kao izlazni format koji daje fino zrnastu kontrolu nad rezultirajućim dokumentom. U osnovi, PDF format se sastoji od niza instrukcija koje opisuju kako se crta na stranici, što znači da se tekstualni podaci ne čuvaju kao pasusi ili reči, već kao posebni karakteri sa instrukcijama gde će biti nacrtani na stranici. Kao rezultat, većina semantike sadržaja se gubi kada je tekst konvertovan u dokument u PDF formatu.

Mogući uzroci problema u ekstrakciji teksta iz dokumenata u PDF formatu mogu biti:

1. Zaštita od čitanja
2. Karakteri van stranice
3. Mali i nevidljivi karakteri na stranici
4. Veliki broj razmaka
5. Mali broj razmaka
6. Detekcija pasusa i reči
7. Redosled teksta i pasusa
8. Slike

5. BIBLIOTEKE ZA KONVERZIJU DOKUMENATA U WORD FORMATU U DOKUMENTE U PDF FORMATU

Nekada PDF dokument sadrži dodatne razmake između karaktera. To se uglavnom dešava zbog Kerning-a. Kerning je proces prilagođavanja razmaka između karaktera. Jedan od načina za rekonstruisanje ovakvog teksta je tehnikom optičkog prepoznavanja karaktera (OCR).

5.1 Documents4j

Documents4j je Java biblioteka za konverziju dokumenata u drugi format dokumenta. To se postiže davanjem ovlašćenja konverzije bilo kojoj izvornoj aplikaciji koja razume konverziju date datoteke u željeni ciljni format. Dokument4j dolazi sa adaptacijama za MS Word i MS Excel za Windows operativni sistem, što omogućava, na primer, konverziju dokumenta sa .docx ekstenzijom u PDF dokument bez uobičajenih izobličenja u rezultujućem dokumentu koja se često primećuju kod konverzija izvršenih pomoću proizvoda koji nisu Microsoftovi. Drugim rečima, ova biblioteka je zavisna od softvera MS Word i MS Excel. U slučaju rešenja opisanog u ovom radu, postoji nekoliko mana ove biblioteke koje sprečavaju njeno korišćenje:

1. Da bi funkcionisala, potrebno je posedovati MS Word i MS Excel u okruženju
2. Mikroservis u ovom radu će biti pokrenut na Linux operativnom sistemu, samim tim dodavanje MS Word i MS Excel softvera, nije tako jednostavno
3. MS Word i MS Excel su licencirani proizvodi, nisu besplatni
4. Ova biblioteka ne podržava konverziju dokumenata sa .doc ekstenzijom (obe ekstenzije, .doc i .docx, pripadaju Word dokumentima, gde .doc predstavlja stariji format)

5.2 Apache POI

Apache POI je projekat vođen od strane neprofitabilne kompanije pod nazivom „Apache Software Foundation”. Apache POI pruža biblioteke u Java programskom jeziku, za čitanje i pisanje datoteka u Microsoft Office formatima, kao što su Word, PowerPoint i Excel [4]. Poredeći sa Documents4j bibliotekom, Apache POI podržava konverziju za dokumente sa .doc i .docx ekstenzijom. Nije potreban prateći softver iz grupe Microsoft Office softvera. Takođe, ova biblioteka je nezavisna od platforme na kojoj se nalazi.

Međutim, ova biblioteka takođe poseduje mane zbog kojih u rešenju ovog rada, nije najprikladnija za korišćenje. Prva mana je što implementacija rešenja nije ista za dokumente sa .doc i .docx ekstenzijama. Implementacija za .docx je kratka i bez većih komplikacija. Problem nastaje prilikom implementacije rešenja za dokumente sa .doc ekstenzijom. Konvertovanje dokumenata sa .doc ekstenzijom se vrši ručno, tako što se ekstrahuje tekst, stilovi, razmaci, margine, fontovi, itd. i zatim se pomoću ekstrahovanih osobina dokumenta, kreira PDF dokument. Implementacija ovakvog rešenja već postoji međutim ni jedno od rešenja ne daje dovoljno dobre rezultate i vrlo često se dobije neodgovarajući PDF dokument. Za rešenje u ovom radu, veoma je bitno da PDF dokument odgovara Word dokumentu, i iz tog razloga ova biblioteka nije dovoljno precizna.

5.3 LibreOffice

LibreOffice je nastao na bazi popularnog slobodnog kancelarijskog paketa OpenOffice.org iz želje većine učesnika da se dalji razvoj izmesti u okrilje nezavisne neprofitne fondacije. LibreOffice je besplatan softverski

paket za Windows, Mac OS i Linux sisteme koji radi na uobičajenim računarima. Ova biblioteka, kao i Documents4j, zavisi od eksternog LibreOffice softvera koji mora da se nalazi u okruženju. Prednosti LibreOffice softvera u odnosu na Microsoft Office grupu softvera su:

1. LibreOffice je besplatan
2. Podržan je na Linux operativnom sistemu
3. Podržava konverziju dokumenata sa .doc i .docx ekstenzijama

LibreOffice biblioteka, iako se pokazala kao najbolja biblioteka za potrebe rešenja u ovom radu, postoji jedna mana koja je zajednička kod svih softvera. Implementacija se menja sa verzijama biblioteke i vrlo je moguće da će u budućnosti biti potrebe za izmenama u implementaciji ukoliko dođe do promene verzije biblioteke.

5.4 Gotenberg

Gotenberg je API bez stanja za konverziju HTML, Markdown i Office dokumenata u dokumente u PDF formatu [5]. Gotenberg podržava konverziju .doc i .docx dokumenata koristeći LibreOffice biblioteku. Gotenberg predstavlja rešenje problema sa verzijama u LibreOffice biblioteci, jer na ovaj način, taj problem se problem verzija prebacuje na Gotenberg. Gotenberg je, takođe, podržan na bilo kojoj platformi. Implementacija konverzije Word dokumenata u PDF dokumente se prebacuje na ovu biblioteku.

6. BIBLIOTEKE ZA EKSTRAKCIJU TEKSTA IZ DOKUMENATA U WORD FORMATU

U prethodnoj sekciji je opisan problem konverzije iz dokumenata u Word formatu u dokumente u PDF formatu, dok u ovoj sekciji će se opisati biblioteke za ekstrakciju teksta kao i zašto je potrebna konverzija dokumenata.

6.1 Apache POI

U prethodnoj sekciji su opisane mane Apache POI biblioteke zbog čega se ne koristi u ovom radu za konverziju iz dokumenata u Word formatu u dokumente u PDF formatu. Konverzija dokumenata nije jedino svojstvo Apache POI biblioteke. Ova biblioteka takođe podržava ekstrakciju teksta iz dokumenata u Word formatu. Kao rezultat ekstrakcije se dobija celoukupni tekst iz datog dokumenta. Međutim, za potrebe rešenja u ovom radu, to nije dovoljno. Potrebe ovog rada je da se ekstrahuje tekst tačno na kojoj se stranici nalazi. Celoukupni tekst ne sadrži informaciju o stranicama. Parsiranjem teksta po količini novih redova je moguće, međutim nije u svakom slučaju tačno i dobijaju se nedovoljno precizni podaci.

6.2 Apache PDFBox

Apache PDFBox je Java PDF biblioteka otvorenog koda za rad sa PDF dokumentima. Ovaj projekat omogućava stvaranje novih PDF dokumenata, manipulaciju postojećim dokumentima i mogućnost izdvajanja sadržaja iz dokumenata. [6]

Neke od glavnih karakteristika ove biblioteke su:

- Ekstrakcija teksta iz PDF dokumenata

- Spajanje PDF dokumenata
- Enkripcija i dekrepcija PDF dokumenata
- Lucene pretraživač
- Popunjavanje podataka obrasca FDF (Forms data format) i XFDF (XML Forms data format)
- Kreiranje PDF dokumenata od tekstualnih datoteka
- Kreiranje slika od stranica u PDF dokumentu
- Štampanje PDF dokumenata

Za razliku od biblioteka za ekstrakciju teksta iz Word dokumenata ova biblioteka podržava ekstrakciju teksta po stranici i iz tog razloga, Apache PDFBox će se koristiti u implementaciji rešenja problema ekstrakcije teksta.

7. IMPLEMENTACIJA REŠENJA

Implementacija mikroservisa urađena je kao Spring Boot aplikacija, u Java programskom jeziku. Portabilnost i nezavisnost servisa je rešeno kreiranjem mikroservisa kao Docker slike. Komunikacija sa ovim mikroservisom se vrši putem HTTP-a. Aplikacija sadrži jednu krajnju tačku preko koje je dostupna celom sistemu. Krajnja tačka se nalazi na relativnoj putanji pod nazivom /extract. Ova kranja tačka prima jedan parametar pod nazivom file. Parametar file je tipa MultipartFile što označava da ova metoda može da primi bilo koju datoteku. Zbog ovog parametra, metoda HTTP zahteva je POST. Ukoliko je file parametar izostavljen, rezultat ove metode će biti greška sa HTTP status kodom 400 i porukom koja govori da je file polje obavezno. Ovaj mikroservis treba da ekstrahuje tekst iz dokumenata u Word i PDF pa se iz tog razloga dokument filtrira koristeći ekstenziju. Ukoliko se naziv dokumenta završava sa doc ili docx, znači da je dokument u Word formatu. Ukoliko je naziv dokumenta završava sa pdf, znači da je dokument u PDF formatu. Ukoliko nije ni jedan ova dva formata, rezultat metode će biti greška sa HTTP status kodom 400 i porukom koja govori da dokument mora biti u Word ili PDF formatu. . U zavisnosti od formata, poziva se odgovarajući servis. Ukoliko je dokument u Word formatu, poziva se metoda extract iz WordExtractionService klase. Ukoliko je dokument u PDF formatu, poziva se metoda extract iz PdfExtractionService klase.

Rezultat, nakon uspešne ekstrakcije teksta, je objekat opisan klasom PDF. Atributi ove klase su:

- author – autor dokumenta
- title – naslov dokumenta
- subject – tema dokumenta
- keywords – ključne reči koje opisuju dokument
- numberOfPages – broj stranica
- creationDate – datum kreiranja kao tekstualni podatak u ISO 8601 formatu
- modificationDate – datum poslednje izmene kao tekstualni podatak u ISO 8601 formatu
- pages – lista stranica koje su opisane klasom Page

Pored atributa, ova klasa poseduje pomoćne metode radi lakšeg rukovanja atributima, kao što su:

- addPage – služi za dodavanje stranice

- `setCreationDate` – služi za postavljanje atributa `creationDate` u ISO 8601 format
- `setModificationDate` – služi za postavljanje atributa `modificationDate` u ISO 8601 formatu

`WordExtractionService` je klasa koja služi kao servis za ekstrakciju teksta iz dokumenata u Word formatu. Ova klasa sadrži jednu glavnu metodu, `extract` koja je vidljiva ostatku aplikacije. `Extract` metoda služi za ekstrakciju teksta iz dokumenata u Word formatu kao i dodavanje podataka o samom dokumentu. Ulazni parametar je dokument, dok je rezultat kreiran PDF objekat. Da bi se tekst ekstraktovao, potrebno je konvertovati dokument iz Word formata u dokument u PDF formatu. Konverzija se vrši pomoću metode `wordToPDF` iz `ConvertService` klase. Posle konverzije, sledi ekstrakcija teksta i novonastalog PDF dokumenta. Ekstrakcija teksta se vrši pomoću metode `addText` iz `PdfExtractionService` klase. Na kraju, preostaje da se dodaju podaci o dokumentu.

`ConvertService` klasa služi za konverziju dokumenata u Word formatu u dokumente u PDF format. Ova klasa poseduje jednu glavnu metodu, `wordToPDF`. Ova metoda koristi Gotenberg API koji će konvertovati Word dokument u PDF dokument. Gotenberg se pokreće u istom okruženju kao i mikroservis. Koristi se krajnja tačka na putanji `/convert/office`. Komunikacija između mikroservisa i Gotenberg API-a se odvija putem HTTP-a. Šalje se HTTP zahtev sa POST metodom. Bitno je naglasiti da `ContentType` u zaglavlju HTTP zahteva mora da bude `multipart/form-data`. Telo zahteva treba da sadrži dva polja, `files` i `waitTimeout`, gde je `files` niz dokumenata koji treba da se konvertuju a `waitTimeout` maksimalna dužina čekanja konverzije dokumenata. Rezultat ove konverzije jeste niz bajtova koji predstavljaju dokument u PDF formatu.

`PdfExtractionService` je klasa koja služi za ekstrakciju teksta i meta podataka iz dokumenata u PDF formatu.

Dodavanje teksta se vrši pomoću Apache PDFBox biblioteke, koja služi za izvlačenje teksta po stranici u dokumentu. Kao rezultat rada ove biblioteke, dobija se lista stranica u `PDDocument` formatu. Iz svake stranice je potrebno ekstraktovati tekst posebno. Takav ekstraktovani tekst je potrebno srediti tako što se uklanjaju suvišni razmaci između teksta. Svaku stranicu je potrebno numerisati i na kraju dodati u PDF objekat

Za kreiranje i pokretanje Docker slike su bitna dva fajla, `Dockerfile` i `docker-compose.yml`. `Dockerfile` se koristi za kreiranje okruženja koje je potrebno da bi mikroservis funkcionisao. Da bi ovaj mikroservis mogao da funkcioniše, potreban je Java Development Kit. Zatim, potrebno je prebaciti već u napred pripremljene `.jar` datoteke koje predstavljaju ovaj mikroservis. `Docker-compose.yml` služi da bi povezali Gotenberg API sa mikroservisom za ekstrakciju teksta. U ovom fajlu je bitno definisati na kojem portu će biti dostupan mikroservis i Gotenberg API. Takođe je bitno da se definiše da mikroservis za ekstrakciju teksta zavisi od Gotenberg API-a. Ukoliko pokretanje Gotenberg API ne uspe da se pokrene, ceo proces će biti zaustavljen. Za Gotenberg je bitno da se navede tačan naziv i verzija kao i port na kojem će da „sluša“.

8. ZAKLJUČAK

Ovaj rad objašnjava problem ekstrakcije teksta iz dokumenata u Word i PDF formatu. Diskutuje koja od biblioteka najbolje odgovara problemu konverzije dokumenata iz dokumenata u Word formatu u dokumente u PDF formatu. Opisana je struktura mikroservisa, razlika između SOA i mikroservisa kao i prednosti ovakve arhitekture sistema. Dat je pregled strukture PDF dokumenata, zašto predstavlja jedan od najzastupljenijih formata za pregled dokumenata. Objašnjeno je koji problemi postoje korišćenjem PDF dokumenata i kako se rešavaju ti problemi u praksi, iako mnogi od njih nemaju idealno rešenje. Takođe, objašnjeno je kako se problem Word i PDF dokumenata, svelo na problem ekstrakcije teksta samo iz PDF dokumenata.

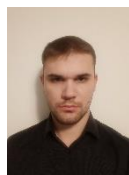
Kao poboljšanje ovog rada može biti povećani broj tipova dokumenata koje ovaj mikroservis podržava. Kao jedno rešenje, može biti takođe Gotenberg API. Ovaj API podržava veliki broj dokumenata, pa kao rešenje može biti konverzija tih dokumenata u PDF format, pa zatim ekstraktovati tekst iz tih dokumenata. Takođe, kao jedan vid unapređenja može biti bolji način detekcije vrsta dokumenta. Većina dokumenata ima „potpis“ po kojem se prepoznaje da li dati dokument odgovara datom formatu. Na taj način se može smanjiti broj grešaka koji se mogu dogoditi unutar mikroservisa i učiniti ceo sistem robusnijim. Kao još jedan vid unapređenja bi moglo da bude ekstrakcija multimedijalnog sadržaja iz dokumenata.

Cilj ovog rada je bio da se kreira mikroservis koji će za svaku stranicu PDF ili Word dokumenta, posebno ekstraktovati tekst i takav ekstraktovani tekst numerisati stranicom na kojoj se nalazi.

9. LITERATURA

- [1] Servisno orijentisana arhitektura i integrisanje poslovnih aplikacija. Preuzeto sa https://www2.masfak.ni.ac.rs/uploads/articles/www2_5_soa_skraceno.pdf
- [2] Servisno-orijentisana arhitektura, IBM, <https://www.ibm.com/cloud/learn/soa>
- [3] Microservice Architecture, <https://microservices.io/patterns/microservices.html>
- [4] Apache POI, https://en.wikipedia.org/wiki/Apache_POI
- [5] Gotenberg, <https://thecodingmachine.github.io/gotenberg>
- [6] Apache PDFBox, https://en.wikipedia.org/wiki/Apache_PDFBox

Kratka biografija:



Dejan Bešić rođen je u Novom Sadu 1995. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika odbranio je 2021.god. kontakt: dejanbesic31@gmail.com

MIKROSERVIS ZA PRETRAGU REČI U PROJEKTU REČNIKA SRPSKOG JEZIKA SEARCH MICROSERVICE FOR SERBIAN LANGUAGE DICTIONARY PROJECT

Tanja Indiđ, *Fakultet tehničkih nauka, Novi Sad*

Oblast – SOFTVERSKO INŽENJERSTVO

Kratak sadržaj – U radu je opisan mikroservis za indeksiranje i pretragu tekstualnog sadržaja, karakteristike ElasticSearch alata, programski jezik Python i okruženje Django. Opisani su razvojni okviri neophodni za implementaciju rešenja. Predstavljeno je implementirano rešenje, opisane su funkcionalnosti koje su osnove rada aplikacije i prikazano je dobijeno rešenje.

Ključne reči: mikroservis, indeksiranje, pretraga, ElasticSearch, Python programski jezik, Django razvojni okvir

Abstract – The thesis describes microservice for indexing and data retrieval, ElasticSearch characteristics, the programming language Python and Django framework. Frameworks and tools needed to for implementation into the solution are described. The implemented solution is presented, the functionalities that are the carriers of the application work are shown and screenshots of developed application are shown.

Keywords: microservice, indexing, search, ElasticSearch, Python programming language, Django framework

1. UVOD

Ovaj rad se bazira na temi razvoja i korišćenja digitalnog rečnika srpskog jezika, uz osvrt na dosadašnje, klasično korišćenje putem knjiga. Kako je čest je slučaj da se određene reči izgovaraju i pišu pogrešno ili one koje se retko koriste padnu u zaborav, a savremen čovek je navikao da mu je sve dostupno na par klikova na računaru, javila se i potreba za kreiranjem online srpskog rečnika - rečnika Matice srpske.

Zadatak ovog rada jeste projektovanje web aplikacije za indeksiranje i pretragu reči u cilju kreiranja i pretrage korpusa za srpski jezik. Rešenje je potrebno implementirati u Python¹ programskom jeziku uz upotrebu Django² razvojnog okvira a uz oslonac na ElasticSearch³ alat, koji implementira funkcije potrebne za indeksiranje i pretragu dokumenata.

2. TEORIJSKE OSNOVE

U ovom poglavlju se nalazi opis teorijskih osnova na kojima je zasnovan ovaj rad.

¹ <https://www.python.org/>

² <https://www.djangoproject.com/>

³ <https://www.elastic.co/>

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Branko Milosavljević, red. prof.

2.1 Rečnik

Sa razvojem ljudske civilizacije, napredovala je i komunikacija i načini ostavljanja traga i zapisivanja istorije čovečanstva. Nakon crteža javljaju se prva slova, reči i jezici. Tako nastaju i prvi rečnici i dugo vremena su se koristili u svom izvornom, pisanom obliku. Međutim, sa napretkom tehnologije, sve više podataka biva dostupno a i traženo u digitalnom formatu. Iz tog razloga postoji potreba za razvojem online srpskog rečnika koji bi bio dostupniji i lakše pretraživ.

2.1.1 Razvoj računara i povećanje obima podataka

Vremenom računari postaju sve dostupniji, njihova primena je sve raznovrsnija, a tehnološkim napretkom brzina procesiranja podataka i količina sa kojom mogu da barataju raste. Samim tim, rano se javlja potreba za njihovom organizacijom, klasifikacijom i efikasnom pretragom, pojavljuju se brojni algoritmi za smeštanje i brzo pronalaženje podataka [1]. Uz to, kako se spektar krajnjih korisnika širi, sve više se radi na razvoju korisničkih interfejsa koji bi pojednostavili i ubrzali rad na računarima. Uporedo sa razvojem računara, radi se i na razvoju infrastrukture koja bi povezala više računara putem telekomunikacija. Rezultat ovih napora bio je ARPANET [2], prva računarska mreža koja je trebala da obezbedi komunikaciju vojnih laboratorija, vladinih biroa i univerziteta. Originalni ARPANET je vremenom prerastao u Internet, što je dovelo do još većeg rasta količine dostupnih podataka, međutim da bi iko iskoristio bilo koji podatak potrebno je prvo da može da ga pronađe.

2.1.2 Alati za pretragu

Jedan od prvih sistema za pretragu bio je Archie [3], koji je služio kao alat za indeksiranje FTP (File Transfer Protocol) fajlova. U odnosu na današnje pretraživače, može se reći da je Archie bio primitivan alat, budući da nije podržavao proizvoljne korisničke upite, niti je indeksirao sadržaje fajlova, te je pretraga bila ograničena samo na tačne nazive. Indeksiranje sadržaja prvi put je podržao Gopher [3] protokol, a ideja iza ovog sistema bila je laka upotreba, hijerarhijski prikaz i kretanje kroz sistem, jednostavna sintaksa i podrška za pretrage. Bio je zamišljen kao globalni fajl sistem prikazan kao hijerarhija hiperlinkova, međutim već od 1993. godine umesto njega počinje da se koristi World Wide Web [2] – web.

Kako se funkcionalnost pretrage i indeksiranja tražila i u manjim i jednostavnijim okruženjima, dolazi do razvoja Lucene [5] biblioteke a zatim i ElasticSearch alata koji postaje globalno korišćen kao dodatak u širokom spektru sistema kojima je potrebna ovakva funkcionalnost. Omogućava samostalno korišćenje a do skoro i kao embedded deo servisa.

3. ALATI ZA RAZVOJ

ElasticSearch je jedan od najpoznatijih distribuiranih alata za pretragu nastao 2010. godine, a njegov autor je Shay Banon [4]. Zasniva se na *Lucene* biblioteci, napisan je u *Java* programskom jeziku i pruža mogućnost rada sa tekstualnim, numeričkim, geospacijalnim, strukturiranim i nestrukturiranim podacima [5].

Podrazumeva rad sa dokumentima kao osnovnom vrstom organizacije podataka koji se reprezentuju u JSON formatu. Podržava HTTP veb interfejs putem kojeg je moguće vršiti pretragu, dodavanje i manipulaciju nad dokumentima. Pre samog indeksiranja, potrebno je definisati model dokumenta i način na koji se njegova polja čuvaju i indeksiraju.

Neretko se dešava da podaci sa kojima se radi budu izvan trenutnog sistema, te se ne može uticati na njihov format. Zbog toga je postupak transformacije podataka u oblik koji je pogodan za indeksiranje i pretragu neophodan i ovaj korak je ključan za dobar rad sistema a enkapsuliran je u obliku dodatka – *plugin*. Ovakav dodatak se naziva *analyzer* i sastoji se iz filtera karaktera, tokenizera i filtera tokena [5].

Filtriranje karaktera podrazumeva uklanjanje, dodavanje i izmenu pojedinačnih karaktera u dokumentu, koristi se pri parsiranju HTML fajlova za uklanjanje oznaka za elemente, transformaciju ili brisanje određenih karaktera, uklanjanje akcenata, prebacivanje iz jednog pisma u drugo.

Tokenizer preuzima transformisani niz karaktera i služi za razdvajanje podataka u tokene, na primer tekst može razdvojiti na niz reči na osnovu razmaka, znakova interpunkcije i slično. Takođe, zadužen je za pamćenje redosleda ili pozicije tokena, kao i početak i kraj originalne reči u nizu karaktera.

Filter tokena služi za krajnju transformaciju nakon razdvajanja teksta na niz tokena. Može prebaciti sve elemente ovog niza u mala slova ili procesirati podatke na osnovu odabranog jezika. Moguće je definisati listu stop reči za određeni jezik, tj. listu često korišćenih reči koje nije potrebno indeksirati i treba izbaciti iz daljeg procesiranja.

3.1 Python programski jezik

Python je interpretirani, objektno orijentisani programski jezik visokog nivoa nastao 1991. godine. Podržava imperativni, objektno-orijentisani i funkcionalni stil programiranja. Svaki podatak predstavljen je kao objekat ili kao relacija između objekata. Budući da koristi strukturu podataka visokog nivoa, u kombinaciji sa dinamičkim kucanjem i dinamičkim vezivanjem, *Python* se često koristi za brz razvoj aplikacija. Ovaj programski jezik podržava dodatne module i pakete, omogućava modularnost i ponovnu iskorišćenost koda. Moguće ga je interpretirati i kompajlirati.

3.2 Django razvojni okvir

Django je razvojni okvir koji omogućava lako razvijanje naprednih i dinamičnih web aplikacija uz korišćenje *Python*-a. Ono što izdvaja ovu platformu je podrška za autorizaciju i autentifikaciju korisnika koja se lako može prilagoditi svakom tipu sistema. Veoma je fleksibilan, omogućava jednostavan i intuitivan rad sa bazama podataka.

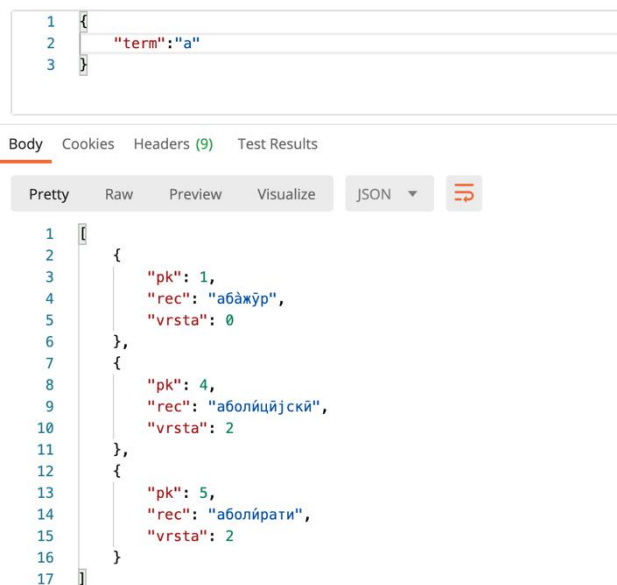
Podržava MTV⁴ arhitekturu (*model-template-views*) koja omogućava deljenje implementacije u više slojeva. Ukoliko posmatramo kako funkcioniše srž ovog alata, može se reći da zapravo podržava standardnu MVC⁵ arhitekturu (*model-view-controller*): koristi svoj mapper za translaciju između definisanih modela podataka i relacionih modela, zatim ima svoj sistem koji podržava korišćenje veb šablona za kreiranje prikaza i takođe sadrži deo za komunikaciju sa klijentima koji je baziran na putanjama – URL (*Uniform resource locator*).

4. IMPLEMENTACIJA REŠENJA

Projekat za Rečnik srpskog jezika je osmišljen kao podrška za razvoj *online* rečnika Maticе srpske. Ovaj projekat se razvija na Katedri za informatiku Fakulteta tehničkih nauka u Novom Sadu. Poput drugih rečnika, ima namenu da omogući krajnjim korisnicima pretragu reči i na ćirilici i latinici, ali je namenjen i za leksikografe koji kreiraju sadržaj samog rečnika. Projektovan je po ugledu na mikroservisnu arhitekturu i pored ovog servisa postoje još i servis za digitalizaciju dokumenata, servis za ekstrakciju sadržaja, klijentska aplikacija za unos i pretragu rećnićkih odrednica i servis za pripremu rećnika za štampu.

Programski jezik korišćen za implementaciju aplikacije za indeksiranje i pretragu je *Python*, a razvojno okruženje je *PyCharm*. Rešenje je implementirano korišćenjem *Django* razvojnog okvira uz oslonac na *ElasticSearch* kao alat za indeksiranje i pretragu. Komunikacija sa u okviru sistema se odigrava preko REST servisa, a *Django* razvojni okvir se koristi za mapiranje JSON opisa na odgovarajuće klase.

Za pretragu rećnićkih odrednica bilo je potrebno implementirati *auto-complete* funkcionalnost i to uz podršku za oba pisma. Od podataka koji se unose pri kreiranju odrednice izdvojeni su primarni ključ, reć i vrsta reći, na osnovu kojih se mogu iz baze izvući dodatne informacije.



```
1 {
2   "term": "a"
3 }
```

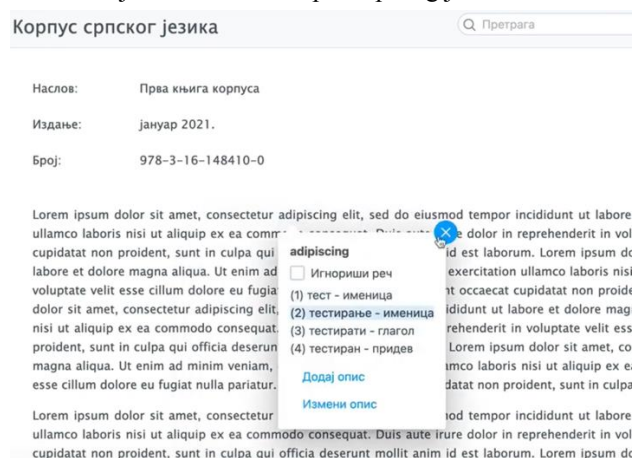
```
1 [
2   {
3     "pk": 1,
4     "rec": "абѡжѡр",
5     "vrsta": 0
6   },
7   {
8     "pk": 4,
9     "rec": "аболѡцѡскѡ",
10    "vrsta": 2
11  },
12  {
13    "pk": 5,
14    "rec": "аболѡрати",
15    "vrsta": 2
16  }
17 ]
```

Slika 1. Prikaz rezultata pretrage odrednica

⁴ <https://djangobook.com/mdj2-django-structure/>

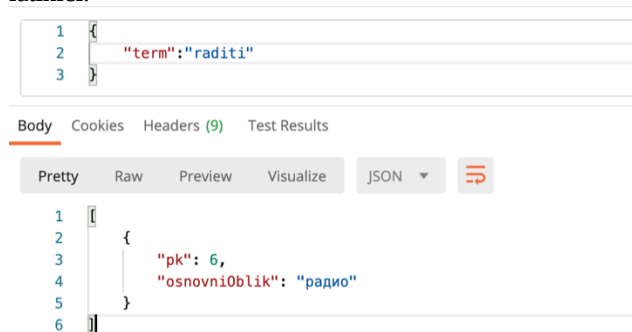
⁵ <https://en.wikipedia.org/wiki/Model-view-controller/>

Druga funkcionalnost predstavlja pretragu već postojećih anotiranih reči. Naime, leksikografi u procesu anotiranja dokumenta prolaze kroz njegov sadržaj i potrebno je omogućiti opciju da se svaka reč iz sadržaja definiše kao nova anotirana reč ili da se poveže sa postojećom. Na taj način dobija se anotirani korpus srpskog jezika.



Slika 2. Prikaz forme za anotaciju reči

Za svaku anotiranu reč bilo je potrebno indeksirati sve njene oblike i takođe podržati pretragu i na ćirilici i na latinici.



Slika 3. Prikaz rezultata pretrage za anotiranu reč

5. ZAKLJUČAK

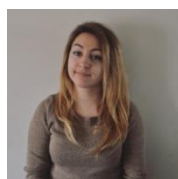
U radu je opisan servis za indeksiranje i pretragu dokumenata, opisan je programski jezik *Python* i njegov razvojni okvir *Django*. Predstavljen je *ElasticSearch* kao alat na koji se ovaj servis oslanja. Prezentovano je rešenje, objašnjene najznačajnije funkcionalnosti i dat je primer korišćenja aplikacije. Kreirano rešenje može biti korisno za Rečnik srpskog jezika, ali i za druge primene, uz određene modifikacije. Takođe, korišćenje *Django* razvojnog okvira, donelo je velike prednosti i olakšanja u implementaciji sistema.

Rad u ovoj oblasti ukazao je na veliki broj mesta i prilika za proširenje. Kreiranje *web* servisa, autentifikacija i autorizacija korisnika mogu biti odrađeni na više načina, ne samo upotrebom *Django*-a. Takođe, indeksiranje i pretraga ne moraju nužno biti izvršeni korišćenjem *ElasticSearch* alata, ali same transformacije podataka i upita pomoću posebno napravljenog dodatka bi morale biti drugačije implementirane kako bi se podržalo i latinično i ćirilčno pismo.

6. LITERATURA

- [1] Martin Kleppmann, *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable and Maintainable Systems*, 2017, ISBN 9781491903070
- [2] Internet, ARPANET. Preuzeto sa <https://en.wikipedia.org/wiki/Internet>
- [3] Search Engine. Preuzeto sa https://en.wikipedia.org/wiki/Search_engine
- [4] ElasticSearch. Preuzeto sa <https://en.wikipedia.org/wiki/Elasticsearch>
- [5] ElasticSearch dokumentacija. Preuzeto sa <https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>

Kratka biografija:



Tanja Indić rođena je 1995. godine u Arandelovcu. Završila je gimnaziju „Miloš Savković“ u Arandelovcu. 2014. godine, upisuje se na studije na Fakultetu tehničkih nauka, odsek Računarstvo i automatika. Za dalje usmerenje u okviru pomenutog odseka, odlučuje se za smer Primenjene računarske nauke i informatika. Master studije je slušala na istom odseku i smeru, a za podsmjer je odabrala Elektronsko poslovanje.

RAZVOJ APLIKACIJE ZA AŽURIRANJE MODELA ELEKTROENERGETSKE MREŽE U OKVIRU CLOUD OKRUŽENJA**DEVELOPMENT OF AN APPLICATION FOR UPDATING THE POWER NETWORK MODEL WITHIN THE CLOUD ENVIRONMENT**

Bogdan Kovačev, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Ovaj rad bavi se kreiranjem aplikacije za ažuriranje modela elektroenergetske mreže u Cloud okruženju korišćenjem Azure Service Fabric platformu. U radu su takođe izvršena i testiranja vremena akcije dodavanja novih entiteta u model mreže za različiti broj klijenata kako monolitne aplikacije, tako i aplikacije implementirane u Cloud okruženju.

Ključne reči: NMS, Cloud, Azure, mikroservisi, elektroenergetski sistem

Abstract – This paper is dedicated to the creation of applications for updating power grid models in the Cloud environment using the Azure Service Fabric platform. The paper also tests the time of action of adding new entities to the network model for a different number of clients, both monolithic applications and applications implemented in the Cloud environment.

Keywords: NMS, Cloud, Azure, microservices, electroenergy system

1. UVOD**1.1 Azure Service Fabric**

Azure Service Fabric [1] je platforma kao usluga (PaaS) koja je dizajnirana da olakša razvoj, primenu i upravljanje visoko skalabilnim i prilagodljivim aplikacijama za Microsoft Azure platformu na Cloud-u. U osnovi to je arhitektura koja se zasniva na mikroservisnoj arhitekturi u kojoj programeri pretvaraju tradicionalne monolitne aplikacije u skup diskretnih servisa.

Uvođenjem Cloud-a, skaliranje i pouzdanost su napravili potpuno novu revoluciju u implementaciji na različite načine u celom sistemu.

Service Fabric omogućava izgradnju i upravljanje skalabilnim i pouzdanim aplikacijama koje su sastavljene od mikroservisa koji su pokrenuti na više mašina, odnosno na klasteru. Koristeći Service Fabric programeri i administratori mogu izbeći rešavanje složenih infrastrukturnih problema i umesto toga se fokusirati na implementaciju kritičnih, zahtevnih zadataka znajući da su oni skalabilni, pouzdani i da je njima moguće upravljati.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Srđan Vukmirović, vanr.prof.

1.2 Elektroenergetski sistem

Elektroenergetski sistem (EES) je tehnički sistem čiji je osnovni zadatak da osigura kvalitetnu isporuku električne energije uz minimalne troškove. Tehnološki proces u EES sastoji se iz sledećih faza: proizvodnja, prenos, distribucija i potrošnja. Za svaki EES postoji glavni centar upravljanja odakle se upravlja proizvodnjom električne energije. Moderni i veliki EES obuhvataju velika područja jedne ili više država pa se upravljanje EES vrši iz više centara. Upravo iz tog razloga se EES deli na mnogo velikih delova koje je potrebno nadzirati i pratiti stanja u kojima se ti delovi mreže nalaze.

2. OPIS KORIŠĆENIH TEHNOLOGIJA I ALATA

Azure Service Fabric je platforma distribuiranih sistema koja olakšava implementaciju i upravljanje skalabilnim i pouzdanim mikroservisima.

C# je jezik iz familije C jezika, koji je u potpunosti baziran na principima OOP(objektno orijentisanog programiranja).

.NET Framework predstavlja programsko okruženje koje služi za lakši razvoj programa.

WPF (Windows Presentation Foundation) predstavlja tehnologiju za pravljenje klijentskih aplikacija na Windows platformi, koja nudi mnogobrojne opcije za podešavanje izgleda aplikacije.

Microsoft Visual Studio je softversko razvojno okruženje koje omogućava razvoj aplikacija za sve Microsoft-ove platforme i koristi .NET Framework pa je zato pogodan za ovaj projekat.

3. OPIS PROBLEMA

Tematika problema jeste da se omogući rad aplikacije za ažuriranje modela elektroenergetske mreže u Cloud okruženju korišćenjem Azure Service Fabric platformu. Aplikacija ima mogućnost prikaza informacija željenog objekta učitano iz CIM profila kao i omogućavanje korisnicima izmenu, dodavanje i brisanje željenih entiteta. Potrebno je korisniku omogućiti prikaz informacija u realnom vremenu, onemogućiti mu rad sa nevalidnim entitetima, brzo ažuriranje informacija, kao i obaveštenja ukoliko je došlo do promena nekog entiteta sa kojim korisnik trenutno radi. Mikroservisna arhitektura predstavlja evoluciju servisno orijentisane arhitekture (SOA) i sačinjena je od skupa nezavisnih servisa od kojih svaki implementira jedan poslovni zahtev. Glavni cilj jeste podeliti aplikaciju na više manjih nezavisnih mikroservisa i omogućiti međusobnu komunikaciju

mikroservisa kako bi se omogućio neometani rad aplikacije u Cloud okruženju. Neki od benefita korišćenja mikroservisne arhitekture u odnosu na tradicionalnu monolitnu arhitekturu su: nezavisno objavljivanje servisa na produkcijske servere bez posledica po ostatak sistema, veća skalabilnost, nezavisan razvoj servisa, itd.

4. TEORIJSKE OSNOVE

4.1 Mikroservisna arhitektura

Mikroservisna arhitektura [2] poslednjih godina postaje sve popularnija, a naročito u velikim kompanijama koje, zbog njenih mogućnosti, sve više prelaze sa monolitnih aplikacija na mikroservisnu. Među prvim su prešli Amazon i Netflix, kao i kompanije sa ogromnom bazom korisnika kao što je SoundCloud. Manje kompanije i organizacije teže prelaze na mikroservisnu arhitekturu jer još uvek među ljudima vlada mišljenje da je lakše održavati jednu aplikaciju nego više njih i da je to dobro samo za velike kompanije. Međutim, ključna stvar jeste prepoznati pravi trenutak u kojem bi trebalo preći sa monolitne na mikroservisnu arhitekturu, kada monolitna aplikacija nije više optimalna. Problemi koji se najčešće javljaju u monolitnim aplikacijama su: uska grla, održavanje, spor razvojni ciklus, spora isporuka, kao i zastareli softver.

Veliki problem u monolitnim aplikacijama sa velikom obradom podataka predstavlja i održavanje. Ako se obrada tih podataka smesti u jednu veliku grupu pozadinskih procesa, obično je potreban celi tim programera kako bi održao samo taj sistem. Zbog svega ovoga, timovi obično uložio više vremena na rešavanje tehničkih problema nego na pravljenje nekih novih i zanimljivijih stvari. Ovo automatski vodi do novog problema, a to je spora isporuka, što se dešava aplikacijama koje su vremenom nakupile mnogo asset-a i zavisnosti. Ovakve monolitne aplikacije ne odgovaraju ni novopečenim programerima, a ni onima koji su tu od nastanka aplikacije. Novim programerima je potrebno mnogo više vremena da razumeju kod i da budu sposobni da unesu neku malu promenu u kodu.

4.2 Korišćenje Cloud-a

Cloud [2] zamenjuje skupu IT infrastrukturu, omogućava veću fleksibilnost u radu, jednostavnost i sigurnost za poslovne procese uz značajnu uštedu. Garantuje potpunu privatnost i bezbednost podataka uz najsavremenije sisteme zaštite. Ima automatski backup koji omogućava pouzdan oporavak sistema u slučaju nekih poremećaja i kvarova. Podacima se može pristupiti sa bilo kog uređaja, bitno je samo da je taj uređaj povezan na Internet. Među najpopularnijim Cloud platformama danas su Microsoft Azure Service Fabric i Amazon Web Services.

5. IMPLEMENTACIJA REŠENJA

5.1 Arhitektura rešenja

Arhitektura rešenja se sastoji od tri Service Fabric aplikacije pokrenute u Cloud okruženju i dve pokrenute na lokalnu. Service Fabric aplikacije: NMS, Transaction Coordinator i Calculate Engine. Na lokalnu se nalazi Importer koji učitava model iz XML dokumenta i prosledjuje ga na dalju obradu ka GDA Microservice-u. Druga aplikacija na lokalnu jeste korisnička UI aplikacija koja pruža korisniku prikaz informacija entiteta različitih

tipova, pretragu entiteta po svim vrednostima atributa koje sadrže entiteti tog tipa, dodavanje i brisanje entiteta, pa i izmenu vrednosti atributa željenih entiteta. Na slici 1 se nalazi prikaz arhitekture realizovanog rešenja sa svim Service Fabric aplikacijama i njihovim mikroservisima. Prikazani su komunikacioni kanali između mikroservisa koji međusobno komuniciraju, kao i tipovi mikroservisa (Stateless ili Stateful).

Transaction Coordinator Microservice je zadužen za kontrolu distribuiranih transakcija između servisa (NMS i Calculation Engine). Potrebno je da se oba servisa jave Transaction Coordinator-u i tek tada se smatra da je transakcija uspešno završena u suprotnom se vrši rollback. Prilikom podizanja servisa, potrebno je da se prijave Transaction Coordinator-u pozivom Enlist metode. Back up (kopija čitavog modela) se čuva po principu Deep Copy.

NMS Transaction Microservice Nakon uspešno izvršene akcije učitavanja modela i kreiranje Delte, kreira se kopija modela po principu Deep Copy. Ako se uspešno podigne svi servisi i omogućiti se transakcija modela između NMS-a i CE-a poziva se metoda Commit gde se vrši akcija transakcija modela između pomenutih servisa.

NMS Command Microservice je servis zadužen za primanje komandi koje je izdao korisnik korišćenjem UI aplikacije. Komande koje može da primi su vezane za model mreže gde korisnik ima mogućnosti dodavanja novog entiteta, brisanje postojećeg entiteta, kao i izmenu vrednosti atributa negok već postojećeg entiteta.

NMS GDA Microservice je zadužen za kreiranje učitavanje modela, kreiranje Delte, ažuriranje modela, kao i kreiranje kopije modela (Deep Copy). Kad se učita model, kreira Delta i ustanovi da su svi servisi uspešno pokrenuti i spremni za transakciju, učitani model se dalje prosledjuje Calculate Engine-u na obradu. Kada korisnik izda komandu za ažuriranje modela (insert, update ili delete entiteta) i komanda sa svim potrebnim parametrima stigne do GDA Microservice-a vrši se pozivanje metode ApplyDelta. Ova metoda vrši obradu pristiglih podataka i izvršava željenu akciju korisnika, ukoliko je moguće izvršiti.

CE Transaction Microservice nakon uspešno izvršene akcije učitavanja modela i kreiranje Delte i nakon potvrde o uspešnosti podizanja servisa, omogućena je transakcija modela između NMS-a i CE-a. Model se prosledjuje do Cache Microservice-a na dalju obradu.

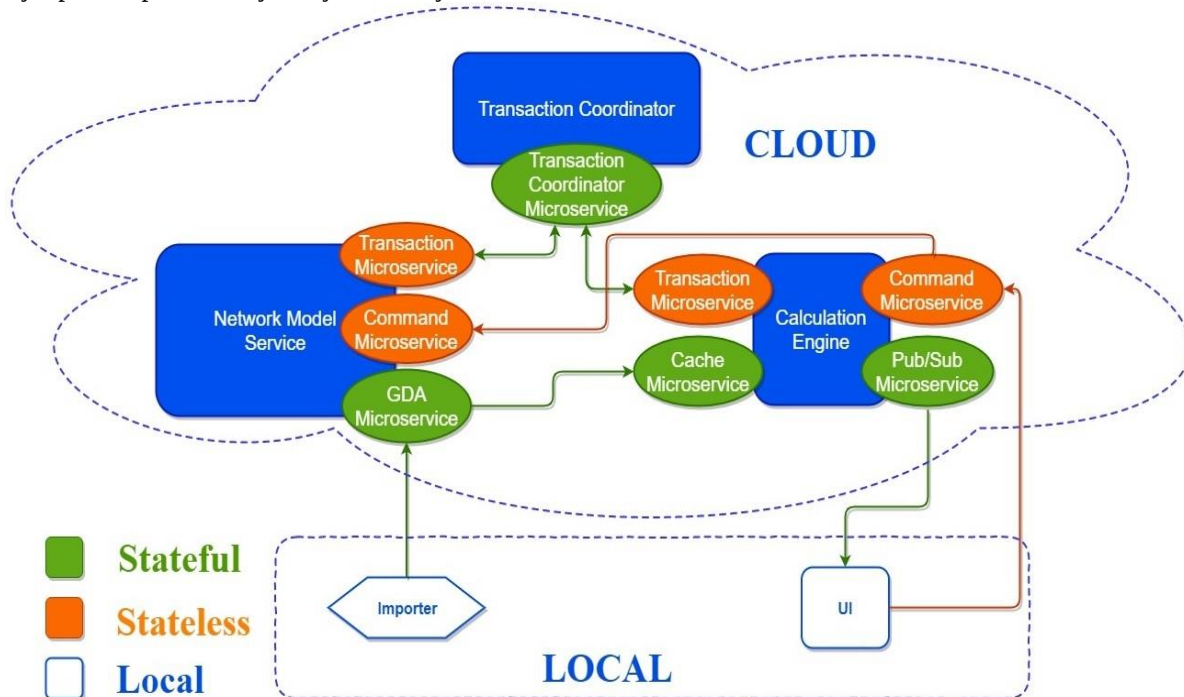
CE Command Microservice je servis zadužen za primanje komandi koje je izdao korisnik korišćenjem UI aplikacije. Jedan deo komandi je vezan za metode koje pruža GDA Microservice, dok je drugi deo komandi vezan za Cache Microservice. Komande vezane za GDA Microservice prosledjuju se ka NMS Command Microservice na dalju obradu, dok se drugi deo komandi prosledjuje ka Cache Microservice-u.

Cache Microservice je kreiran kao Stateful mikroservis zbog toga što se na njemu nalazi model mreže prosleđen sa NMS-a. Kada model stigne sa NMS-a, Cache Microservice kreira IReliableDictionary u kom se čuva pristigli model mreže. Nakon podizanja aplikacije i pretplaćivanje korisnika na Pub/Sub mehanizam, Cache Microservice iz modela izvlači GID-ove entiteta koji se

prosleđuju ka UI aplikaciji kako bi se korisniku omogućio prikaz GID-ova entiteta.

Pub/Sub Microservice pruža mogućnosti Publisher-Subscriber mehanizma. Pub/Sub servis je servis baziran na Publisher/Subscriber obrascu. Obrazac je dizajniran po principu pretplate na određeni događaj. Klijent se pretplaćuje na određene događaje, a servis je dužan da klijenta obavesti o događaju. Komunikacija sa klijentima se obavlja putem poruka koje klijent kasnije obradi.

Pretplaćeni klijenti se moraju čuvati na servisu da bi servis znao koga je potrebno obavestiti o događajima. Komunikacija između klijenta i servisa je asinhrona što znači da klijent nije blokiran dok čeka na poruke. Informacije o klijentima Pub/Sub servis čuva u IReliableDictionary da bi se i prilikom gašenja servisa informacije o klijentu sačuvala.



Slika 1. Arhitektura rešenja

5.3 Azure Service Fabric node

Service Fabric Cluster [3] je mrežno povezan skup virtuelnih ili fizičkih mašina na kojima se mikroservisi pokreću i vrši se njihovo upravljanje. Mašina ili VM koji je deo klastera naziva se čvor klastera. Klasteri se mogu skalirati na hiljade čvorova. Ako se klasteru dodele novi čvorovi, Service Fabric rebalansira replike servisne particije i instance na povećanom broju čvorova. Tip replike određuje njenu ulogu u skupu replika:

Primary (P): U skupu replika postoji jedna primarna koja je odgovorna za izvođenje operacija čitanja i pisanja.

ActiveSecondary (S): Ovo su replike koje primaju ažurirana stanja od primarne, primenjuju ih i zatim vraćaju potvrde.

IdleSecondary (I): Ove replike prave primarne replike. Oni primaju stanje iz primarne pre nego što mogu da se unaprede u aktivnu sekundarnu

None (N): Ove replike nemaju odgovornost u kompletnu repliku.

5.4 Testiranje

Testirana je aplikacija za ažuriranje modela elektroenergetske mreže pre i posle njene implementacije u Cloud okruženju. Aplikacija ima mogućnost dodavanja, brisanja i izmenu entiteta modela mreže. Dodavanje novog entiteta u model mreže predstavlja najkompleksniju akciju u odnosu na prethodno navedene akcije i zbog toga se u ovom testu vrši merenje brzine dodavanja novih entiteta u

model mreže od strane različitog broja klijenata istovremeno.

Tabela 1. Rezultati testiranja

Br. klijenata	Monolitna aplikacija [s]	Aplikacija u Cloud okruženju[s]
10	1	1
50	2	1
100	4	3
250	8	6
500	17	11
1000	31	19

U tabeli 1 prikazana su vremena izvršavanja akcije dodavanja novih entiteta u zavisnosti od broja klijenata koji istovremeno zahtevaju kreiranje entiteta. U prvoj koloni prikazan je broj klijenata koji zahtevaju kreiranje novih entiteta, u drugoj koloni se nalaze vremena izvršavanja akcije vezane za monolitnu aplikaciju pre implementacije u Cloud okruženju, dok se u trećoj koloni nalaze vremena vezana za aplikaciju implementiranu u Cloud okruženju.

Vrednosti u tabeli predstavljane su u sekundama i zaokružene su na celobrojne vrednosti. Iz tabele se jasno može zaključiti da su vremena izvršavanja akcije dodavanja novog entiteta za mali broj korisnika približna, dok se sa porastom broja klijenata vrednosti međusobno

razlikuju. Sa povećanjem broja korisnika, monolitnoj aplikaciji treba više vremena za izvršenje akcije u odnosu na aplikaciju implementiranu u Cloud okruženju i na osnovu toga se može zaključiti da je rad monolitne aplikacije sporiji u odnosu na aplikaciju koja je implementirana u Cloud okruženju.

6. ZAKLJUČAK

Cilj multiklijentskog korisničkog rešenja za ažuriranje modela elektroenergetske mreže jeste ostvarivanje mogućnosti praćenja trenutnih stanja modela mreže sa svim njenim elementima i kontrolisanje toka podataka između elemenata sistema. Aplikacija pruža mogućnosti jednostavne izmene modela, dodavanja i brisanja entiteta, te i prikaz svih informacija o entitetima iz modela.

Azure Service Fabric je platforma kao usluga koja je dizajnirana da olakša razvoj, primenu i upravljanje visoko skalabilnim i prilagodljivim aplikacijama za Microsoft Azure platformu na Cloud-u koja predstavlja arhitekturu koja se zasniva na mikroservisnoj arhitekturi. Teži se ka tome da se tradicionalne monolitne aplikacije transformišu u skup diskretnih mikroservisa. U ovom radu je opisana jedna takva transformacija gde je izvršena transformacija monolitne aplikacije u aplikaciju sačinjenu od više diskretnih mikroservisa pokrenutih u Cloud okruženju.

7. LITERATURA

- [1] Haishi Bai, Programming Microsoft Azure Service Fabric, 2016..
- [2] M.Simonović, Tehnologija Cloud Computing-a, Univerzitet Singidunum, 2013.
- [3] Haishi Bai, Programming Microsoft Azure Service Fabric, 2nd Edition, 2018.

Kratka biografija:



Bogdan Kovačev rođen je 20.04.1996. godine u Novom Sadu. Diplomirao je na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Primenjeno softversko inženjerstvo 2019. godine.

РАЗВОЈ И ИНТЕГРАЦИЈА SIEM СОФТВЕРСКОГ РЕШЕЊА У SCADA СИСТЕМЕ

DEVELOPMENT AND INTEGRATION OF SIEM SOFTWARE SOLUTION INTO SCADA SYSTEMS

Ђорђе Тошић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У раду су описане сајбер претње по сигурност критичних инфраструктура са акцентом на SCADA системе. Поред теоријских основа рада, имплементирана је апликација која у себи садржи SIEM софтверско решење.

Кључне речи: Критичне инфраструктуре, SCADA, SIEM, DDoS, сајбер напади

Abstract – The paper describes cyber threats to the security of critical infrastructures with an emphasis on SCADA systems. In addition to the theoretical foundations of the work, an application containing a SIEM software solution has been implemented.

Keywords: Critical infrastructures, SCADA, SIEM, DdoS, cyber attacks

1. УВОД

Сајбер систем представља кичму критичних инфраструктура, што значи да било какав компромис сајбер система могао би имати значајан утицај на поуздан и безбедан рад физичких система који се на њега ослањају. Фокус овог рада је на сајбер сигурности електроенергетских инфраструктура.

У раду су описани типични сајбер напади на критичне инфраструктуре, са акцентом на опис најучесталијих напада на SCADA системе. Задатак мастер рада јесте имплементација и интеграција SIEM софтвера у DERMS апликацији, која је имплементирана током мастер студија и њене функционалности су укратко описане у овом раду.

За потребе тестирања симулирани су сајбер напади (DDoS напад, подметање неисправног модела). Циљ је имплементирати SIEM компоненту која ће бити способна да препозна такве нападе, и да обавести корисника путем корисничког интерфејса.

Пре саме имплементације SIEM компоненте, направљен је threat model DERMS апликације, како би се анализирале слабе тачке система и места која могу бити мета потенцијалних сајбер напада.

2. ОПИС КОРИШЋЕНИХ ТЕХНОЛОГИЈА И АЛАТА

У овом поглављу ће бити укратко описане коришћене технологије и алати.

• **Microsoft Visual Studio** је софтверско развојно окружење које омогућава развој апликација за све Microsoft-ове платформе. Садржи следеће алате: *code editor* за писање кода и аутоматско проналажење синтаксних грешака у коду, *debugger* за проналазак свих типова грешака у коду и њихово исправљање, *code profiler* алат за профилрање кода (путем њега се може проверити како се рукује меморијом, какве су перформансе програма итд.) као и алате за дизајнирање и алате који помажу програмеру да се лакше снађе у *Visual studio* попут *Solution Explorer-a*, *Object Browser* итд.

• **C#** је језик из фамилије C језика, који је у потпуности базиран на принципима ООП (Објектно оријентисаног програмирања).

• **XAML** (*Extensible Application Markup Language*) је маркуп језик чија је намена поједностављење креирања UI за интеракцију са корисником у .NET Framework апликацијама. Погодан је за креирање и иницијализацију објеката. Поента XAML-a, је да омогући програмерима да раде са експертима из других области.

• **.NET** представља програмско окружење које служи за лакши развој програма. Садржи CLR (*Common Language Runtime*) који извршава написане програме, који брине о капацитету процесора, рукује заузетом меморије, управља изузецима и обезбеђује безбедност апликације.

• **WPF** (*Windows Presentation Foundation*) је графички подсистем за рендеровање корисничког интерфејса у апликацијама заснованим на Windows-u. Представља технологију за прављење клијентских апликација на Windows платформи, која нуди многобројне опције за подешавање изгледа апликације.

• **Microsoft Threat Modeling Tool** је кључни елемент *Microsoft Security Development Lifecycle (SDL)*. Омогућава софтвер архитектама да идентификују и ублаже потенцијалне сигурносне проблеме, пре него што дође до искориштавања сигурносних проблема. Рано уочени проблеми у великој мери смањују укупне трошкове и време утрошено на развој софтвера.

Напомена:

Овај рад је проистекао је из мастер рада чији ментор је био др Дарко Чапко, ванр. проф.

3. ОПИС ОСНОВНИХ САЈБЕР НАПАДА НА КРИТИЧНЕ ИНФРАСТРУКТУРЕ

Континуирано функционисање земаља, влада, међународних организација, корпорација и многих јавних служби често зависи од несметаности приступа критичној инфраструктури која може бити дефинисана као системи и имовина, физичка или виртуелна, која је толико витална, па би онеспособљеност ових система или уништавање имало велики утицај на националну, економску или оперативну сигурност, као и на јавно здравље или безбедност.

3.1. Сајбер напади на SCADA системе

Када је реч о нападима на критичне инфраструктуре, мета напада често представљају SCADA системи. Претње SCADA системима су класификоване као интерне и спољне претње[1]. Интерне претње вребају од људи који су запослени у организацији, који због тога имају већи физички приступ поверљивим и критичним информацијама SCADA система. Напади од стране нападача који немају приступ пословним објектима третирају се као спољне претње. Обе врсте претњи су једнако опасне.

Сигурност SCADA система зависи од слабих сигурносних тачки система. Уобичајена претња је убацивање малвера путем преносивих уређаја за складиштење и малициозним прилозима који се шаљу електронском поштом. Недовољно заштићене бежичне приступне тачке у систему су потенцијална улазна тачка. Већина SCADA мрежа повезана је са корпоративним мрежама помоћу *Virtual Private Networks (VPN)*. Ако нападач има приступ корпоративној мрежи, он може приступити SCADA мрежи релативно лако. Зато је веома важно обезбедити корпоративну мрежу, како се њој не би могло приступити преко несигурних веза [2].

3.1.1. Сајбер напади на хардвер SCADA система

Претње по хардвер SCADA система јесу задобијање даљинске контроле над уређајима од стране нападача. Ови напади могу проузроковати да уређаји откажу при ниским вредностима, или да се неки аларм не укључи када би требао. Друга могућност напада јесте да нападач неовлашћено задобије приступ, и да након тога мења вредности које се приказују оператеру. У том случају могло би се десити да оператер не буде свестан да се одређени аларм упалио, и да указује на опасност, самим тим би изостала реакција оператера, што би могло да проузрокује хаварију система.

Главни проблем код превенције сајбер напада на хардвер SCADA система представља контрола приступа. Као један од напада који користе поменути слабу тачку, и који се често догађају, јесу тзв. „*doorknob-rattling*“ напад. Нападач комбинује неколико уобичајених комбинација корисничког имена и лозинке, на неколико рачунара, што резултује врло малим бројем промашаја, и тако нападач не пробије дозвољени број покушаја, и приступи систему. Овај напад може остати непримећен, осим ако се подаци који се односе на пријаву не сакупљају и не проверавају на „*doorknob-rattling*“ напад.

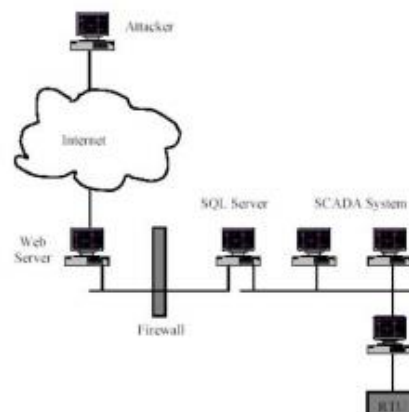
3.1.2. Сајбер напади на софтвер SCADA система

Buffer Overflow – многи напади се свде на то да проузрокују *Buffer Overflow* као средство које ће евентуално проузроковати неуобичајен рад програма. Неке опште методе за постизање *Buffer Overflow-a* су *stack smashing* и манипулација показивачем функције.

Ефекти оваквих напада су ресетовање лозинки, модификација података, инсталирање малициозног кода и друге. SCADA уређаји који се налазе у пољу се јако ретко рестартују, због тога, посебно у застарелим мрежама долази до фрагментације меморије, која доводи до застоја програма [3].

SQL Injection – већина малих и индустријских база података користи SQL исказа за модификацију и манипулацију подацима. Данас када постоји приступ SCADA системима преко интернета, *SQL Injection*, као један од најчешћих напада на интернету, има велики утицај у креирању сигурности SCADA система [4].

SQL Injection напад дешава се када је нападачу омогућена модификација уноса података преко *Web* апликације, која не успе да препозна модификовани унос, и тако нападач убаци нежељене изразе у SQL упит. Намерне модификације базе података могу проузроковати катастрофалне последице. Пример *SQL Injection* напада приказан је на слици 1.



Слика 1. *SQL Injection* напад

3.1.3. Комуникациони сајбер напади на SCADA системе

Нападе на комуникациони стек, који могу нанети штету SCADA системима, можемо поделити на: нападе на мрежни слој, транспортни слој, апликативни слој и на нападе у зависности од протокола који SCADA системи користе.

Distributed Denial of Service (DDoS attack) – Напади који имају за циљ да преоптерете и тако онемогуће услугу сервера. Зарад ефикаснијих напада, користе се групе рачунара тзв. „*botnet*“. Термин *botnet* представља групу рачунара који су заражени малициозним садржајем, и под контролом су нападача. Методе на који се рачунари из *botnet* мреже заразе су: тројанац у комбинацији са *backdoor*, или путем социјалног инжењеринга.

Примери *DDoS* напада су [5]:

- *SYN Flood* – искориштава слабости *TCP three-way handshake* конекције. Сервер добија *SYN* поруку да започне „*handshake*“. Сервер затим узвраћа *ACK* поруку, страни која је иницирала почетак комуникације, која затим прекида комуникацију. Након тога страна која иницирала конекцију прекида комуникацију, а сервер остаје загушен истим захтевима, који резултују тајмаутом и тако загушују сервер.
- *Zero-day DDoS attack* – искориштава рањивости система, слабе тачке, за које још увек нису направљене закрпе.
- *Ping of Death* – манипулише *IP* протоколом, шаље пингове, са циљем да онеспособи сервер. У прошлости је био заступљен, данас се ретко среће.
- *Smurf Attack* - манипулише *IP* протоколом и *ICMP* протоколом користећи малициозни софтвер звани *Smurf*. Лажира *IP* адресу, и помоћу *ICMP* протокола шаље пингове.

3.2. Најпознатији напади на критичне инфраструктуре

У данашњици веома моћно оружје у сукобима између држава постали су сајбер напади. Најпознатији напади на критичне инфраструктуре у скорој историји су [6]:

Heartland Payment Systems 2008. године – хакерска група „*Shadowcrew fame*“ која је починила многобројна дела из групе сајбер криминала, украла је око 100 милиона бројева кредитних картица, чиме је нанела штету од 140 милиона долара.

Stuxnet 2010. године - Прво сајбер оружје које је способно да изазове напад који изазива кинетичко дејство јесте вирус назван *Stuxnet*, који је развијен и изграђен у информатичким лабораторијама *NSA*, у сарадњи *CIA* и израелске обавештајне службе. Био је намењен да физички уништи опрему иранског нуклеарног објекта у Натанзу.

Под маском да преузима индустријску контролу над системом *SCADA*, софистицирани црв требало је да оштети око хиљаду центрифуга за обogaћивање нуклеарног материјала.

Night Dragon 2011. године - У фебруару 2011. године компанија која се бави производњом антивирусног софтвера (*McAfee*) објавила је да су пет међународних енергетских и нафтних компанија биле мете комбинованих напада укључујући социјални инжењеринг, тројанце и друге нападе. Потврђено је да су напади под називом „*Night Dragon*“ трајали више од две године и верује се да су усмеравани из Кине.

BlackEnergy напад на електроенергетску мрежу Украјине 2015. године – Напад је изведен тако што су нападачи слали *mail-ove*, са *Microsoft Office* документима, који када се отвори аутоматски зарази корисника. Напад је проузроковао да 1,4 милиона корисника остане без напајања електричном енергијом.

Сајбер напад на болницу у Лос Анђелесу 2016. године – Нападаци су успоставили контролу над базом података болнице, у којој су се налазиле информације о пацијентима, њихова историја болести, терапије које су примали. Нападаци су шифровали податке и доктори нису могли да приступе бази без кључа. Нападаци су тражили 3,6 милиона долара за кључ.

4. МЕТОДИ ЗАШТИТЕ КРИТИЧНИХ ИНФРАСТРУКТУРА ОД САЈБЕР ПРЕТЊИ

Сајбер сигурност је најкритичнији аспект технолошки заснованог света. Све технологије које се заснивају на комуникационим и информационим системима зависне су од сајбер сигурности.

Јавни и приватни сектор сваке године троше огромне своте новца на технологије, сигурносне софтвере и хардвер уређаје који ће повећати сајбер сигурност у њиховим компанијама, али су ти системи и даље рањиви.

4.1. SIEM софтвер

SIEM систем је сложена група технологија, дизајнирана да пружи визију и јасан преглед система у којем се користи. Од велике је користи аналитичарима сигурности система и *IT* администраторима. Пружа холистички поглед на оно што се дешава на мрежи у реалном времену, и помаже *IT* тимовима да буду активнији у решавању сигурносних претњи.

Неке од основних функционалности су:

- *Log management*
- *IT regulatory compliance*
- *Event correlation*
- *Active response*
- *Endpoint security*

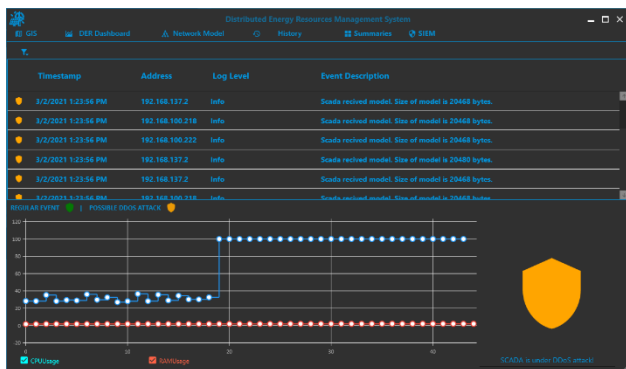
5. ИМПЛЕМЕНТАЦИЈА И ИНТЕГРАЦИЈА SIEM ПРОГРАМСКОГ РЕШЕЊА У DERMS АПЛИКАЦИЈУ

У овом поглављу биће описано *SIEM* програмско решење и његове функционалности у оквиру *DERMS* апликације. *SIEM* компонента прикупља лог фајлове у које се бележи рад осталих сервиса, прикупљене податке анализира и по унапред дефинисаним правилима проверава да ли је дошло до неког сајбер напада, или неке друге неуобичајене активности.

5.1. Детекција и последице DDoS напада

На слици 2. приказано је стање *SCADA* сервиса током *DDoS* напада. *SIEM* компонента функционише тако што на одређени временски период чита информације из лог фајлова.

Пошто се подаци уписују у логове константно, и после гашења апликације, информације у логовоима остају трајно у њима. Зарад тога учитавају се информације из логова, само које су уписане у току рада апликације.



Слика 2. Детекција DDoS напада

Као што је приказано на слици 2, на графику на којем се приказује оптерећеност CPU и RAM, уочава се период пре почетка DDoS напада и након почетка симулације DDoS напада. Пре напада процесор није много оптерећен, и те промене оптерећености процесора не варирају много кроз време. Након покретања симулације DDoS напада, SIEM компонента региструје велики број конекција са различитих машина, које у истом тренутку шаљу податке на SCADA сервис. У том тренутку оптерећеност процесора расте на 100%, и SIEM компонента, путем корисничког интерфејса алармира корисника да је у току DDoS напад.

6. МАШИНСКО УЧЕЊЕ У SIEM СИСТЕМИМА

Машинско учење се односи на грану вештачке интелигенције. Машинско учење користи алгоритме вештачке интелигенције за учење из својих искустава током времена након почетног уноса података. Стога, машинско учење у SIEM системима узима правила и податке о сајбер сигурности, како би олакшало аналитику података о сигурности. Као резултат, може смањити напор или време утрошено на базичне задатке или чак и на оне софистицираније. Уколико се правилно конфигурише, машинско учење заправо може доносити одлуке на основу података које прима и у складу са тим променити понашање [7]. Машинско учење у SIEM системима може омогућити аналитику претњи и створити обавештења о ризику у реалном времену.

7. ЗАКЉУЧАК

Сајбер напади на критичне инфраструктуре су са дигитализацијом тих система постали свакодневница. Последице сајбер напада на критичне инфраструктуре могу бити катастрофалне. Поред огромне финансијске штете коју сајбер напади изазивају, постоји опасност и по људске животе, животну средину и друге аспекте живота. Из тих разлога неопходно је сигурност система подићи на највиши ниво. Софтверско решење SIEM се последњих две деценије истиче као моћан алат за детекцију и превенцију сајбер претњи, као и мониторинг целокупног система.

У раду је описан утицај DDoS напада на перформансе система и функционалност апликације. Након симула-

ције напада уочава се пораст оптерећења процесора, док се оптерећење RAM меморије незнатно повећава. У току напада основне функционалности апликације било је отежано извршавати, команде које се шаљу на сервис погођен DDoS нападом нису успешно извршаване.

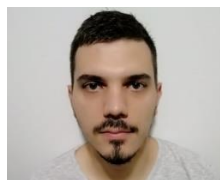
За разлику од DDoS напада, подметање неисправног модела није значајно утицало на перформансе система. Рад апликације, што се перформанси тиче, настављао се неометано. Последице напада огледале су се у различитим резултатима за исте елементе система, што јасно указује да модел није валидан, јер се резултати не подударају.

За даље усавршавање апликације засноване на SIEM софтверском решењу, алгоритми машинског учења би се могли имплементирати у такве системе. Тако би систем, на основу претходних сајбер напада, могао да детектује нове сајбер претње, без претходног дефинисања услова који асоцирају на поједини сајбер напад.

8. ЛИТЕРАТУРА

- [1] W. T. Shaw, „Cybersecurity for SCADA Systems“, 2008.
- [2] Siddharth Sridhar, G. Manimaran, „Data Integrity Attacks and their Impacts on SCADA Control System“, Department of Electrical and Computer Engineering Iowa State University, 2010.
- [3] Siddharth Sridhar, G. Manimaran, „Data Integrity Attacks and their Impacts on SCADA Control System“, Department of Electrical and Computer Engineering Iowa State University
- [4] T. Paukatong, „SCADA Security: A New Concerning Issue of an In-house EGAT-SCADA“, IEEE/PES Transmission and Distribution Conference & Exhibition: Asia and Pacific Dalian, China, 2005.
- [5] Douligieris, Christos, and Aikaterini Mitrokotsa, "DDoS attacks and defense mechanisms: classification and state-of-the-art." Computer Networks, 2004.
- [6] Kim Zetter "Everything We Know About Ukraine's Power Plant Hack". Wired, 2016.
- [7] Anumol, E. T. "Use of machine learning algorithms with SIEM for attack prediction." Intelligent Computing, Communication and Devices. Springer, New Delhi, 2015.

Кратка биографија:



Ђорђе Тошић рођен је у Сремској Митровици 1996. године. Дипломирао је 2019. године на Факултету техничких наука, смер Примењено софтверско инжењерство, на којем исте године уписује мастер студије.

**PREDIKCIJA POZICIJE FUDBALSKOG IGRAČA UPOTREBOM ALGORITAMA
MAŠINSKOG UČENJA****PREDICTION OF FOOTBALL PLAYER'S POSITION USING MACHINE LEARNING
ALGORITHMS**

Dragan Škiljević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Fudbal je kolektivni sport koji se igra između dvije ekipe, sa po jedanaest igrača. Iako igrači igraju na unaprijed određenoj poziciji, oni mogu lako preći i na neku drugu poziciju. U ovome radu je vršena predikcija najbolje pozicije igrača na osnovu njegovih fizičkih i psihičkih osobina. Osnovni motiv ovoga rada jeste olakšavanje posla fudbalskim stručnjacima koji se profesionalno bave svojim poslom. Rješenje ovoga projekta bi u velikoj mjeri olakšalo posao trenerima čiji klubovi se susreću sa mnoštvom povreda, pa je potrebno često vršiti promjenu formacije tima. To bi pomoglo da se u maksimalnoj mjeri iskoristi potencijal svakog igrača. Da bi se što lakše odredila pozicija na kojoj će određeni igrač igrati, u ovom radu, koristimo skup podataka sa 65 atributa za svakog igrača, na osnovu kojih će se određivati pozicija uz pomoć obučavanja sledećih modela: Multinomial Logistic Regression, K-Nearest Neighbors, Random Forest, Gaussian Naive Bayes, Support Vector Machine.

Ključne reči: fudbal, pozicija, mašinsko učenje, klasifikacija, multilabel

Abstract – Football is a collective sport that is played between two teams, composed of eleven players each. Although players play in a predetermined position, they can quickly move to another position. In this paper, the prediction of the best position for a football player was made based on his physical and mental characteristics. This approach's primary motivation is to provide additional information to coaches whose clubs are faced with many injured players. Each player was represented with a vector of 65 attributes, and the following models: Multinomial Logistic Regression, K-Nearest Neighbors, Random Forest, Gaussian Naive Bayes, Support Vector Machine were experimented with.

Keywords: football, position, machine learning, classification, multilabel

1. UVOD

S obzirom da je fudbal najvažnija sporedna stvar na svijetu, on je široko prisutan u svim narodnim masama. Sa porastom popularnosti rastu i cijene igrača koje u današnje vrijeme dostižu milionske iznose.

NAPOMENA:

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

Cijene igrača su takođe tijesno povezane sa pozicijom na kojoj igrač igra. Obično najveće cijene imaju fudbaleri koji igraju u napadu. Kako se sve veći novac ulaže u kupovinu igrača, veoma je bitno da se uloženi novac isplati. Pozicija na kojoj će igrač igrati je jedan od ključnih faktora da bi se iskoristio sav njegov potencijal.

Poznato je da ishod fudbalske utakmice često zavisi od izbora pravilne formacije. Kako je izbor formacije lična odluka trenera, tako je on suočen sa problemom izbora najboljih igrača za svaku poziciju. Cilj ovog rada je izbor najbolje pozicije za fudbalskog igrača na osnovu njegovi fizičkih i psihičkih osobina. To bi u velikoj mjeri pomoglo trenerima prilikom izbora igrača za određenu poziciju. Sistem predstavljen u ovom radu bi u velikoj mjeri eliminisao subjektivnu pristrasnost prilikom izbora igrača koji će nastupiti. To bi kasnije dovelo do boljih rezultata i smanjenja nepotizma u timu. Fudbalski skauti su često izloženi problemu izbora igrača koji bi se najbolje uklopilo u njihov tim. Pošto su cijene igrača enormno porasle pravljenje grešaka mora biti svedeno na minimum, jer svaka greška može da dovede do finansijske krize kluba. Ovo rješenje bi takođe olakšalo rješavanje ovog problema.

U ovom radu će biti predstavljeno rješenje koje će vršiti izbor pozicije na osnovu sledećih parametara: brzina, kontrola lopte, dribling, skok, pregled igre, agresivnost, smirenost itd.

Preostali dio rada ograničavan je na sledeći način. U drugom poglavlju će biti prezentovani radovi koji se bave sličnom problematikom. U trećem poglavlju će biti opisan skup podataka i način pripreme podataka za obučavanje i validaciju modela. U četvrtom poglavlju biće predstavljena metodologija za klasifikaciju koja je korišćena za rešavanje problema određivanja najbolje pozicije igrača. U petom poglavlju su prikazani i diskutovani rezultati koji su postigli modeli. Na kraju će biti izveden zaključak ovog rada i navedeni pravci u kojima bi se dalje mogla razvijati i unapređivati ova problematika.

2. PRETHODNA RJEŠENJA

U radu [1], koji se bavi predviđanjem i prepoznavanjem talenta u fudbalu na osnovu individualnih kvaliteta igrača. Kvaliteti su klasifikovani u 3 grupe: fizičke, mentalne i tehničke. Tu spadaju: brzina, agilnost, skok, visina, snaga, sposobnost čitanja igre, smirenost, kreativnost, samouvjerenost, sposobnost u predvođenju tima, pas igra, šut, igra glavom, završnica, uklizavanje, ubačaj, odbrana.

Algoritmi klasifikacije koji su korišteni u ovom radu su: *Bayesian Networks*, *Decision Trees*, i *K-Nearest Neighbor*. Najveća tačnost klasifikacije postignuta tokom eksperimenata iznosila je 99% za *Bayesian Networks*, *Decision Tree* 98%, a *Nearest Neighbor* 97%. Dodatna evaluacija sistema je vršena uz pomoć 20 fudbalskih stručnjaka koji su bili treneri i/ili fudbalski menadžeri. Od njih je zatraženo da prodju kroz funkcionalnosti koje sistem nudi i ostave povratnu informaciju o tome kako su zadovoljni sistemom. Analizom je utvrđeno da se 70% korisnika u potpunosti slaže i misli da bi sistem bio od pomoći u realnom svijetu, 15% se prilično slaže sa sistemom, a 15% korisnika nisu sigurni da bi sistem bio od koristi i da bi se mogao implementirati u fudbalskim organizacijama.

Cilj projekta [2] jeste implementacija modela za preporučivanje pozicije igrača na osnovu fizičkih atributa. Rješenje je predstavljeno uz pomoć sledećih linearnih tehnika: *Linear regression*, *Linear discriminant analysis*, *Quadratic discriminant analysis* i *Multinomial logistic regression*. Ulazne promenljive su snaga i izdržljivost i imaju vrijednosti između 0 i 100. Izlazna promenljiva je pozicija i uzima vrijednosti {"CB": srednji bek, "CM": srednji vezni, "ST": napadač} koji predstavljaju 3 ključne pozicije. Odbrambeni igrači su klasifikovani tako da su u prosjeku jači i imaju manje izdržljivosti. Igrači centralnog veznog reda su u prosjeku prepoznati kao slabiji. Karakteristike napadača su dobra kontrola lopte. Prema podacima, prosječna snaga i izdržljivost napadača su ustvari između prosjeka za odbrambene i centralne vezne igrače. Rezultati su dobijeni uz pomoć *precision* metrike.

LR = 50.5% (CB: 52.9% CM: 56.6%, ST: 41.9%)

LDA = 51.0% (CB: 54.5%, CM: 57.0%, ST: 41.5%)

QDA = 51.3% (CB: 53.1%, CM: 57.4%, ST: 43.4%)

MLR = 51.1% (CB: 54.2%, CM: 57.4%, ST: 41.7%)

Za razliku od navedenog rada, dodate su nove pozicije: golman, bek i krilo.

Rad [3] se bavi istraživačkom analizom uz pomoć koje se predviđa položaj odnosno pozicija igrača koristeći različite algoritme mašinskog učenja. Podaci koju su korišćeni sadrže oko 18.000 igrača sa 75 karakteristika po igraču. Prilikom izrade modela korišteni su sledeći algoritmi: *K-Nearest Neighbors*, *Random Forest*, *Support Vector Machine*, *Neural Network* (NNET). Tačnost pomenutih algoritama iznosila je: KNN-81.97%; RF-83.06%; SVM-82.89%; NNET 0.00%. Ovo nije iznenađenje jer se njihove tehničke osobine znatno razlikuju. Bitno je napomenuti da je tačnost prilikom izbora golmana iznosila 100%. Za razliku od pomenutog rada, sistem predstavljen u ovom radu omogućava da igrač može igrati na više od jedne pozicije, što dovodi do *multilabel* problema koji ćemo riješiti na nekoliko načina. Prilikom izrade modela iskoristićemo gore pomenute algoritme *Random Forest* i *SVM*.

Rad [4] je vršio je predikciju pozicije igrača iz popularne igre FIFA 19. Skup podataka je preuzet iz FIFA 19 baze podataka koja sadrži 18207 igrača od kojih svaki sadrži preko 80 različitih atributa. U inicijalnom skupu podataka igračima je bila dodjeljena 1 od 27 pozicija. Da bi se

uprostila struktura formirane su ukupno 4 pozicije koje mogu biti dodjeljene igračima i to su: golman, odbrambeni, vezni, i napadač. Prilikom izrade modela za predikciju pozicije igrača su korišteni *Decision Tree* i *Random Forest* algoritmi. Koristeći *Decision Tree* algoritam postignuta je tačnost od 85% uz pomoć *accuracy* metrike koja predstavlja odnos tačno klasifikovanih igrača i ukupnog broja igrača. Tačnost po klasama je bila najveća za golmana i iznosila je 100%. Odbrambeni igrači su tačno klasifikovani u 79% slučajeva, vezni igrači u 84% slučajeva, a napadači su ostvarili tačnost od 91%. Koristeći *Random Forest* algoritam ostvarena je tačnost od 93%. Analizom atributa izabran je najuticajniji atribut koji je u ovom slučaju bio *sliding tackle* (uklizavanje). Samo na osnovu ovoga parametra moguće je odrediti poziciju sa tačnošću od 72%.

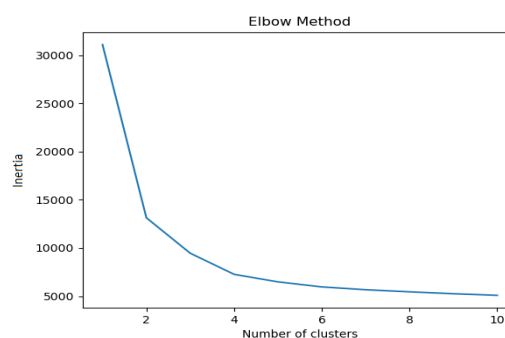
3. SKUP PODATAKA

U ovom poglavlju se opisuje postupak koji dovodi do formiranja setova podataka, koji se dalje koriste za formiranje modela sistema. Inicijalni set podataka je preuzet sa stranice (https://public.tableau.com/s/sites/default/files/media/fifa18_clean.csv), a evaluacija parametara je vršena na osnovu svjetskog prvenstva 2018. u Rusiji. Skup podataka se sastoji od 17981 igrača od kojih svaki posjeduje 65 parametara. Neki od glavnih parametara su: godine igrača, brzina, agresivnost, agilnost, kontrola lopte, smirenost, ubačaj, dribling, volej, uklizavanje, igra glavom, skok, sprint, snaga, izvođenje penala, pregled igre, itd.

Ovaj skup podataka je podijeljen u omjeru 80 : 10 : 10, što znači da se 80% podataka koristi kao obučavajući skup, 10% kao validacioni skup, a preostalih 10% kao testni skup.

3.1 Analiza podataka

Inicijalni data set posjeduje 12 pozicija na kojima igrači mogu biti raspoređeni. Analizom klastera uz pomoć *Elbow* metode dobijeni su rezultati prikazani na grafiku 3.2.1.

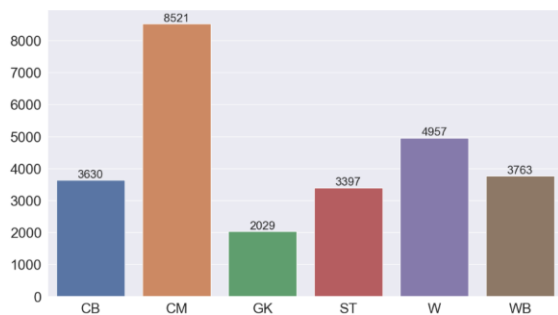


Grafik 3.2.1 Rezultat Elbow metode

Sa slike se vidi da 12 ne predstavlja optimalan broj klasa, već je optimum 4, ali s obzirom da današnji fudbal zahtjeva više od 4 pozicije, kao kompromis sistem će koristiti 6 pozicija nad kojima će se izvršavati algoritmi, a to su: Golman (GK), Štoper (CB), Bek (WB), Vezni (CM), Krilo (W), Špic (ST).

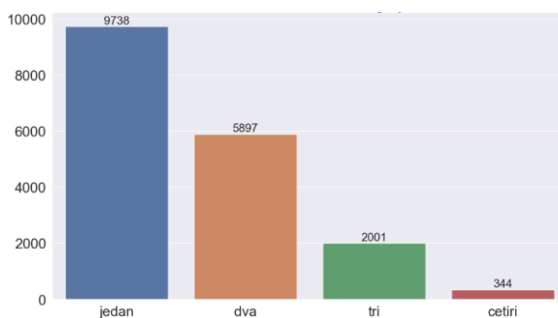
Kako inicijalni skup podataka sadrži neke parametre koji nisu potredni za analizu, kao sto su ime, klub, država, mjesto rođenja itd. oni su izbačeni iz skupa atributa. Golman posjeduje neke atribute koji nemaju vrijednosti, pa su u cilju smanjenja grešaka ove vrijednosti popunjene nulama.

Daljom analizom inicijalnog skupa podataka je utvrđen broj igrača koji igraju na određenim pozicijama. Rezultat analize je prikazan na grafiku 3.2.1



Grafik 3.2.1 Raspodjela podataka po pozicijama

Sa slike se može jasno vidjeti da najveći dio igrača iz skupa podataka pripada kategoriji veznih igrača, jer se oni ne previše ne ističu ni po kojim osobinama i predstavljaju najveći problem prilikom klasifikacije. Nasuprot tome igrači na poziciji golmana, koji imaju istaknute odbrambene osobine, predstavljaju najmanji dio data seta, ali se jasno razlikuju po karakteristikama. S obzirom da se vidi sa slike da je zbir svih igrača po pozicijama veći od ukupnog broja igrača iz data seta, dolazimo do zaključka da neki igrači mogu igrati na više pozicija. Daljom analizom je utvrđena podjela igrača po broju pozicija, a to je prikazano na grafiku 3.2.3.



Grafik 3.2.3 Raspodjela podataka u odnosu na broj pozicija

Grafik 3.2.3 nam pokazuje jasno da najveći broj igrača igra na samo jednoj poziciji odnosno 9738 igrača što čini 54.15% od ukupnog broja.

U inicijalnom setu postoje i neka obilježja čije vrijednosti nedostaju pa su, zbog nemogućnosti pojedinih algoritama da formiraju model nad nepotpunim skupom podataka, te vrijednosti zamijenjene nulom.

Normalizacija podataka jedan je od pristupa pre obrade gdje se podaci skaliraju ili transformiraju kako bi dali jednak doprinos svakog obilježja. Kako u ovome slučaju postoje obilježja čiji se domen u velikoj mjeri razlikuju, prije pravljenja modela za predikciju neophodno je obilježja normalizovati. U ovome slučaju normalizacija je vršena po sledećoj formuli:

$$X' = \frac{X}{X_{max}}$$

S obzirom da naša glavna labela Pozicija predstavlja kategoričko obilježje tj. sadrži konačan skup diskretnih vrijednosti bez međuzavisnosti između njih, ona se kao takva mora transformisati u drugi oblik da bi se prilagodila većini algoritama mašinskog učenja. Biće korištena *One Hot Encoding* transformacija koja od obilježja formira matricu nula i stavlja jedinice na mjesto za onu kolonu, čiju vrijednost posjeduje.

Daljom analizom data seta ispitane su zavisnosti među atributima. To je postignuto uz pomoć matrice korelacija na osnovu koje je utvrđeno da postoje atributi koji su u jakoj korelacionoj vezi. To predstavlja jasan razlog za izbacivanje jednog od koreliranih atributa kako bi se smanjila dimenzionalnost i složenost modela, a samim tim i smanjilo vrijeme potrebno za predikciju. Matricom korelacija je utvrđeno da su atributi GK_reflexes i GK_positioning nalaze u korelaciji, te je prvi atribut izbačen iz skupa podataka.

4. METODOLOGIJA

Algoritmi koji su korišćeni i analizirani u ovom radu su: *Naive Bayes*, *Support Vector Machines (SVM)*, *Random Forest (RF)*, *K Nearest Neighbors (KNN)*, *Multinomial Logistic Resregion (MLR)*.

Svaki klasifikator je implementiran na određeni način i nakon toga je vršena evaluacija dobijenih rezultata. Metrike evaluacije performansi klasifikatora koje su korištene u ovom radu su: *Hamming loss*, *Jaccard score*, *F1 score*.

Kao optimalni kernel za SVM algoritam je izabran *rbf*, a parametar regularizacije C je određen empirijskim putem i iznosi 1. Za RF algoritam optimalna vrijednost parametra *n_estimators*, koji predstavlja broj stabala koja će se formirati unutar algoritma, iznosi 250. Za KNN algoritam optimalna vrijednost parametra *n_neighbors* iznosi 11.

Multilabel problem će biti rešavan na 2 načina, pomoću *Classifier Chains (CC)* algoritma i *Binary Relevance (BR)* algoritma i biće prikazani rezultati u kombinaciji sa svim klasifikacionim algoritmima.

5. ANALIZA REZULTATA I DISKUSIJA

U ovom poglavlju će biti predstavljeni rezultati dobijeni primjenom svakog od ponuđenih algoritama i diskusija njihovih rezultata.

Tabela 1. Rezultati SVM algoritma

SVM	BR	CC
F1 score	85.18%	84.38%
Hamming loss	0.05	0.06
Jaccard index	83.56%	82.94%

SVM algoritam je opravdao svoju popularnost i pokazao se kao klasifikator koji je donio najbolje rezultate. Rešenje uz pomoć BR metode je imalo tačnost od 85.18%, dok je rešenje uz pomoć CC metode imalo tačnost od 84.38%. Ostvareni rezultat je gotovo jednak rezultatu u radu [3] koji iznosi 82.89%, tačnije razlika je oko 2% što je u našem slučaju zanemarivo.

Tabela 2. Rezultati Naive Bayes algoritma

Naive Bayes	BR	CC
F1 score	63.85%	62.93%
Hamming Loss	0.21	0.22
Jaccard index	52.38%	52.65%

Naive Bayes algoritam je svoju najbolju tačnost pokazao zajedno sa BR algoritmom i ona iznosi 63.85%. Nešto manju tačnost za oko 1% ima CC algoritam i ona iznosi 62.93%. Ipak ovaj algoritam nije pokazao izrazito dobre rezultate u odnosu na ostale algoritme, ali ono u čemu se on izdvaja jeste vrijeme njegovog izvršavanja koje iznosi oko 0.5 sekundi, što ga stavlja na prvu poziciju po brzini izvršavanja.

Tabela 3. Rezultati Random Forest algoritma

RF	BR	CC
F1 score	84.75%	85.06%
Hamming Loss	0.06	0.05
Jaccard index	83.17%	83.53%

Random Forest algoritam je pokazao veoma dobru tačnost prilikom predviđanja i ona iznosi 85.06% za standardne parametre u kombinaciji sa CC algoritmom. Za oko 0.30% manje rezultate nudi BR metoda, a njena tačnost jeste 84.75%. U radu [4] se ovaj algoritam pokazao kao najbolje rešenje sa tačnošću od 93%, međutim u ovom radu je broj pozicija veći od 4, što je dovelo do smanjenja tačnosti rezultata. RF algoritam je dao rezultate slične SVM algoritmu kao i u radu [3]. Ono što je bitno napomenuti jeste da ovaj algoritam ima najduže vrijeme izvršavanja i da se ono proporcionalno povećava u odnosu na broj stabala formiranih u okviru RF algoritma.

Tabela 4. Rezultati KNN algoritma

Metrike	BR	CC
F1 score	82.50%	82.51%
Hamming Loss	0.07	0.06
Jaccard index	80.67%	80.97%

KNN algoritam je dao veoma dobre rezultate u odnosu na svoju jednostavnost. Rezultati su bili veoma slični i za CC i za BR algoritam i iznosili su respektivno 82.51% i 82.50%. Neznatno malu prednost ima CC algoritam i to za 0.01%. Slično kao u radovima [1] i [3] ovaj algoritam se nije pokazao kao najbolje rešenje, ali veoma malo zaostaje u odnosu na optimalno rešenje. Ono što je interesantno za ovaj algoritam jeste relativno malo vrijeme obučavanja modela ali veliko vrijeme predviđanja koje je ponekad veće od vremena obučavanja. Analizom atributa izabrano je 50% najuticajnijih atributa koji su ostvarili veću tačnost za 0.5%.

Tabela 5. Rezultati MLR algoritma

Metrike	BR	CC
F1 score	83.16%	82.96%
Hamming Loss	0.07	0.07
Jaccard index	80.51%	80.56%

MLR klasifikacioni algoritam ponudio je veoma dobre rezultate uzimajući u obzir njegovo malo vrijeme izvršavanja, a po tom parametru se nalazi odmah iza Naive Bayes algoritma. Najveću tačnost je dao zajedno sa BR algoritmom i ona iznosi 83.16%. Nešto manju tačnost, za oko 0.20% dao je CC algoritam za rešavanje multilabel problema, a ona iznosi 82.96%. U radu [3] ostvaren je

dosta slabiji rezultat od 51.1%, ali je za evaluaciju rezultata korištena *precision* metrika, što predstavlja veoma bitnu razliku. Analizom izbora atributa ostvareni su rezultati koji nisu imali bolju tačnost.

6. ZAKLJUČAK

U ovom radu istražena je primjena algoritama mašinskog učenja u procesu klasifikacije podataka koji su prikupljeni sa interneta. Motiv za obradu ove teme prije svega nalazi se u potrebi za određivanjem za koju poziciju fudbaler ima najbolje predispozicije, što bi napravilo ogromnu olakšicu svim trenerima. Obučavajući skup podataka se sastojao iz 17981 igrača sa svojim psihičkim i fizičkim osobinama koje su procenjene na osnovi stručnjaka iz te oblasti. Procjene su vršene na osnovu svjetskog prvenstva u Rusiji 2018 godine. U ovom slučaju za klasifikaciju iskorišteni su *K-Nearest Neighbours*, *Random Forest*, *Support Vector Machine*, *Gaussian Naive Bayes* i *Multinomial Logistic Regression*. Kao tehnike za rešavanje multilabel problema korišteni su *Binary Relevace* i *Classifier Chains* algoritmi. *Classifier chains* algoritam je pokazao nešto bolju tačnost u odnosu na *Binary Relevance* algoritam.

Najbolju tačnost je pokazao SVM klasifikator u kombinaciji sa *Classifier Chains* algoritmom i ona iznosi 85.18%. Osnovna prednost ovoga rada u odnosu na druge radove jeste proširenje broja pozicija uz minimalno gubljenje na tačnosti. Mana ovoga rada jeste to što su svi parametri za igrače preuzeti sa svjetskog prvenstva u Rusiji, što može dovesti do potencijalno pogrešnih parametara jer su oni preuzeti na osnovu trenutnog stanja igrača a ne višegodišnje statistike. Ovaj rad bi mogao da se unaprijedi u više pravaca, recimo da se proširi broj pozicija na osam ili devet koje su zastupljene u današnjem fudbalskom svijetu. S obzirom da je mašinsko učenje dosta popularno u današnje vrijeme smatra se da je njegovo povezivanje sa najpopularnijim sportom ostvarilo dobar rezultat.

7. LITERATURA

- [1] N. Razali, A. Mustapha, F. A. Yatim, „Predicting Player Position for Talent Identification in Association Football”, International Research and Innovation Summit (IRIS2017) 6–7 May 2017, Melaka, Malaysia
- [2] N. Chandarana, “Football Player Position Prediction”, Towards data science, May 2019, Dostupno: <https://towardsdatascience.com> [Pristupljeno: februar 2021]
- [3] D. Schoch, “Predicting player positions”, Schocastics, Nov 2017, Dostupno: <http://blog.schochastics.net> [Pristupljeno: februar 2021]
- [4] M. Wiseman “Machine Learning using FIFA 2019”, LinkedIn, Feb 2019, Dostupno: <https://www.linkedin.com> [Pristupljeno: februar 2021]

Kratka biografija:



Dragan Škiljević rođen je u Gacku 1996. godine. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika odbranio je 2021. godine.

Kontakt: jasamdragan@gmail.com

**УТИЦАЈ СКЛАДИШТА ЕЛЕКТРИЧНЕ ЕНЕРГИЈЕ НА КРАТКЕ СПОЈЕВЕ У
НАПОЈНОЈ МРЕЖИ****INFLUENCE OF ENERGY STORAGE SYSTEMS ON SHORT CIRCUITS IN THE
SUPPLY NETWORK**

Невена Крунић, Зоран Стојановић, *Факултет техничких наука, Нови Сад*

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак сажај – У раду су описани основни принципи функционисања складишта електричне енергије. Пример њиховог рада дат је кроз модел дистрибутивне мреже, у којој је симулиран трополни кратак спој, са акцентом на LVRT функционалност.

Кључне речи: Складиште електричне енергије, Батерије, Кратак спој, LVRT функционалност

Abstract – This paper presents the basic principles of energy storage systems. An example of their work is given through a distribution network model, in which a three-phase fault is simulated, with an emphasis on LVRT functionality.

Keywords: Energy storage systems, Batteries, Short circuits, LVRT functionality

1. УВОД

Електроенергетски систем је централизован систем, који се састоји из електроенергетских извора, преносне мреже, дистрибутивне мреже и потрошача електричне енергије. Ова структура подразумева јединствени ток енергије од извора ка потрошачу, као и поузданост и стабилност рада система. Међутим, овакав систем има и доста недостатака. Један од њих је и удаљеност електрана од потрошачких подручја, што доводи до великих губитака приликом преноса електричне енергије до крајњих потрошача. Такође, ови системи доприносе емисији штетних гасова, који доводе до стварања ефекта стаклене баште, производње нуклеарног отпад, негативног утицаја на околину, итд. Велики део ових проблема могуће је решити применом обновљивих извора електричне енергије. Обновљиви извори енергије представљају неисцрпне изворе енергије из природе.

Дистрибуирани електроенергетски ресурси (ДЕР) обухватају све енергетске изворе за генерисање електричне енергије који се могу прикључити на систем изван постојећих централизованих структура [1]. Најчешће се прикључују на дистрибутивну мрежу, а могу се наћи и у оквирима самих потрошача, што помаже смањењу губитака у преносном систему и представљају децентрализовану производњу електричне енергије.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Зоран Стојановић, ванр. проф.

Производња електричне енергије из обновљивих извора је по природи непредвидива, мења се у току времена и зависи од различитих услова, као што су временски услови, доба дана, итд. Из тог разлога створила се потреба за акумулацијом електричне енергије у периодима повећане производње, а та акумулисана енергија се користи у периоду смањене производње. Нису само обновљиви извори били повод за формирање складишта електричне енергије. Један од главних разлога њиховог развоја је и неравномерна потрошња електричне енергије у току дана, њиховом употребном остварује се енергетски биланс (усклађује се неистовремена потрошња и производња електричне енергије).

У наставку рада биће описана складишта електричне енергије, LVRT функционалност, модели складишта који су коришћени у симулацији, као и резултат симулације.

2. ДИСТРИБУИРАНА СКЛАДИШТА ЕНЕРГИЈЕ

Дистрибуирана складишта енергије су управљиви извори енергије у смислу контролисане промене активне и реактивне снаге. То је и главни разлог њиховог интензивног развоја последњих година. Употребом ових складишта повећава се ефикасност, унапређује поузданост и сигурност у снабдевању енергијом, смањују се трошкови производње, олакшава се употреба обновљивих извора енергије, смањују се инвестициона улагања, врше се напајање изван мреже, итд.

Како би се у наизменичном систему извршило складиштење и чување електричне енергије на одређено време, потребно ју је претворити у други облик енергије који је погодан за складиштење, као што је електромагнетска, електрохемијска, кинетичка или потенцијална енергија. Енергија се складишти у интервалима када производња енергије надмашује њену потрошњу, а ускладиштене резерве се користе када потрошња енергије надмашује њену производњу. На овај начин се производња електричне енергије не треба драстично повећавати или смањивати према захтевима потрошње, него се одржава у равномерном односу.

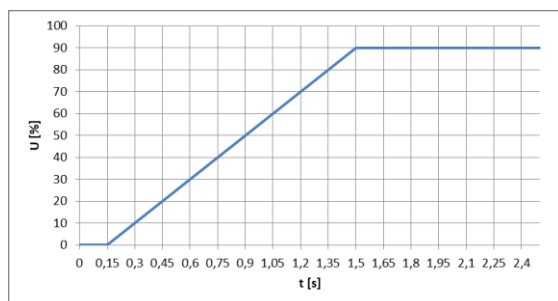
Комбиновани системи су системи са једним примарним извором енергије на улазу и две или више врста енергије на излазу. Док су хибридни системи системи са две или више врста улазних енергија и једном врстом енергије на излазу (ветро-фотонапонски системи).

Постоје различити критеријуми поделе складишта енергије. Најчешће коришћена метода поделе складишта је према облику унутрашње енергије и према том критеријуму постоје: складишта механичке енергије (потенцијална и кинетичка енергија), складишта електромагнетске енергије, складишта електрохемијске енергије и складишта топлотне енергије [2].

3. ПРАВИЛА О ПОГОНУ ДИСТРИБУТИВНИХ МРЕЖА

Приликом појаве квара у мрежи обновљиви извори, односно у овом случају батерије, имају унапред дефинисане начине реаговања. У зависности од посебних захтева који су уграђени у Правилма о погону дистрибутивних мрежа различитих земаља, инвертори могу бити подешени на различите начине. Већина развијених земаља има јасно дефинисана Правила о погону преносних и дистрибутивних мрежа. У оквиру тих правила дефинисани су и посебни захтеви „Fault Ride Through – FRT“, односно детаљније „Low Voltage Ride Through – LVRT“. Ови захтеви дефинишу услове и начин рад ДЕР приликом појаве сниженог напона на месту његовог прикључења на мрежу [3].

Као што је већ речено, већина развијених земаља има своје стриктно дефинисане захтеве FRT. У овом раду управљање ДЕР је вршено по немачком стандарду, те ће он бити и објашњен. Од ДЕР се захтева да остане прикључен на мрежу 0.15s у случају да напон на месту њиховог прикључења на мрежу падне на нулу. Ако је напон између 0% и 90% од номиналне вредности, онда граница останка ДЕР на мрежи линеарно расте између 0.15s и 1.5s. После 1.5s, ако им је напон већи од 90% од номиналне вредности, ДЕР мора да остане трајно у погону. Уколико је напон испод те граничне линије дозвољено је искључење ДЕР са мреже, након истека предвиђеног времена. С обзиром да гранична линија линеарно расте, највећи број ДЕР треба да остане прикључен на мрежу у моменту појаве квара, док после неколико милсекунди, ако квар није елиминисан, на мрежи остају само они ДЕР чији је напон на месту њиховог прикључења на мрежу изнад 90% од номиналне вредности напона. На слици 1 приказани су захтеви LVRT у правилима о раду дистрибутивних мрежа Немачке.

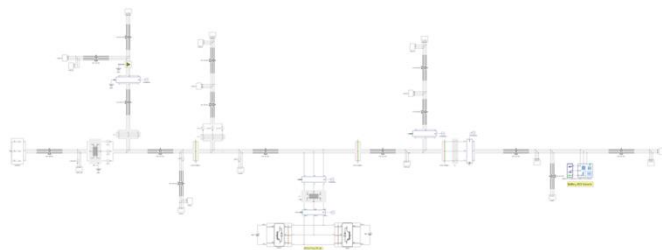


Слика 1. Захтеви LVRT у правилима о раду дистрибутивних мрежа Немачке

4. ОСНОВНИ ЕЛЕМЕНТИ МОДЕЛА ЕЛЕКТРОЕНЕРГЕТСКОГ СИСТЕМА

На слици 2 приказан је модел дистрибутивног система на којем се заснива овај рад. Основни елементи који

се јављају у приказаном систему су: напонски извор, вод, трансформатор, потрошач, складиште електричне енергије и мерни уређаји.

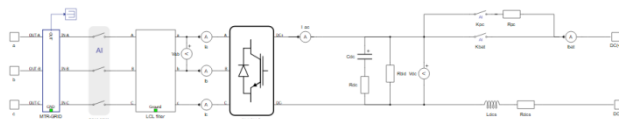


Слика 2. Приказ модела система

Како је акценат у овом раду на складиштима електричне енергије, у наставку ће само она бити детаљно објашњена. У оквиру овог рада коришћена су два типа складишта:

- EPC Battery inverter

Први тип се зове EPC Battery inverter и моделован је преко два трофазна инвертора, која нам омогућавају двосмерни проток енергије. То значи да компонента може да се понаша и као потрошач и као извор, у зависности од потреба за енергијом у систему. На слици 3 приказан је изглед једног од инвертора.



Слика 3. Изглед једног инвертора типа EPC Battery inverter

Као што се може видети, коло се састоји од трофазног прекидача који нам омогућава прикључење инвертора на мрежу, LCL филтра који се налази са AC стране претварача (служи као заштита од превеликих вредности струја и напона и за филтрирање виших хармоника), претварача који врши претварање наизменичне струје у једносмерну и обрнуто. После њега се налази једносмерни део кола који нам омогућава прикључење батерије на излазу, у којој се и врши само складиштење енергије.

Да би омогућили потпуно управљање овим дистрибуираним складиштем електричне енергије неопходно је да користимо HIL604 у комбинацији са EPC-ом. Да би управљање било одговарајуће и да би дефинисали начин понашања овог складишта електричне енергије у различитим условима, неопходно је било користити EPyQ програм. Он нам омогућава да извршимо потпуну комуникацију са HIL уређајем и EPC-ом. Сви подаци који се овде подесе, шаљу се на EPC, а он их даље прослеђује HIL-у.

Да би на коректан начин обезбедили LVRT управљање, било је неопходно да дефинишемо промену напона у функцији од времена по немачком стандарду, активне снаге у функцији од напона (VoltWatt) и реактивне снаге у функцији од напона (VoltVar).

- Battery ESS Generic

Battery ESS Generic је други тип складишта електричне енергије који је коришћен у овом раду. У оквиру генералних података можемо дефинисати

номиналну активну и привидну снагу овог складишта, номинални напон и фреквенцију. Ова компонента у оквиру себе има укључен трансформатор, како би се на најједноставнији начин прикључила на мрежу. Могуће је изабрати по ком стандарду желимо да се понаша батерија када детектује пропад напона. Овде је такође изабран немачки стандард.

5. РЕЗУЛТАТИ СИМУЛАЦИЈЕ

У оквиру рада вршена је симулација трополног кратког споја, јер је једино он симетричан и уравнотежен. Међутим, постављане су различите вредности импедансе уземљења на месту избијања кvara, чији је основни циљ да омогући различите падове напона, како на месту избијања кvara, тако и у остатку мреже. У следећем разматрању биће приказан један случај, када је импеданса на месту кvara једнака нули.

Референтна вредност активне снаге EPC Battery inverter-a је 50kW. То значи да складиште у нормалним условима ињектира активну снагу у мрежу, односно производи је и потпомаже мрежи.

Референтна вредност реактивне снаге EPC Battery inverter-a је 50kvar. Складиште у нормалним условима производи реактивну снагу и потпомаже мрежи.

Референтна вредност активне снаге Battery ESS Generic-a је 0.5г.ј. од номиналне вредности ($P_n = 300kW$), односно она је 150kW. И ово складиште у нормалним условима производи активну снагу и потпомаже мрежи.

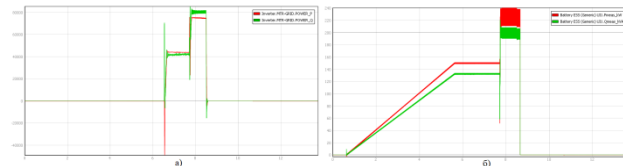
Референтна вредност реактивне снаге Battery ESS Generic-a је 0.5г.ј. То значи да она износи 50% од номиналне вредности реактивне снаге и да се производи у нормалним условим ($Q_n = 264.57kvar$), односно она је 132.29kvar.

Напон на месту прикључења складишта на мрежу током периода кvara није пао на нулу. То је последица веће удаљености места прикључења складишта на мрежу од места избијања кvara и постојања идеалног напонског извора, који наставља да напаја мрежу иако је детектован кvar у њој, као и деловања ових складишта. С обзиром да су складишта релативно близу један другом и вредности њихових напона су слични, како у нормалном режиму, тако и у периоду кvara.

На слици 4 приказане су промене активне и реактивне снаге оба складишта електричне енергије. На првој слици је приказана промена код EPC Battery inverter-a, где можемо уочити да се ово складиште после одређеног временског интервала нагло прикључује на мрежу, аутоматски иде на своје референтне вредности и њих држи све док се не детектује кvar у мрежи. Када детектује пропад напона, он креће да производи номиналну активну и реактивну снагу и у том стању остаје временски период који је прорачунао на основу вредности пропада напона на месту његовог прикључења на мрежу и жељеног LVRT стандарда. Након истека тог времена он се искључује, што се и види са слике.

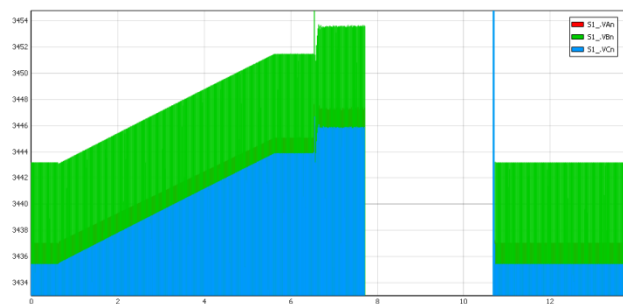
На другој слици можемо видети промену код Battery ESS Generic-a и овде можемо уочити да се он

постепено укључује, односно његове вредности активне и реактивне снаге постепено иду ка референтним вредностима, које су дефинисане на почетку ове главе. Након достизања референтних вредности, он остаје у том стању све док не детектује кvar. Када је кvar детектован, вредности ињектиране активне и реактивне снаге одлазе на одговарајуће вредности. Након истека израчунатог времена по немачком стандарду складиште се искључује.



Слика 4. Приказ промене активне и реактивне снаге на а) EPC Battery inverter-у б) Battery ESS Generic-у

На слици 5 приказана је увеличана промена напона, како би се видео утицај прикључења оба складишта електричне енергије на мрежу. На самом почетку слике видимо да је првих 0.5s напон константан и да има номиналну вредност. Након тога, он креће постепено да расте. То је последица постепеног прикључења Battery ESS Generic-a на мрежу јер он производи реактивну снагу, а самим тим утиче на мрежу тако што повећава напон. Када је Battery ESS Generic достигао своју референтну вредност снага, и напон престаје да расте све до тренутка када се EPC Battery inverter не прикључи на мрежу. Како његова вредност снага аутоматски иде на референтну вредност, тако и напон нагло расте, јер и ово складиште електричне енергије производи реактивну снагу у нормалном режиму рада. Након тога, вредност напона је константна све док не дође до кvara. У периоду кvara, напон на месту кvara је једнак нули јер је симулиран метални трополни кратак спој. Кад је кvar елиминисан, напон поново има номиналну вредност, коју је имао и на самом почетку симулације, пре прикључења складишта електричне енергије.

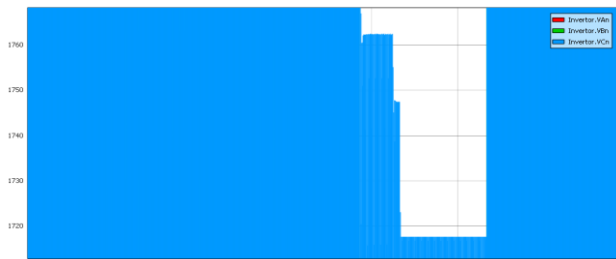


Слика 5. Приказ увеличане промене напона на месту избијања кvara

Утицај складишта електричне енергије на напон у периоду трајања кvara приказан је на слици 6. У периоду кvara могу се уочити три нивоа напона. Први ниво, који наступа одмах након избијања кvara и који је највиши, одговара периоду кад су оба складишта прикључена на мрежу. Други ниво, који је нешто нижи, одговара периоду када је EPC Battery inverter искључен са мреже, а Battery ESS Generic је још увек прикључен на мрежу. И трећи ниво одговара периоду када су оба складишта електричне енергије искључена са мреже.

Да би одредили временски интервал, колико дуго дата батерија треба да буде прикључена на мрежи спрам немачког стандарда, прво је потребно да одредимо колики пад напона се јавља у периоду квара. То можемо израчунати помоћу следеће релације:

$$u_{drop} = \frac{U_{during_fault}}{U_{nom}} * 100\% \quad (1)$$

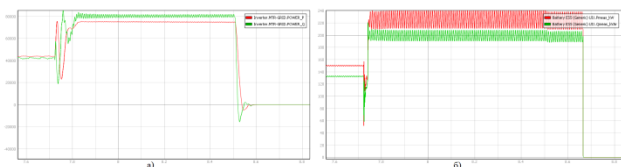


Слика 6. Увеличан приказ промене напона у периоду квара на месту прикључења EPC Battery inverter-a на мрежу

Као што се види са слике 4 и 6, складишта електричне енергије се не искључују у истом тренутку, главни разлог за то је што немају исти вид управљања. Управљање EPC Battery inverter-a је у виду каскаде и он ће често остајати дужи временски интервал прикључен на мрежу него што је за дати пропад напона предвиђено тим стандардом. Међутим, код њега се време држања прерачунава само једном и то у тренутку када напон пропадне, односно када се квар деси. Из тог разлога њему одговара ниво напона када су оба складишта искључена са мреже (1717V), односно пропад напона од $u_{drop_min} = 49.98\%$. Из тог разлога његово време држања је 0.8s. Код Battery ESS Generic-a, иако он ради по линеарној криви, он остаје нешто дуже прикључен на мрежу, јер обрачунава време држања више пута у току квара. То значи да он, када детектује пропад напона, израчуна време држања али поново мери ниво напона када уђе у LVRT режим, односно када је ниво напона подигнут као последица његовог деловања и деловања EPC Battery invertera, и спрам те вредности прерачунава нови временски интервал колико треба да буде прикључен на мрежу. Из тог разлога њему одговара ниво напона од 1762V, односно пропад напона $u_{drop_max} = 51.29\%$. Да би израчунали његов период држања неопходно је да користимо линеарну криву која одговара немачком стандарду:

$$\frac{u_{drop}}{100\%} = 0.667 * t_{hold} - 0.1 \rightarrow t_{hold} = 0.9189s \quad (2)$$

Са слике 7 можемо видети да израчуната времена држања одговарају временима држања из симулације.



Слика 7. Приказ промене снага у периоду квара на а) EPC Battery inverter-у б) Battery ESS Generic-у

6. ЗАКЉУЧАК

У овом раду укратко је објашњен принцип деловања два типа складишта електричне енергије приликом појаве трополног кратког споја у мрежи, са посебним акцентом на LVRT функционалност.

Да би се верификовало понашање складишта електричне енергије, у практичном делу рада вршена је симулација у Typhoon HIL Control Center-у. Такође, извршена је упоредна анализа EPC Battery inverter-a и Battery ESS Generic-a, како са аспекта понашања у мрежи, тако и са аспекта LVRT функционалности. Оба складишта имају своје предности и мане, али вернији стандарду је свакако Battery ESS Generic. Да би EPC Battery inverter имао што бољи начин реаговања, односно вернију дужину трајања LVRT-ја одговарајућем стандарду, потребно је да његово понашање дефинишемо са што више тачака.

7. ЛИТЕРАТУРА

- [1] В. Стрезоски, „Основи електроенергетике“, Факултет техничких наука, Нови Сад, 2015.
- [2] Paul W. C. Northrop, M.L. Crow, „Battery Energy Storage System (BESS) and Battery Management System (BMS) for Grid-Scale Applications“, Proceedings of the IEEE, 2014.
- [3] L. Strezoski, B. Dumnić, K. A. Loparo, „Novel Fault Models for Electronically Coupled Distributed Energy Resources and their Laboratory Validation“, IEEE Transactions on Power Systems, 2019.

Кратка биографија:.



Невена Крунић рођена је у Новим Саду 09.07.1996. године. 2015. године је уписала Факултет техничких наука, струдијски програм Енергетика, електроника и телекомуникације. На студијама се определила за смер Електроенергетика – Електроенергетски системи и дипломирала 2019. године. Мастер студије је уписала 2019. године.



Зоран Н. Стојановић рођен је 22.07.1979. год. у Пожаревцу. Докторску дисертацију под називом „Усмерени релеји базирани на дигиталном фазном компаратору“ одбранио је 2012. године на Електротехничком факултету Универзитета у Београду. Области научноистраживачког рада којима се до сада бавио су релејна заштита, разводна постројења и мониторинг и дијагностика високонапонских потрожења.

PREDIKCIJA BROJA INDEKSNIH POENA IGRAČA U ABA LIGI SA FOKUSOM NA PRIKUPLJANJU I EKSPLOATIVNOJ ANALIZI PODATAKA**PREDICTION OF THE NUMBER OF INDEX POINTS OF PLAYERS IN THE ABA LEAGUE WITH A FOCUS ON DATA COLLECTION AND EXPLORATORY ANALYSIS**

Miloš Nišić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Ovaj rad se bavi predikcijom broja indeksnih poena koje igrač ostvari na košarkaškoj utakmici. Fokus rada je na prikupljanju i eksplorativnoj analizi podataka. Prikupljanje podataka je vršeno sa sajta eurobasket.com pomoću tehnika web-scrapinga. Nakon sređivanja skupa podataka, ekstrakcije obeležja i eksplorativne analize podataka vršena je predikcija pomoću tri različita regresora: Lasso, Random Forest i LightGBM. Optimizacijom hiperparametara implementacija ovih algoritama došlo se do modela pomoću kojih je vršena predikcija broja indeksnih poena. Najbolje rezultate među njima pokazao je model LASSO regresije sa srednjom apsolutnom greškom $MAE = 5.617$. Izneti su predlozi za poboljšanje skupa podataka, a samim tim i za dalji razvoj ovog rešenja.

Ključne reči: Lasso regresija, Random Forest regresija, LightGBM regresor, predikcija indeksa korisnosti

Abstract – This paper presents the machine learning approach to automatically predict the number of index points that a player achieves in a basketball game. The focus of the work is on data collection and exploratory analysis. Data was collected from eurobasket.com using web-scraping techniques. After cleaning the data set, feature extraction and exploratory data analysis were performed. The prediction was made using three different regressors: Lasso, Random Forest, and LightGBM. By optimizing the hyperparameters of these algorithms' implementations, we came up with models that were used to predict the number of index points. The LASSO regression model achieved the best results with the mean absolute error $MAE = 5.617$. The paper proposes further improvements of the data set as future work.

Keywords: Lasso regression, Random Forest regression, LightGBM regressor, Personal index rating predictions

1. UVOD

Kada je sport nastajao, njegov osnovni cilj bilo je takmičenje i zabava. Međutim, vremenom, sport je prerastao u unosan biznis i veliku industriju. Jedan od najzastupljenijih sportova danas, u koji se ulažu enormne sume novca, jeste košarka. Prosečna vrednost timova u

NBA (engl. *National Basketball Association*) ligi je 2020. godine bila preko dve milijarde dolara [1]. Kao dodatni vid zabave za ljubitelje sporta osmišljene su razne igre na sreću, kao i sve popularnije fantazi (engl. *fantasy*) igre.

ABA liga, kao najveće i najpoznatije regionalno sportsko takmičenje, je takođe pokrenula svoju fantazi igru. S obzirom na to nastala je ideja primene mašinskog učenja u svrhe predikcije broja indeksnih poena koje će igrač osvojiti na utakmici čime bi se olakšalo takmičenje u fantazi ligi.

Kao osnovni parametar prilikom bodovanja učinka igrača za ovu fantazi igru koristi se broj indeksnih poena koje igrač ostvari na utakmici. U ovom radu vršena je predikcija ovog košarkaškog statističkog parametra. U ovom radu je prikazan postupak prikupljanja i eksplorativne analize podataka neophodnih za treniranje modela mašinskog učenja. Prikupljanje podataka je vršeno sa sajta eurobasket.com pomoću tehnika web-scrapinga. Nakon sređivanja skupa podataka, ekstrakcije obeležja i eksplorativne analize podataka vršena je optimizacija različitih regresionih modela. Kao najbolji model se pokazao model LASSO regresije koji je postigao srednju apsolutnu grešku od 5.617.

2. PRETHODNA REŠENJA

Još od devedesetih godina prošlog veka ljudi su počeli ozbiljnije da se bave analizom statističkih podataka u košarci. U literaturi postoji mnogo radova koji se bave košarkaškom statistikom. Mnoge knjige, radovi i naučni članci u časopisima fokusirani su na uticaje različitih parametara na individualnu i timsku statistiku [2, 3, 4, 5]. Takođe, prethodna istraživanja bave se predikcijama mnogih drugih stvari koje se tiču sporta, kao što su različite mere učinka igrača na utakmici ili pak čitavoj karijeri [6, 7], a takođe i predikcijama uspešnosti timova [8] i ishoda utakmica [9, 10].

Problem rešavan u ovom radu je da se na osnovu statističkih podataka iz prošlosti pronadu ili kreiraju parametri koji mogu pomoći u predviđanju igračevog učinka. Postoji dosta sličnih radova na ovu temu, što u košarci [6, 7, 10], što u nekim drugim sportovima [11, 12], ali nijedan koji se konkretno bavi ABA ligom.

Različitim merama učinka igrača najviše se bave autori rada [3], ali više informativno, dajući formule za svaku od

NAPOMENA:

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

navedenih metrika i objašnjavajući parametre koji se koriste u tim formulama. Oni se takođe bave i različitim parametrima koji utiču na predikciju, između ostalih ofanzivnim i defanzivnim rejtingom, efektivnim i pravim procentom šuta, tempom kojim igraju timovi i procentom uspešnosti skokova.

Rad koji se najviše bavi obeležjima i na koji se najviše referenciraju drugi radovi je [7]. U tom radu predikuje se broj poena, za koje kažu da su tradicionalno dominantna mera učinka igrača na utakmici, ali ne i najtačnija, a takođe se pominju i druge mere koje su pominjane u radu [3] i koje su prednosti, odnosno mane jednih i drugih.

3. METODOLOGIJA I ALATI

U ovom poglavlju predstavljena je arhitektura rešenja, kako se došlo do skupa podataka, kako je on izgledao, a predstavljena je i izgradnja modela, kao i evaluacija.

3.1. Arhitektura rešenja

Arhitektura rešenja sastoji se iz četiri dela i čine je:

1. Kreiranje skupa podataka
2. Ekstrakcija obeležja
3. Izgradnja modela
4. Evaluacija modela

Ulaz sistema predstavljaju podaci sa utakmica koji su prikupljeni sa *eurbasket.com* sajta, odnosno obeležja koja su dobijena iz prikupljenih podataka i čine ih prosečne vrednosti ostvarenih učinaka koje su igrač, njegova i protivnička ekipa ostvarile na prethodne 3 i na prethodnih 5 utakmica, dok izlaz sistema predstavlja predikciju indeksa korisnosti koji je igrač ostvario na utakmici.

3.2. Kreiranje skupa podataka

Kreiranje skupa podataka je vršeno sa sajta *eurobasket.com* tehnikama *web-scraping*-a. Preuzeti podaci su sređivani, odnosno čišćeni od nedostajućih vrednosti, pogrešno unetih podataka, itd. U ovom koraku vršeno je oblikovanje skupa podataka, odnosno organizovanje u oblik pogodan za ekstrakciju obeležja i kasnije korišćenje u modelima.

Za prikupljanje podataka korišćena je *BeautifulSoup* biblioteka [14] i *Selenium Web Driver* [15]. *Selenium Web Driver* se koristio za automatski pristup stranicama sa podacima, nakon čega je *BeautifulSoup* parsirao *Hypertext Markup Language (HTML)* dokument.

3.3. Ekstrakcija obeležja

U delu ekstrakcije obeležja vršena je transformacija “sirovih” podataka u podatke koji se mogu analizirati i koristiti tako da generišu validna i upotrebljiva znanja. Ovaj korak čine prvenstveno razumevanje i istraživanje podataka.

Na osnovu postojećih obeležja kreirana su nova obeležja, koja predstavljaju naprednu košarkašku statistiku i koja su upotrebljavana u drugim radovima.

Za predikciju indeksa korisnosti igrača na nekoj utakmici ne mogu se koristiti podaci sa te utakmice. Zbog toga sledeći problem bio je dobiti istorijski relevantne podatke za utakmicu za koju se predviđa indeks korisnosti.

Zbog toga su za svaku kategoriju kreirane dve nove koje su se odnosile na ostvaren prosek u određenoj statističkoj kategoriji na prethodne 3 i na prethodnih 5 utakmica. Na osnovu ovako dobijenih obeležja, kao i drugih koja su se ticala karakteristika igrača, vršila se predikcija.

Nakon kreiranja obeležja vršena je i eksplorativna analiza podataka uz pomoć matrica korelacije i toplotnih mapa. Za pregled najuticajnih obeležja na predikciju indeksa korisnosti, pre nego što nam sami modeli pokažu, korišćena je i *SelectKBest*¹ klasa koju nudi *Scikit-learn* [16] biblioteka.

3.4. Izgradnja modela

Bez kvalitetnih podataka nema ni kvalitetnih rešenja, međutim, odabir modela predstavlja takođe jako bitan deo ovog sistema. Ovaj segment rada predstavlja odabir algoritma mašinskog učenja, kao i odabir hiperparametara koji se koriste tokom treniranja nekih od modela. U ovom radu razmatrani su i upotrebljeni modeli *LASSO* regresije, *Random Forest* regresije, kao i *Light Gradient Boosting Machine (LightGBM)* model.

Optimizacija hiperparametara modela predstavlja problem pronalaska optimalnih parametara za algoritme učenja. Od modela do modela zavisi koliko oni iziskuju ovu operaciju. *LASSO* regresija i *Random Forest* algoritam sadrže mnogo manje parametara koje je potrebno podesiti od *LightGBM* modela.

Za *LASSO* regresiju odabir *alfa* vrednosti urađen je odabirom najboljeg rezultata prilikom unakrsne validacije, pri čemu je opseg vrednosti automatski odredio sam algoritam, odnosno njegova implementacija iz *Scikit-learn* biblioteke.

Za pronalaženje hiperparametara za *Random Forest* algoritam korišćena je nasumična pretraga, uz pomoć *RandomizedSearchCV*, što takođe predstavlja *Scikit-learn* implementaciju ovog načina pretrage.

Hiperparametri i opsezi vrednost iz kojih su birani mogu da se vide u tabeli 1.

¹ *SelectKBest* rangira obeležja koristeći prosleđenu funkciju mere, u ovom slučaju *f_regression* [17], a zatim uklanja sve osim prvih *k* najbolje rangiranih obeležja. *f_regression* je linearni model za ispitivanje individualnog efekta svakog od mnogih regresora. Ovo je funkcija bodovanja koja se koristi u postupku odabira obeležja.

Tabela 1: Hiperparametri i njihov opseg za *Random Forest* algoritam

Hiperparametri	Opseg vrednosti
n_estimators	10 vrednosti iz opsega [1, 2000]
max_features	['auto', 'sqrt']
max_depth	11 vrednosti iz opsega [10, 110]
min_samples_split	[2, 5, 10]
min_samples_leaf	[1, 2, 4]
bootstrap	[True, False]

Za *LightGBM* algoritam korišćena je Bajesova optimizacija hiperparametara uz pomoć *HyperOpt* biblioteke [18]. Hiperparametri koji su korišćeni i opsezi vrednosti iz kojih su birani mogu da se vide u tabeli 2.

Tabela 2: Hiperparametri i njihov opseg za *LightGBM* algoritam.

Hiperparametri	Opseg vrednosti
n_estimators	10 vrednosti iz opsega [1, 2000]
max_depth	11 vrednosti iz opsega [10, 110]
num_leaves	6 vrednosti iz opsega [100, 1300]
subsample	5 vrednosti iz opsega [0.1, 1]
colsample_bytree	5 vrednosti iz opsega [0.5, 1]
learning_rate	1 vrednosti iz opsega [0.1, 1]
reg_alpha	6 vrednosti iz opsega [0.1, 0.6]
reg_lambda	6 vrednosti iz opsega [0.1, 0.6]
min_child_samples	4 vrednosti iz opsega [0.1, 100]
verbose	-1

Nakon optimizacije hiperparametara korišćenih modela, vršena je evaluacija na prethodno izdvojenom test skupu i njihovo poređenje, kao i poređenje sa rezultatima sličnih radova.

4. REZULTATI I DISKUSIJA

Konačan skup podataka *ABA 2010-2019* sadrži 23 677 instanci. Za svrhe primene algoritama mašinskog učenja, podeljen je na trening i test skup, tako što trening skup čini 80%, a test skup 20% od ukupnog broja instanci. Odnosno, trening skup sadrži 18 940 instanci, dok test skup ima ukupno 4 736 instanci.

Za evaluaciju modela korišćena je srednja apsolutna greška (engl. *MAE* – *Mean Absolute Error*). Ova greška predstavlja apsolutnu razliku između ciljne vrednosti i vrednosti predviđene modelom.

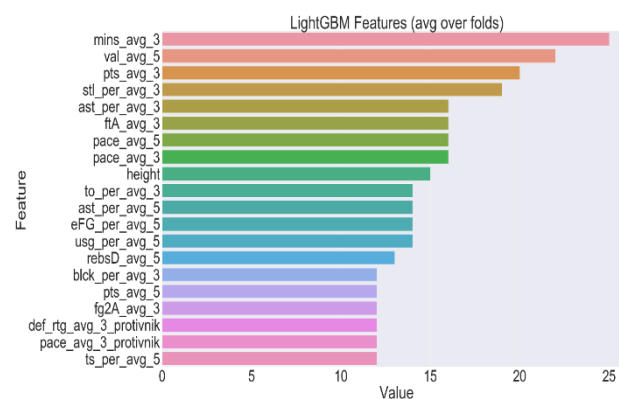
Najbolje rezultate među korišćenim algoritmima pokazao je model *LASSO* regresije sa srednjom apsolutnom greškom *MAE* = 5.617. Rezultati svih modela mogu da se vide u tabeli 3.

Kao najbitnija obeležja koja isticala su se prosečan broj indeksnih poena na prethodnih 3 i 5 utakmica, prosečan broj poena na prethodnih 3 i 5 utakmica, prosečan broj odigranih minuta na prethodnih 3 i 5 utakmica, kao i procenat upotrebe na prethodnih 3 i 5 utakmica.

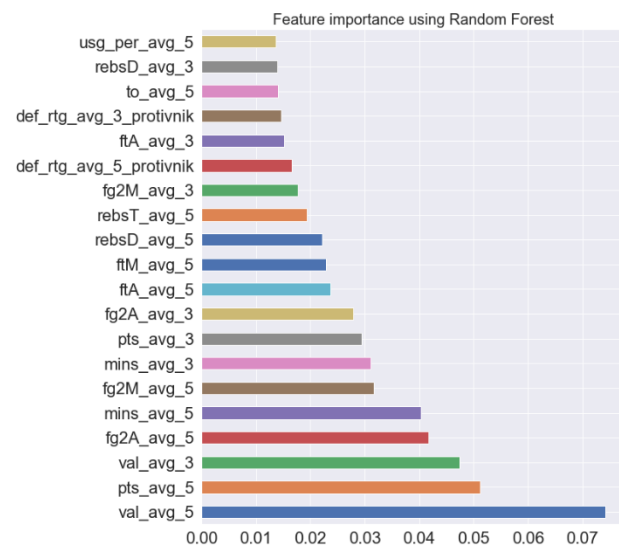
Na slici 1 mogu da se vide najbitnija obeležja koja je izdvojio *LightGBM* model, a na slici 2 ona koja je izdvojio model *Random Forest*.

Tabela 3: Poređenje rezultata upotrebljenih modela.

Model	MAE trening	MAE test	Opis
Linearna regresija/Lasso		5,617	Sa laso selekcijom obeležja
<i>LightGBM</i>	5,72	5,669	Sa optimalnim hiperparametrima
<i>Random Forest</i>		5,882	Sa standardno podešenih hiperparametrima
<i>Random Forest</i>	5,708	5,625	Sa optimalnim hiperparametrima



Slika 1: 20 najuticajnijih obeležja koje je odabrao *LightGBM* model.



Slika 2: Najuticajnija obeležja koja je odabrao *Random Forest* algoritam.

U radu [13] najbolji rezultat je dobijen uz pomoć neuronskih mreža, pri čemu je srednja apsolutna greška iznosila $MAE = 6.8805$.

5. ZAKLJUČAK

Zadatak ovog rada je predikcija indeksa korisnosti košarkaša za ABA ligu. Indeks korisnosti predstavlja jednu od mera učinka igrača na utakmici i kao takva služi za bodovanje u fantazi igri koju organizuje ABA liga. Glavni problem u radu bilo je prikupljanje podataka, kreiranje odgovarajućeg skupa podataka, njihovo analiziranje, kao i kreiranje naprednih košarkaških statističkih kategorija, sve u cilju bolje predikcije broja indeksnih poena.

Rezultati koji su ostvareni su sasvim zadovoljavajući. Međutim, za dalji rad i unapređenje do sada urađenog najbitnije bi bilo proširenje skupa podataka ABA 2010-2019, kao i pronalazak i kreiranje dodatnih obeležja, odnosno njihovo detaljno analiziranje kakav uticaj imaju na igru, odnosno na kreiranje indeksa korisnosti. Da bi se došlo do ovoga potrebno je i da se sama liga više pozabavi domenom statistike.

6. LITERATURA

- [1] NBA Team Values 2020, <https://www.forbes.com/sites/kurtbadenhausen/2020/02/11/nba-team-values-2020-lakers-and-warriors-join-knicks-in-rarefied-4-billion-club/#2d8633322032>
- [2] Oliver, Dean. Basketball on paper: rules and tools for performance analysis. Potomac Books, Inc., 2004.
- [3] Kubatko, Justin, et al. "A starting point for analyzing basketball statistics." Journal of Quantitative Analysis in Sports 3.3 (2007).
- [4] Page, Garritt L., and Fernando A. Quintana. "Predictions based on the clustering of heterogeneous functions via shape and subject-specific covariates." Bayesian Analysis 10.2 (2015): 379-410.
- [5] South, Charles, et al. "A Starting Point for Navigating the World of Daily Fantasy Basketball." The American Statistician 73.2 (2019): 179-185.
- [6] FiveThirtyEight's Career-Arc Regression Model Estimator with Local Optimization, <https://projects.fivethirtyeight.com/carmelo/>
- [7] Casals, Martí, and A. Jose Martinez. "Modelling player performance in basketball through mixed models." International Journal of Performance Analysis in Sport 13.1 (2013): 64-82.
- [8] Cai W, Yu D, Wu Z, Du X, Zhou T. A hybrid ensemble learning framework for basketball outcomes prediction. Physica A: Statistical Mechanics and its Applications. 2019 Aug 15;528:121461.
- [9] Thabtah F, Zhang L, Abdelhamid N. NBA game result prediction using feature analysis and machine learning. Annals of Data Science. 2019 Mar 7;6(1):103-16.
- [10] Shreyas S. Shivakumar. "Learning to Turn Fantasy Basketball Into Real Money." https://shreyasskandan.github.io/Old_Website/files/report-ChanHuShivakumar.pdf
- [11] Porter, Jack W. "Predictive Analytics for Fantasy Football: Predicting Player Performance Across the NFL." (2018).
- [12] Lutz, Roman. "Fantasy football prediction." arXiv preprint arXiv:1505.06918 (2015).
- [13] Kengo Arai, <https://github.com/KengoA/fantasy-basketball/blob/master/report.pdf>
- [14] Beautiful soup, <https://www.crummy.com/software/BeautifulSoup/b4/doc/>
- [15] Selenium Web Driver, <https://www.selenium.dev/documentation/en/>
- [16] Scikit-learn, <https://scikit-learn.org/>
- [17] f_regression, https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_regression.html#sklearn.feature_selection.f_regression
- [18] Hyperopt, <https://github.com/hyperopt/hyperopt>

Kratka biografija



Miloš Nišić rođen je 1994. godine u Novom Sadu. Osnovnu školu "Desanka Maksimović" je završio u Futogu 2009. godine. Gimnaziju "Isidora Sekulić" u Novom Sadu je završio 2013. godine. Iste godine je upisao Fakultet tehničkih nauka, odsek Računarstvo i automatika. Zvanje diplomirani inženjer elektrotehnike i računarstva je stekao 2018. godine. Master akademske studije je upisao 2018. godine na istom studijskom programu na Fakultetu tehničkih nauka.

kontakt: milosnisic94@gmail.com

SISTEM ZA UPRAVLJANJE DISTRIBUIRANIM IZVORIMA ENERGIJE OD STRANE KRAJNJIH POTROŠAČA**SYSTEM FOR MANAGING DISTRIBUTED ENERGY RESOURCES BY END CONSUMERS**

Filip Filipović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Sa razvojem tehnologije dolazi do nastanka distribuiranih izvora energije (DER) što dovodi do potrebe za nastankom aplikacija za upravljanje i nadzor koje se nazivaju distribuirani sistemi za upravljanje resursima (DERMS). Sa padom cene distribuiranih izvora energije u koje spadaju solarni paneli i baterije dolazi do nastanka prosumer-a. Prosumer predstavlja potrošača koji poseduje DER, što znači da je to potrošač koji u isto vreme i proizvodi i troši električnu energiju. Ovaj rad je baziran na aplikaciji za upravljanje solarnih panela i baterija od strane prosumer-a. Korisnik aplikacije nakon prijavljivanja na servis ima pristup informacijama o trenutnom radu solarnih panela i baterija kao i o trenutnoj potrošnji. Korisnik takođe može da ima pristup istorijskim podacima rada kao i da izvršava manuelno komandovanje solarnih panela ili da pokrene postupak optimizacije sa ciljem ostvarenja što boljih rezultata rada solarnih panela i baterije.

Ključne reči: Distribuirani sistem za upravljanje resursima, distribuirani izvori energije, Prosumer-i

Abstract – With the development of technology comes the emergence of distributed energy sources (DER) which leads to the need for the emergence of management and monitoring applications called distributed resource management systems (DERMS). With the fall in the price of distributed energy sources, which include solar panels and batteries, prosumers are emerging. Prosumer represents a consumer who owns DER, which means that it is a consumer who at the same time produces and consumes electricity. This paper is based on an application for control of solar panels and batteries by prosumer. After logging in to the service, the user of the application has access to information on the current operation of solar panels and batteries as well as on current consumption. The user can also have access to historical data as well as perform manual commands of solar panels or start the optimization process in order to achieve the best possible operation of solar panels and batteries.

Keywords: Distributed energy management systems, Distributed energy resources, Prosumers

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Bošković, docent.

1. UVOD

Elektroenergetski sistem (EES) predstavlja najveći i najkompleksniji infrastrukturni sistem na svetu. Sastoji se od velikog broja komponenti i učesnika što čini održavanje i upravljanje veoma kompleksnim.

DER predstavlja distribuirane generatore električne energije, od kojih su najčešći solarni paneli, vetrogeneratori i baterije. Iako su DER-ovi obnovljivi izvori energije i dalje mogu da imaju negativan uticaj na životnu sredinu.

Tako da na primer vetrogeneratori prilikom rada proizvode veliku buku koja ometa kretanje ptica, ili solarni paneli koji zauzimaju velike površine. Takođe je potrebno da se obezbedi adekvatan način za odlaganje pokvarenih baterija koje mogu da imaju negativan uticaj na životnu sredinu. Sa druge strane distribucijom DER-ova može da se reguliše pritisak na životnu sredinu.

Postoje razne tehnologije, polise i korisnički doprinosi koji značajno utiču na razvoj i implementaciju DER-ova unutar EES. To vodi ka pozitivnom ekonomskom i ekološkom razvoju.

Prednosti upotrebe DER-ova:

- Smanjenje operativnih troškova,
- Pobojšanje pouzdanosti,
- Smanjenje ugljenika,
- Dodatni izvori prihoda,
- Omogućava elektrifikaciju vozila.

U nastavku ovog rada, u poglavlju 2, sledi objašnjenje implementacije i korišćenja distribuiranog sistema za upravljanje resursima (DERMS). U poglavlju 4 biće opisana funkcionalnost aplikacije na kojoj se ovaj rad bazira. Dok u poglavlju 3 biće opisane tehnologije koje su korišćene prilikom implementacije. Zaključak istraživanja ovog rada dat je u poglavlju 6.

2. DISTRIBUIRANI SISTEM ZA UPRAVLJANJE RESURSIMA (DERMS)

Kompanije i mrežni operateri se suočavaju sa izazovima upravljanja različitim DER-ova, i time dolazi do potrebe za razvojem novih tehnologija i poslovnih modela.

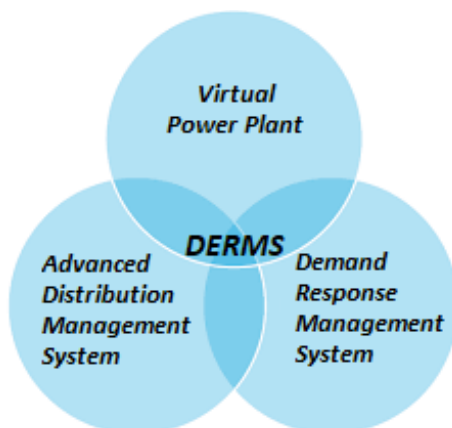
Tehnologije koje su dizajnirane za upravljanje sve raznovrsnijim potencijalom DER uređaja, koji su integrisani u distributivnoj mreži, kreću se od hardverskih uređaja pa do softvera koji dobijaju informacije od sofisticiranih alata za analizu podataka. Kako se primena DER-ova povećava tokom vremena, fokus tehnologija za upravljanje se okreće više prema softveru.

Elektroprivreda i drugi pružaoci energetske usluge koriste DERMS kako bi upravljali različitim DER-ovima u jednom sistemu.

Postoji dosta različitih tehnologija koja su dobila ime distributed energy management systems (DERMS). Iz tog razloga mogu da se pojave tehnologije pod imenom

DERMS koje zapravo ne predstavljaju konkurenciju jedna drugoj.

Slika 1 pokazuje kako različiti sistemi različitih proizvođača konvergiraju u DERMS. DERMS možda ima sve funkcionalnosti kao i ostali sistemi, ali proizvođači imaju drugačiji pristup datim funkcionalnostima.



Slika 1. Konvergencija ka DERMS sa drugačijim pristupom

DERMS bi trebalo da nudi programsko rešenje koje je okrenuto korisnicima. Korisnici bi trebali da se uloguju na portal koji im pruža uvid u rad njihovih DER-ova, uvid u potrošnju električne energije kao i informacije o naplati i tarifama. Korisnicima se takođe pruža mogućnost potpisivanja ugovora sa mrežnim provajderom, kako bi mogao da vrši prodaju električne energije. Tako da DERMS treba da ima mogućnost prikazivanja informacija o predatoj električnoj energiji u mrežu kao i informacije o zaradi za prodatu električnu energiju.

3. TECHNOLOGY STACK

Softversko rešenje na kojem je ovaj rad baziran je implementirano upotrebom C# programskog jezika u Visual Studio 2019 okruženju. Softversko rešenje je veb aplikacija pisana u ASP.NET Core framework-u koji je naslednik od ASP.NET framework-a.

3.1 ASP.NET Core

ASP.NET Core je web framework i naslednik ASP.NET-a. Razvijen je od strane Microsoft-a. Do verzije 3 je podržavao .NET Framework ali sada podržava samo .NET Core. Framework spaja do sada razdvojene ASP.NET MVC i ASP.NET Web API aplikacije u jedan programski model. I ako je novi framework ima dosta kompatibilnosti sa ASP.NET. ASP.NET Core podržavaju mogućnost da različite aplikacije, pokrenute na istoj mašini, mogu da gađaju različite verzije ASP.NET Core, što nije bilo moguće kod ASP.NET.

Karakteristike ASP.NET Core:

- Vrš automatsko kompajliranje,
- Ima podršku za NuGet pakete,
- Cloud-optimized runtime,
- Generalizovan je putem Open Web Interface for .NET,
- Ujedinjuje veb UI i veb API,
- Okruženje spremno za implementaciju na Cloud,
- Light-weight i modular HTTP request pipeline,

- Pruža mogućnost pokretanja aplikacija na različitim platformama (Windows, MacOS i Linux),
- Open-source,
- Ugrađena podrška za dependency Injection.

3.3 .NET Core vs .NET Framework

.NET Framework je prvi softverski framework predstavljen od strane Microsoft-a, i mogao je da se koristi samo na Windows operativnom sistemu. .NET Framework pomaže pri razvoju i pokretanju aplikacija. Jedna od mana .NET Framework je u tome što ne podržava doprinose programerske zajednice.

.NET Core nova verzija framework-a koja otkloni ograničenja .NET Framework-a. .NET Core može da se koristi ne samo na Windows operativnom sistemu nego i na Linux i MacOS. Takođe podržava doprinose programerske zajednice.

Dobra strana oba framework-a je u tome što mogu da dele kod i komponente kada je to potrebno. .NET Standard je biblioteka koja se koristi kada je potrebno napisati kod koji je potreban i .NET Core-u i .NET Framework-u.

Takođe treba dodati da je .NET Core nastao iz potpune rekonstrukcije .NET Framework-a koji je imao dosta "legacy" koda, koji ga je usporavao u razvoju.

3.4 Veb aplikacija

Veb aplikacija je softver koji se pokreće na veb servisu, za razliku od kompjuterskih programa koji se pokreću na operativnom sistemu. Veb aplikacijama može da se pristupi preko veb pretraživača od strane svakog korisnika koji ima internet konekciju. Primeri nekih veb aplikacija su Gmail, Yahoo!, Facebook, Erste Bank Net Banking.

ASP.NET Core predstavlja tehnologiju koja se razvila kako bi moglo da se vrši kordinisanje između tehnologija klijentske strane i serverske strane.

4. FUNKCIONALNOST APLIKACIJE

Aplikacija koja se opisuje u ovom dokumentu je namenjena za upravljanje distribuiranim izvorima energije od strane prosumer-a. Distribuirani izvori energije koji se mogu pojaviti u aplikaciji su solarni paneli i baterije.

Aplikacija je realizovana kao veb aplikacija, koja se sastoji iz veb API glavnog servera i veb API simulatora kao i veb UI-ja (korisnički interfejs) preko kojeg korisnici mogu da vrše interakciju sa aplikacijom.

Aplikacija je implementirana kao ASP.NET Core Web Application, za implementaciju glavnog servera i simulatora je korišćen ASP.NET Core Web API šablon dok je za implementaciju korisničkog interfejsa korišćen ASP.NET Core Web App (Model-View-Controller) šablon.

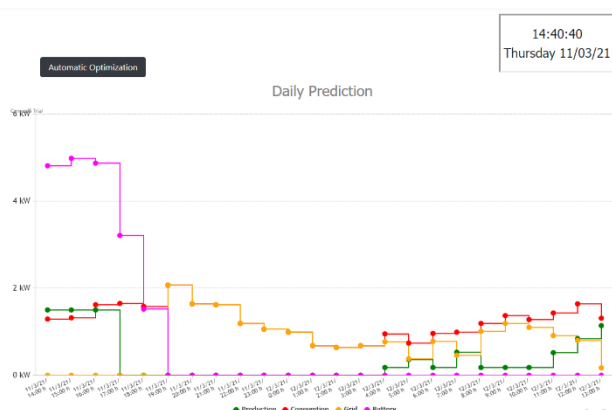
Preko aplikacije korisnici mogu da vide informacije o trenutnoj proizvodnji, potrošnji, količini struje u baterijama i energiji koja se predaje ili uzima iz mreže. Ima prozor u kojem mogu da se prikazuju istorijske informacije rada. Informacije mogu da budu na dnevnom, mesečnom ili godišnjem nivou.

Korisnik takođe ima prikaz o ukupnoj proizvedenoj, potrošenoj i predatoj energiji u mrežu u toku tekućeg meseca. Korisnik može da kontroliše manuelno rad solarnih panela, gde može da podešava limit proizvodnje solarnog panela, kao i da ga u potpunosti isključi. Takođe može da pozove komandu za automatsku optimizaciju gde će se optimizovati rad solarnih panela i baterije sa ciljem smanjenja potrebe uzimanja električne energije iz mreže.

Takođe postoji ekran sa kog korisnik može da vidi druge korisnike koji pripadaju istoj Area of responsibility i sa kog može da vidi trenutnu proizvodnju, potrošnju, energiju u baterijama i količinu predate ili uzete energije iz mreže.

5. REZULTATI

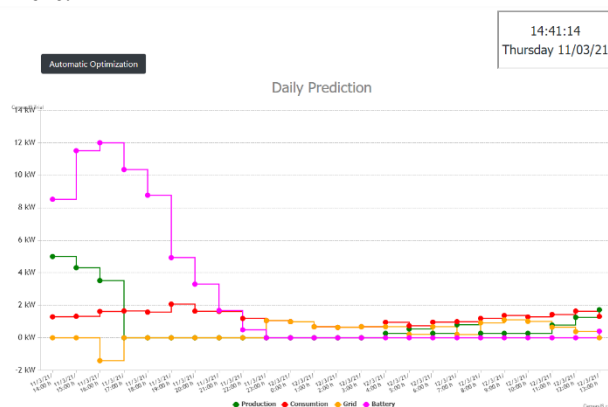
U ovom poglavlju će se predstaviti rezultati rada aplikacije prilikom automatske optimizacije.



Slika 2. Predikcija rada pre pokretanja automatske optimizacije

Na slici 2 možemo da vidimo kakvo je stanje predikcije rada pre nego što se pokrene automatska optimizacija. Sa slike vidimo da u periodu od 17 časova do 4 časa narednog dana nemamo proizvodnje jer je u tom periodu sunce zašlo i solarni paneli ne mogu da prizvode električnu energiju. Sa slike 2 takođe možemo da vidimo

da baterija nije u potpunosti napunjena i da u 19 časova dolazi do potrebe za uzimanjem električne energije iz mreže.



Slika 3. Predikcija rada nakon automatske optimizacije

Na slici 3 možemo da vidimo predikciju rada nakon optimizacije. Kao i na slici 2 nema proizvodnje u periodu od 17 časova do 4 časa narednog dana zbog zalaska sunca, ali sada vidimo da je baterija u potpunosti napunjena. Za razliku od slike 2 sada možemo da vidimo da je potrošnja bila zadovoljena iz baterije sve do 23 časa.

Razlog promene predikcije je u promeni ograničenja koje je postavljeno na solarne panele kao maksimalna količina energije koja sme da se proizvede u satu. Takođe, u slučaju da je potrebno, dolazi do pomeranja ograničenja energije koja sme da se unese u bateriju u toku sata.

Iz ovog testa možemo da vidimo kako automatska optimizacija teži tome da smanji potrebu korisnika da uzima električnu energiju iz mreže i pobojša iskorišćenost DER-ova koje korisnik poseduje.

6. ZAKLJUČAK

Sa razvojem društva i podizanjem životnog standarda dolazi do sve veće potražnje za električnom energijom. Kako proizvodnja električne energije predstavlja postupak koji značajno utiče na životnu sredinu, dolazi do potrebe za otkrivanjem novih tehnika i tehnologija za generisanje električne energije sa ciljem smanjenja negativnih uticaja po životnu sredinu.

Proizvodnja električne energije se sve više okreće upotrebi obnovljivih izvora energije. I ako predstavljaju pobojšanje upotreba obnovljivih izvora energije i dalje može da ima negativnih posledica po životnu sredinu. Rad vetrogeneratora stvara buku koja može da utiče negativno na faunu, takođe zbog svoje veličine mogu negativno da utiču na migraciju ptica. Upotreba solarnih panela stvara potrebu za velikim područjem gde bi se solarni paneli mogli postaviti, što može da dovede do potrebe za sečom šuma. Takođe mora da se nađe adekvatan način za odlaganje starih baterija koje mogu biti jako štetne za životnu sredinu ukoliko se sa njima ne postupa na pravilan način.

Solarni paneli, vetrogeneratori, baterije predstavljaju distribuirane izvore energije koji mogu da se instaliraju blizu samih potrošača. Sa razvojem tehnologije dolazi do pada u ceni što stvara novčano isplativu investiciju gde firme, preduzeća i krajnji potrošači kupuju svoje distribuirane izvore energije, i sa čijom upotrebom mogu da smanje svoje mesečne troškove električne energije.

Sa distribucijom proizvodnje dolazi do smanjenja pritiska na životnu sredinu. Takođe dolazi do potrebe za nastankom sistema koji će da rukuje i upravlja tim distribuiranim izvorima energije kako bi se očuvao balans u sistemu. I iz tog razloga nastaje aplikacija distribuirani sistemi za upravljanje resursima (DERMS).

Ukoliko krajnji potrošač sklopi ugovor sa elektrodistribucijom pruža mu se mogućnost prodaje viška električne energije.

Cilj rada je bio razvijanje aplikacije za upravljanje solarnih panela i baterija od strane krajnjeg potrošača (prosumer). Krajnjim potrošačima koji poseduju DER-ove je potrebna aplikacija na kojoj mogu da imaju uvid u rad svojih DER-ova. Takođe ukoliko njihovi DER-ovi nisu priključeni na neki distribuirani sistem za upravljanje resursima koji vrši kontrolu rada na globalnom nivou, putem ove aplikacije treba da im se obezbedi funkcija pomoću koje mogu da upravljaju svojim solarnim panelima i baterijama.

Korisnik aplikacije ima mogućnost maueknog upravljanja solarnim panelima time što može da podešava maksilanu proizvodnju solarnog panela. Takođe može da pozove funkciju za automatsku optimizaciju rada solarnih panela i baterija. Cilj optimizacije je da se obezbedi što efikasniji rad DER-ova sa ciljem što manjeg uzimanja električne energije iz mreže. Sa ovom optimizacijom se postiže i jedan od glavnih ciljeva instalacije distribuiranih izvora energije a to je smanjivanje troškova naplate električne energije na kraju meseca.

Jedan od narednih koraka u razvoju ove aplikacije bi mogla da bude podrška za druge vrse distribuiranih izvora energije, kao što su vetrogeneratori. Dodavanje različitih režima optimizacije kao što je režim za očuvanje životnog veka DER-ova ili režim punjenja baterije u trenucima kada je cena električne energije na tržištu niska takođe mogu da predstave razvojni put kojim ova aplikacija može da pođe.

7. LITERATURA

- [1] John Twidell, Tony Weir: Renewable Energy Resources, Third Edition, 2015.
- [2] D. Yogi Goswami, Frank Kreith: Energy Efficiency and Renewable Energy Handbook, Second Edition, 2016.
- [3] Roberto Rodriguez Labastida, Peter Asmus, Brett Feldman: Navigant Research, Distributed Energy Resources Management Systems, 2018.
- [4] docs.microsoft.com: Visual Studio 2019 Release Notes, 2021.
- [5] Flanagan David: JavaScript – The definitive guide (7 ed.), 2020.

Kratka biografija:



Filip Filipović rođen je 1996. godine u Novom Sadu. Nakon završetka srednje škole 2015. godine, Fakultet tehničkih nauka upisuje iste godine, smer Primenjeno softversko inženjerstvo. Osnovne akademske studije je završio 2019. godine. Master akademske studije upisuje 2019. godine, smer Primenjeno softversko inženjerstvo.

UTICAJ REGULACIJE NAPONA NA RAD DISTANTNE ZAŠTITE**IMPACT OF VOLTAGE REGULATION ON DISTANCE PROTECTION OPERATION**Marina Mitrović, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu predstavljen je primer kroz koji se razmatra uticaj regulacije napona i različiti efekti na distantnu zaštitu. Da bi se obezbedila kvalitetna isporuka električne energije, električna mreža mora da bude izgrađena tako da i pri najvećim opterećenjima može da obavlja svoju funkciju, sem toga mora da bude otporna na kvarove i poremećaje, tj. da ih dovoljno brzo eliminiše i smanji trajanje prekida napajanja na ekonomski opravdanu meru. U radu su navedene i objašnjene vrste zaštite vodova, moderna rešenja distantne zaštite, pojedinosti regulacionih transformatora, i data analiza primera.

Ključne reči: zaštita električnih mreža, distantna zaštita, savremena rešenja distantne zaštite, regulacija napona, regulacioni transformatori.

Abstract – This paper presents an example, through which the impact of voltage regulation and different effects on distance protection are considered. In order to ensure high quality delivery of electrical energy, the electrical grid must be built so that even at the highest loads it can perform its function, in addition it must be resistant to faults and disturbances, in other words, it must eliminate them fast enough and reduce the duration of power outages to an economically justified measure. The paper lists and explains the types of line protection, modern distance protection functions and applications, details of regulating transformers and gives an analysis of examples.

Keywords: line protection, distance protection, modern distance protection functions, voltage regulation, regulating transformers.

1. UVOD

Osnovni zahtev koji se postavlja pred distributivnu električnu mrežu (DEM) je rad koji je bezopasan za okolinu (ljude i materijalna dobra). Da bi se obezbedila kvalitetna isporuka električne energije tj. “neprekidna” isporuka, pri čemu je napon u dopuštenim granicama, DEM mora da bude izgrađena tako da i pri najvećim opterećenjima može da obavlja svoju funkciju. Sem toga mora da bude otporna na kvarove i poremećaje, tj. da ih dovoljno brzo eliminiše i smanji trajanje prekida napajanja na ekonomski opravdanu meru. Da bi se obezbedile zadovoljavajuće naponske prilike, distributivni električni sistem (DES) kao zadnja karika u lancu proizvodnja, prenos, distribucija mora da

ima mogućnost regulacije napona. Ova regulacija se uglavnom ostvaruje primenom regulatora napona pod opterećenjem u razvodnim transformatorskim stanicama 110/X kV ali i regulatora napona u transformatorskim stanicama X/0.4 kV [1].

Da bi elektroenergetski sistem mogao da funkcioniše potrebno je obezbediti zaštitu svakog njegovog elementa (generator, transformator, vod, motor, itd.), a takođe i pojedinih celina u okviru elektroenergetskog sistema od kvarova. Upravo zaštita celokupnog elektroenergetskog sistema od kvarova se ostvaruje relejnom zaštitom [2].

U nastavku rada, u drugoj glavi, biće predstavljeni principi zaštite vodova, kao i osnovni podaci o svakoj od tih zaštita. U trećoj glavi data su neka od savremenih rešenja i funkcija distantne zaštite i njihove pojedinosti. U četvrtoj glavi je predstavljen primer koji će se razmatrati i kroz koji će se analizirati regulacija napona i njen uticaj na distantnu zaštitu. Takođe, dat je matematički model vodova i regulacionih transformatora, informacije i brojne vrednosti elemenata distributivne mreže, kao i način na koji je izvršena normalizacija parametara i formiranje strukture mreže. U petoj glavi prikazani su rezultati i data analiza izvršenih proračuna. U šestoj glavi je iznet zaključak, dok se u sedmoj glavi navodi korišćena literatura.

2. ZAŠTITA ELEKTRIČNIH MREŽA

Osnovna funkcija EES-a jeste kontinualno snabdevanje potrošača električnom energijom odgovarajućeg kvaliteta. Pod kvalitetnom električnom energijom se, pored, propisanog napona i frekvencije podrazumeva i pouzdanost snabdevanja [1].

Zaštita elektroenergetskog sistema se vrši od nenormalnih stanja:

- stanja sa kvarom (kvarovima) i
- opasnih pogonskih stanja.

Zbog složenih oblika elektroenergetskih sistema i zbog potrebe za brзом eliminacijom kvara, bilo je potrebno napraviti zaštitu kojom bi se delovalo brže ako je mesto kvara bliže. Vreme delovanja ovim relejima zavisi od vrednosti merene impedanse. Ako se izabere da je merena impedansa jednaka impedansi od mesta ugradnje zaštite do mesta kvara, onda će se na osnovu vrednosti te impedanse dobiti informacija o udaljenosti (od mesta ugradnje) releja do mesta kvara. Na ovaj način je postignuto da vreme delovanja ovim relejima zavisi od udaljenosti ili distance od releja do mesta kvara. Zato se ovi releji nazivaju distantni (daljinski) releji [2].

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je dr Duško Bekut, red. prof.

Za zaštitu vodova se koriste sledeće vrste zaštita:

- prekostrujna,
- distantna,
- diferencijalna i
- osigurač.

Na vodovima srednjih napona (od 10 kV do 35 kV) se najčešće primenjuje prekostrujna zaštita. U mrežama radijalnog tipa se koristi neusmerena prekostrujna zaštita, dok se u petljastim mrežama mora koristiti i usmerena prekostrujna zaštita. Primena distantne zaštite je karakteristična za mreže naponskog nivoa 110 kV i više, dok je primena diferencijalne zaštite karakteristična za slučajeve kada se prethodne dve zaštite ne mogu efikasno i lako primeniti [2].

2.1. Neusmerena prekostrujna zaštita

Ova vrsta zaštite predstavlja najjednostavniju i najčešće primenjivanu zaštitu na sredjenaponskim vodovima (od 10 kV do 35 kV). Selektivnost zaštite sa strujno nezavisnom karakteristikom se postiže vremenskim i strujnim podešavanjem zaštite. Pri podešavanju zaštita uvek se polazi od krajnjih ka zaštitama koje su bliže izvoru napajanja radijalne mreže [2].

2.2. Usmerena prekostrujna zaštita

Radijalne jednostrano napajane mreže koje imaju paralelne vodove koji polaze sa istih sabirnica i završavaju se na nekim drugim sabirnicama, dvostrano napajani vodovi i prstenasto napajane mreže, ne mogu se štititi pomoću običnih prekostrujnih releja, jer se ne bi mogla obezbediti selektivnost. U ovim slučajevima se mora koristiti i usmerena prekostrujna zaštita koja predstavlja kombinaciju prekostrujnog i usmernog releja [2].

2.3. Distantna zaštita

Ova zaštita ima dva parametra za podešavanje: 1. impedansu pomoću koje se određuje domet zaštite i 2. vreme delovanja. Kombinacijom ova dva parametra se postiže traženi kvalitet zaštite, što je kvar bliže releju to se on brže eliminiše. Impedantno podešenje prvog stepena distantne zaštite mora biti manje od impedanse voda koji se štiti, jer bi u protivnom postojala mogućnost da se deluje i pri kvarovima na susednim vodovima [2].

2.4. Diferencijalna zaštita

Koriste se dve vrste diferencijalnih zaštita: uzdužna i poprečna. Uzdužna zaštita se obično primenjuje za vodove do dužina od oko 15 km i predstavlja najselektivniji i praktično najbrži tip zaštite koji se može koristiti za eliminaciju kvarova na vodu. Poprečna diferencijalna zaštita se primenjuje na paralelnim vodovima. U normalnom pogonu i kod kvarova van tih paralelnih vodova, struje po paralelnim vodovima su iste, a njihova razlika je brojno jednaka nuli. Kada se dogodi kvar na jednom od vodova, razlika struja je brojno različita od nule i tada se zaštitom deluje [2].

2.5. Automatsko ponovno uključanje voda

Eksploataciona iskustva su pokazala da ako se posle isključenja kvara ponovo uključi vod, u 70 % do 80 % slučajeva dolazi do spontane eliminacije kvara. Uređaji za automatsko ponovno uključanje se dele prema [2]:

- brzini delovanja,
- broju obuhvaćenih faza,
- broju ciklusa,

- dopunskim uslovima i
- načinu aktiviranja.

2.6. Termička zaštita

Kod pojave prevelikog zagrevanja vodova dolazi do povećanja ugiba iznad dozvoljenog, zbog čega se smanjuje sigurnosna visina između provodnika i tla. Termička zaštita vodova bi podrazumevala zaštitu kojom bi se delovalo kada temperatura provodnika premaši neku zadatu vrednost. U praksi se termička zaštita najčešće izvodi kao zaštita od preopterećenja, odnosno zaštitom se deluje ako je preopterećenje veće od nekog zadatog [2].

3. SAVREMENA REŠENJA DISTANTNE ZAŠTITE

Povećanjem opterećenja prenosne mreže i smanjenjem stabilnosti, zahteva se brže reagovanje relejne zaštite, a da se pritom zadrži siguran rad releja. Proizvođači releja su u modernu distantnu zaštitu integrisali veliki broj elemenata zaštite, kontrole, nadzora i merenja koji korisniku pružaju mogućnosti kreiranja napredne zaštite i kontrole, koja nije bila moguća korišćenjem ranijih tehnologija [3].

3.1. Pouzdanost

Inženjeri teže da postignu veoma brzo reagovanje za sve kvarove na prenosnoj mreži, pomoću relejne zaštite podržane komunikacionim kanalima. Moderni releji imaju integrisanu funkciju komunikacije u okviru iste opreme distantnog releja i omogućavaju digitalnu komunikaciju između releja za veoma brzu zaštitu vodova, nadzor i kontrolu. Mogućnost upotrebe sinhronizovanog merenja fazora će omogućiti napredne strategije za zaštitu i kontrolu, a rezultat su povećana pouzdanost, bolja zaštita i kontrola [3].

3.2. Zaštitne funkcije

Distantni releji obuhvataju sledeće integrisane zaštitne elemente:

- visoko-osetljivu usmerenu zemljospojnu zaštitu,
- više izbora karakterističnih krivih za usmerenu i neusmerenu prekostrujnu zaštitu, podnaponske ili nadnaponske elemente,
- termičku zaštitu,
- zaštitu u slučaju kvara prekidača,
- principe zaštite za udaljene kvarove, potpomognute kanalima komunikacije,
- za iznenadan gubitak opterećenja,
- itd.

Veoma brza zaštita vodova je poželjna osobina na kritičnim vodovima sa uskim marginama za stabilnost [3].

3.3. Kontrola, nadzor i merenje

Integrisani elementi za kontrolu uključuju tajmere, digitalne brojače, naprednu logiku, matematičke funkcije i binarne ulaze koji detektuju promenu stanja na spoljašnjim kontaktima. Funkcije nadzora imaju važnu ulogu u održavanju [3].

3.4. Komunikacija

Moderni digitalni releji imaju integrisane funkcije za komunikaciju, koje su značajne u mnogim oblastima povezanim sa zaštitom i razmenom podataka neophodnih za kontrolu, nadzor i snimanje događaja i kvarova. Moderni distantni releji poseduju više prednjih i zadnjih serijskih portova za komunikaciju, kako bi korisnik mogao da pristupi izveštajima o snimljenim događajima i izvrši izmene podešenja, lokalno u trafostanici ili sa udaljenosti putem modema [3].

3.5. Lociranje kvara

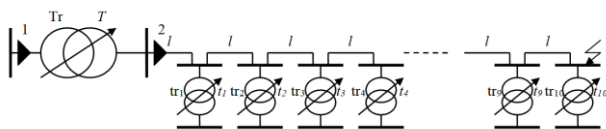
Precizno lociranje kvara pomaže operatorima u restoraciji napajanja i smanjenju vremena bez napajanja, troškova operacija i pritužbi potrošača. Moderni distantni releji imaju integrisane digitalne kanale za komunikaciju sa sposobnošću da razmenjuju analogne vrednosti i podatke između uređaja, uz logičke podatke o statusima uređaja [3].

3.6. Sinhronizovano merenje fazora

Sinhronizovanje uzoraka, koje koriste zaštitni releji, u okviru 1 μ s, omogućeno je zahvaljujući razvoju satelitskih sistema i kompjuterske tehnologije. Ovaj dodatak u distantnim reljima rezultuje u povećanoj pouzdanosti sistema i omogućava lakše vršenje raznih analiza, zaštitu i kontrolu, nego što je to slučaj kada se koriste drugi izvori podataka [3].

4. POSTAVKA PROBLEMA

Razmatra se jednostavna distributivna mreža (DM) prikazana na slici 1.



Slika 1. Distributivna mreža

Potrebno je:

1. za različite pozicije regulacione sklopke transformatora Tr, odrediti impedansu koju meri zaštita 1 i impedansu koju meri zaštita 2 za dvopolni kratak spoj na SN sabirnici transformatora tr_{10} ,
2. ponoviti prethodni zadatak ako se uvaži efekat zagrevanja provodnika pri kratkom spoju,
3. ponoviti zadatak pod rednim brojem 1 kada se na SN sabirnicu tr_4 priključi distributivni generator koji injektira 5 MVA ($S_{PKSMIN}=1000$ MVA, $x_d=25$ %, $S_n=5$ MVA, $V_n=21$ kV) i
4. ponoviti prethodni zadatak ako se uvaži efekat zagrevanja provodnika pri kratkom spoju.

4.1. Vodovi

Vodovi su elementi za prenos električne energije. Mogu biti s golim provodnicima i u vidu kablova. Prvi se koriste isključivo kao nadzemni, okačeni preko izolatora na stubove. Kablovi se skoro isključivo pružaju ispod zemlje. Izuzeci su samonoseći kablovi na niskom naponu koji se kače na stubove [4].

4.2. Regulacioni transformatori

Regulacioni transformatori spadaju u najvažnije regulacione resurse za regulaciju napona i reaktivnih snaga elektroenergetskih sistema. Oni se uobičajeno izvode sa razdvojenim namotajima, ali i kao autotransformatori.

4.3. Podaci o elementima distributivne mreže

Dvonamotajni regulacioni transformator 110/21 kV/kV: $V_{n1}=110.0$ kV, $V_{n2}=21.0$ kV, $S_n=31.5$ MVA, $u_k=8$ %

Dvonamotajni regulacioni transformatori 20/0.4 kV/kV: $V_{n1}=20.0$ kV, $V_{n2}=0.4$ kV, $S_n=1.0$ MVA, $u_k=6$ %

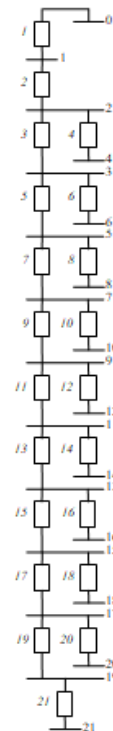
Vodovi 20 kV: $r=0.32$ Ω /km, $x=0.33$ Ω /km, $b=120.0$ μ S/km, $g=0.0$ μ S/km, $l_1=l_2=l_3=l_4=l_5=l_6=l_7=l_8=l_9=l_{10}=1$ km

4.4. Normalizacija parametara distributivne mreže

Za bazne vrednosti izabrane su sledeće veličine: $S^b=3.15$ MVA, $V_{110}^b=110$ kV, $V_{20}^b=20$ kV, $V_{0.4}^b=0.4$ kV

4.5. Formiranje strukture mreže

Na slici 2 je prikazana struktura mreže sa numerisanim granama i čvorovima.



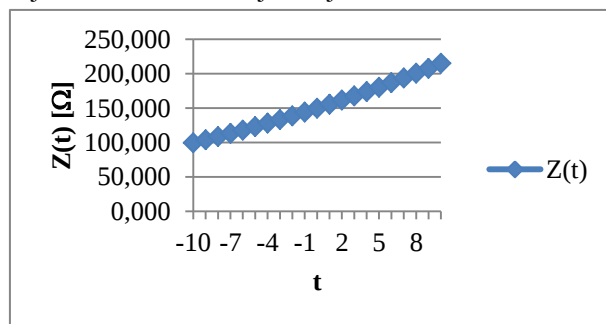
Slika 2. Struktura mreže

5. NUMERIČKA VERIFIKACIJA PRORAČUNA

Razmatraju se efekti ugradnje distributivnog generatora i povećanja termičke otpornosti na rad distantne zaštite.

5.1. Prvi slučaj

Na slici 3 je dat grafički prikaz vrednosti impedanse koju meri zaštita 1 za prvi slučaj, dok je vrednost impedanse koju meri zaštita 2 za taj slučaj sledeća: $Z=4.6$ Ω .

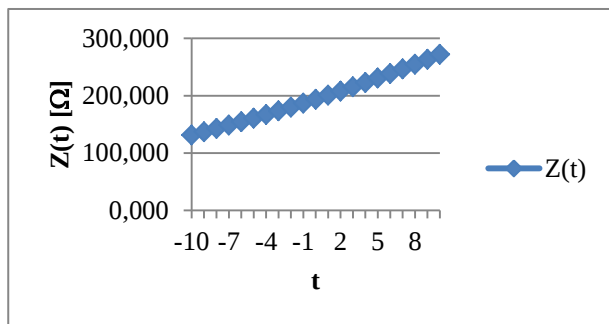


Slika 3. Impedansa koju meri zaštita 1 za prvi slučaj

5.2. Drugi slučaj

Na slici 4 je dat grafički prikaz vrednosti impedanse koju meri zaštita 1 za drugi slučaj, dok je vrednost impedanse koju meri zaštita 2 za taj slučaj sledeća: $Z=6.417$ Ω .

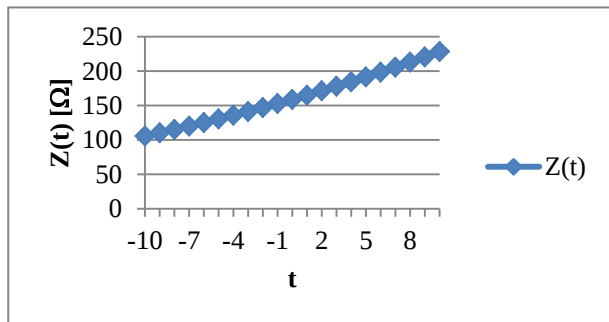
Vrednost impedanse koju meri zaštita na mestu 2 je konstantna, ne zavisi od promene regulacione sklopke transformatora, na nju utiču samo vrednosti impedanse vodova.



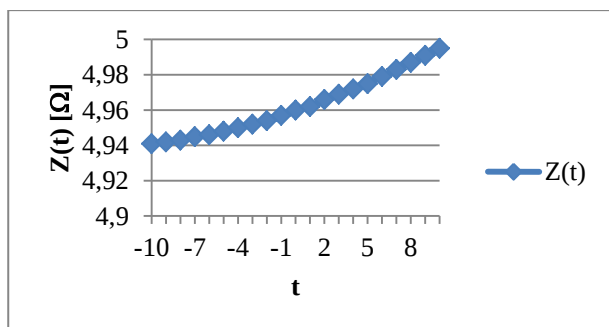
Slika 4. Impedansa koju meri zaštita 1 za drugi slučaj

5.3. Treći slučaj

Na slici 5 je dat grafički prikaz vrednosti impedanse koju meri zaštita 1 za treći slučaj, dok je vrednost impedanse koju meri zaštita 2 za taj slučaj data na slici 6. Primećuje se povećanje u odnosu na prvi slučaj kada generator nije bio priključen. Dakle, u ovoj situaciji bi bilo potrebno promeniti podešenja distantne zaštite.



Slika 5. Impedansa koju meri zaštita 1 za treći slučaj

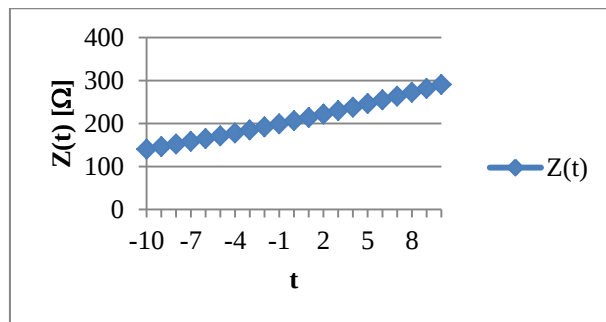


Slika 6. Impedansa koju meri zaštita 2 za treći slučaj

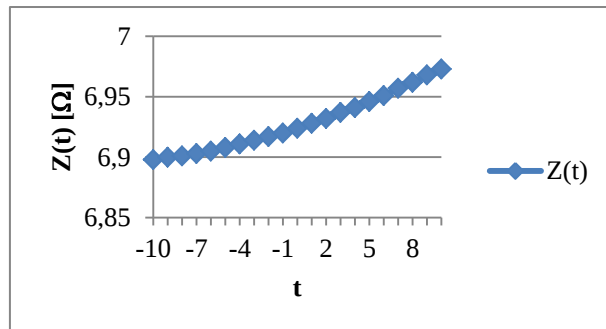
5.4. Četvrti slučaj

Na slici 7 dat je grafički prikaz vrednosti impedanse koju meri zaštita 1 za četvrti slučaj, dok je vrednost impedanse koju meri zaštita 2 za taj slučaj data na slici 8.

Do povećanja je došlo usled povećanja realnog dela impedanse, odnosno, otpornosti, jer je otpornost koja se meri na vodovima sada dobijena korišćenjem nove vrednosti podužne otpornosti.



Slika 7. Impedansa koju meri zaštita 1 za četvrti slučaj



Slika 8. Impedansa koju meri zaštita 2 za četvrti slučaj

6. ZAKLJUČAK

Izvršena je analiza brojnih vrednosti iz prethodna četiri primera. Razmatrani su efekti koje ima regulacija napona na vrednosti impedanse koje meri distantna zaštita. Takođe, razmatrani su uticaji uvažavanja efekta zagrevanja provodnika pri kratkom spoju, kao i ugradnje distributivnog generatora. Konstatovano je u kojim slučajevima treba promeniti podešenje zaštite.

7. LITERATURA

- [1] Miroslav D. Nimrihter: *Elektrodistributivni sistemi*, Fakultet tehničkih nauka, Novi Sad, 2009.
- [2] Duško Bekut: *Relejna zaštita*, Fakultet tehničkih nauka, Novi Sad, 2009.
- [3] Demetrios A. Tziouvaras, Simon R. Chano, Jean-Luc Chaneliere, Juergen Holbach, Silvia Moroni, Mohindar S. Sachdev, Constantine E. Zavolas: *Modern Distance Protection; Functions and Applications*, CIGRE publication, 2008.
- [4] Vladimir Strezoski: *Analiza elektroenergetskih sistema*, Fakultet tehničkih nauka, Novi Sad, 2012.
- [5] Goran S. Švenda: *Osnovi elektroenergetike: Matematički modeli i proračuni*, Fakultet tehničkih nauka, Novi Sad, 2014.

Kratka biografija:



Marina Mitrović rođena je u Novom Sadu 1994. god. Osnovne studije završila je na Fakultetu tehničkih nauka 2018. godine iz oblasti Elektrotehnike i računarstva, smer Elektroenergetski sistemi. Master rad na istom fakultetu, smer Elektroenergetika – Elektroenergetski sistemi, odbranila je 2021. godine.

Kontakt: marina.mitrovic12@hotmail.com

**UTICAJ OBNOVLJIVIH IZVORA ENERGIJE NA PROPADE NAPONA U
DISTRIBUTIVNIM MREŽAMA****INFLUENCE OF RENEWABLE ENERGY SOURCES ON VOLTAGE DROP IN THE
DISTRIBUTION NETWORKS**

Vuk Vidačić, Vladimir Katić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Ovaj rad prikazuje uticaj obnovljivih izvora energije na propade napona u distributivnim mrežama. Simulirani su kratki spojevi sa i bez obnovljivih izvora energije u mreži i prikazan je njihov uticaj.

Ključne reči: Obnovljivi izvori energije, Propad napona, IEEE 13 mreža

Abstract – This paper shows the impact of renewable energy sources on voltage dips in distribution networks. Short circuits with and without renewable energy sources in the network are simulated and their impact is displayed..

Keywords: Renewable energy sources, Voltage drop, IEEE 13 network

1. UVOD

U današnjem svijetu zbog konstantnog razvoja industrije i sve veće populacije planete Zemlje potreba za novim izvorima energije iz dana u dan raste. Veći deo potreba (oko 76%) pokriva se iz neobnovljivih izvora energije, većinom fosilnih goriva (uglja, nafte i prirodnog gasa). za čiji nastanak potrebno je da prođu milioni godina [1]. Prema podacima kompanije „British petroleum“, jedne od najvećih vodećih energetskih firmi u svijetu, zalihe neobnovljivih izvora energije će u skorijoj budućnosti biti potpuno iscrpljene (predviđa se da će uglja u svijetu biti još 150 godina, nafte 40 godina i plina 50 godina) [2]. Takođe, svjedoci smo sve većeg zagađenja životne sredine, intenzivnije promjene klime, rasta cijene fosilnih goriva i sve manje količine istog. Pored toga, korišćenje energije mora biti ekološki prihvatljivo („čisto“) i održivo. Iz tih razloga današnje preferencije idu ka široj upotrebi električne energije dobijene iz obnovljivih izvora energije, odnosno energije sunca, vetra, vode (hidro), biomase, geotermalne i dr. [3].

Priključenje obnovljivih izvora u tradicionalne elektroenergetske mreže obično je na distributivnom nivou, što stvara nove (specifične) uslove za eksploataciju. Mada su takve mreže efikasnije (smanjeni su troškovi prenosa) i sa stabilnijim napajanjem, učestanost kvarova na njima je veća. pa se sve više pažnje posvećuje kvalitetu električne energije, a posebno kvalitetu isporučene energije ili kvalitetu napona [4,5].

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Vladimir Katić.

U elektroenergetskim sistemima, posebno u industrijskim sistemima oko 92% od svih poremećaja u napajanju električnom energijom kao uzrok imaju propade napona [6]. Propad napona često vodi do velikih finansijskih gubitaka, dolazi do oštećenja opreme i do velikih zastoja u industrijskim pogonima.

Jedan od načina da se propadi napona ublaže je ugradnja obnovljivih izvora energije. Oni su sve češća pojava u distributivnim mrežama sa ciljem da preuzmu dio opterećenja glavnog izvora energije. Takođe, oni se mogu iskoristiti za podršku mreži, odnosno da se njihovim radom ublaži propad napona pri nastanku kvara u mreži. To se postiže specifičnim zahtevima o radu (*Grid Code*), odnosno LVRT (*Low Voltage Ride Through*) sposobnošću, koji podrazumevaju da oni pri pojavama kvara, odnosno propada napona povećavaju proizvodnju reaktivne energije i time pozitivno utiču na naponske prilike i ispađe potrošača sa mreže [7].

Cilj ovog rada je da ispita uticaj priključenja obnovljivih izvora na nivo propada napona u distributivnoj mreži. Radi referentne analize istraživanje sprovedeno je korišćenjem standardnih distributivnih test mreža, konkretno IEEE test mreže sa 13 čvorova (*IEEE 13-bus Test Grid*).

2. DISTRIBUTIVNE TEST MREŽE

Distributivne test mreže predstavljaju modele mreža koji su replike ponašanja stvarnih distributivnih mreža. Cilj je reprodukovati karakteristike stvarnih mreža, pri čemu se uvažavaju specifičnosti određenog regiona gdje se mreža nalazi. Distributivne test mreže su veoma koristan alat i veoma često se koriste u istraživanjima u oblasti elektroenergetike. One omogućavaju da rezultati budu međunarodno provjerljivi i da se sva testiranja mogu ponoviti i da se rezultati koji dobiju mogu posmatrati zajedno sa rezultatima drugih istraživanja.

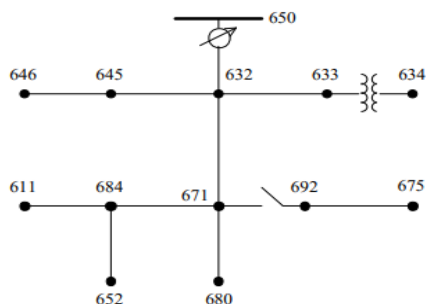
Ovi tipovi mreža su dizajnirani prije svega za testiranje algoritama, pa su svi potrebni parametri za modelovanje jasno i nedvosmisleno definisani i dati, kao i šematski prikazi i instrukcije za modelovanje. Postojanje ovakve dokumentacije standardizuje modelovanje i povećava pouzdanost dobijenih rezultata. Pored električnih osobnosti modela (oprema, vodovi, potrošači, zaštitni uređaji, itd.) u dokumentaciji test mreže definisani su geografski podaci o vodovima kao i međusobnim vezama (dužina vodova, njihove rute i način povezivanja), zatim definisani su parametric potrošača u svakom čvoru mreže (broj potrošača, veličina, klasa, profil potrošnje, vršno

opterećenje-kW kao i faktor snage), i linijski dijagram u kome su ucrtani vodovi i čvorovi i informacije o transformatorima (MVA, kV, impedansama, sprežni brojevi, itd.) [8].

2.1. IEEE test mreže

IEEE test mreže uglavnom predstavljaju delove distributivnih elektroenergetskih sistema u Sjedinjenim Američkim Državama i najčešće su korišćene distributivne test mreže za razna istraživanja. IEEE je kreirao test mreže sa 4, 13, 30, 34, 37, 123 i 8500 čvorova. Originalni dokument koji opisuje test mreže sa 13, 34, 37, i 123 čvora kreiran je 1992. godine, a njegova publikacija je odobrena 2000. godine. Posle toga publikovane su test mreže sa 8500 čvorova (2010.), kao i evropska nisko-naponska test mreža (2015.) [8].

U ovom radu korišćena je IEEE 13-bus test mreža. Ono što karakteriše ovu mrežu je da je ona mala, nebalansirana i u značajnoj mjeri opterećena mreža. Naponski nivo mreže je 4,16 kV. Mreža se sastoji od 13 čvorova koji su međusobno povezani sa 10 vodova, ima jedan glavni izvor napajanja kao i dva transformatora (jedan distributivni transformator ΔY 115/4,16 kV i jedan linijski transformator YY 4,16/0,480 kV), dvije kondenzatorske baterije, kao i neuravnoteženi distributivni potrošači. Mreža je kombinacija trofazne i dijelova dvofaznim i monofaznim vodovima. Izgled IEEE-13 mreže prikazan je na slici 1.



Slika 1. IEEE 13-bus test mreža [8]

2.2. Model test mreže

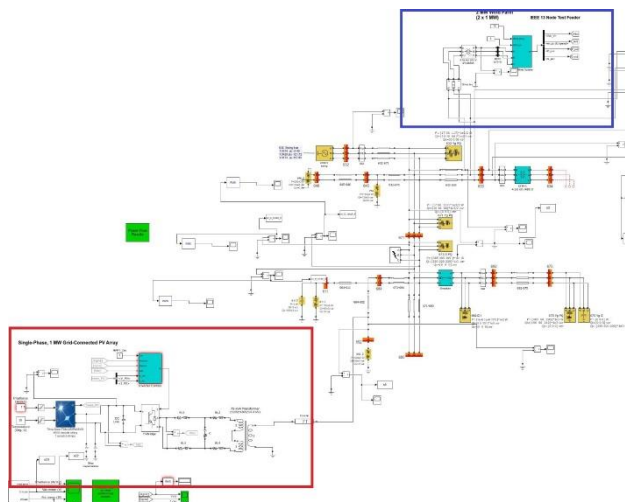
Na slici 2 prikazan je izgled modela test mreže koja je korištena za ovo istraživanje modelovana u Matlab/Simulink softverskom alatu. U modelu su dodate vjetroelektrana (ograničena plavim pravougaonikom) i solarna elektrana (ograničena crvenim pravougaonikom). Čvorovi su prikazani malim narandžastim pravougaonicima, potrošači i glavni izvor energije prikazani su žutim pravougaonicima, transformatori plavim, a mjesto na kojem će biti simulirani kratki spojevi prikazano je bjelim pravougaonikom sa znakom munje u njemu.

3. MODELI DISTRIBUIRANIH IZVORA

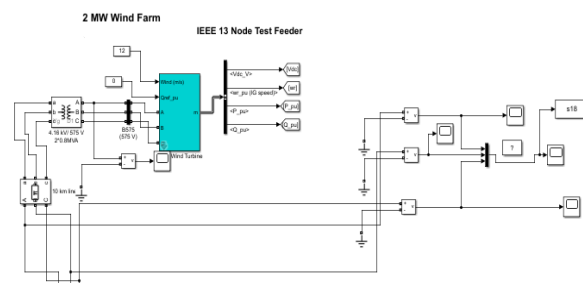
3.1 Model vjetroelektrane

Na test mrežu prikačena je vjetroelektrana nominalne snage 2 MW (4 x 0,5 MW), čiji je Matlab/Simulink model prikazan na slici 3. Vjetroelektrana je sastavljena od četiri vjetrogeneratora, sa dvostruko napajanim klizno kolutnim asinhronim generatorom (DFIG - Doubly Fed Induction Generator) i turbinom promjenljive brzine obrtanja, nominalne snage 0,5 MW. Nominalni napon statora vjetrogeneratora je 575 V. On je preko transformatora

575V/4160V i preko voda dugačkog 10 km povezan sa mrežom (slika 3).



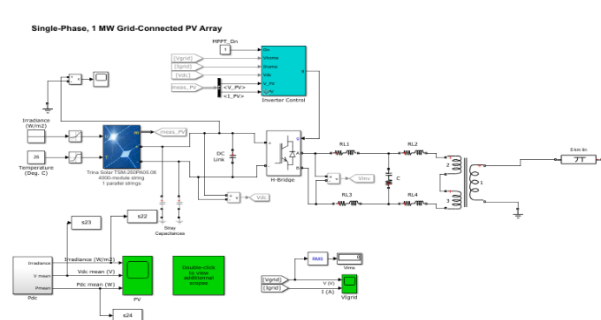
Slika 2. Model IEEE 13-bus test mreže u Matlab-u



Slika 3. Šema vjetroelektrane

3.2 Model solarne elektrane

Solarna elektrana, koja je korišćena u ovom istraživanju sastoji se od 4000 modula. Maksimalna snaga jednog modula je 250 Wp (dakle nominalna snaga elektrane je $1 \times 4000 \times 250 \text{ Wp} = 1 \text{ MWp}$), broj ćelija po modulu je 60, struja kratkog spoja je 8,55 A, napon praznog hoda je 37,6 V, dok je napon pri maksimalnoj snazi 31 V. Na slici 4 prikazana je šema Matlab/Simulink modela ove elektrane.



Slika 4. Šema solarne elektrane

4. PLAN ISTRAŽIVANJA

Cilj istraživanja je da se pokaže kako na nivo propada napona na mjestima priključivanja utiču rad solarne i vjetroelektrane, pri 3 vrste kratkih spojeva. Simuliran je kratak spoj kod čvora 671 i to u 4 slučajeva:

- kada na mrežu nisu priključeni ni solarna, ni vjetroelektrana,
- kada je na mrežu priključena samo solarna elektrana,

- kada je na mrežu priključena samo vjetroelektrana i
- kada su na mrežu priključene obje (i solarna i vjetroelektrana)

Propad napona sniman je u slučaju jednopolnog (zemljospoj faze A), dvopolnog (međufazni spoj faza A i B) i trojnog kratkog spoja, a napon je posmatran (mjereno) na mjestu priključenja solarne (kod čvora 652) i vjetroelektrane (kod čvora 633).

5. POREĐENJE REZULTATA

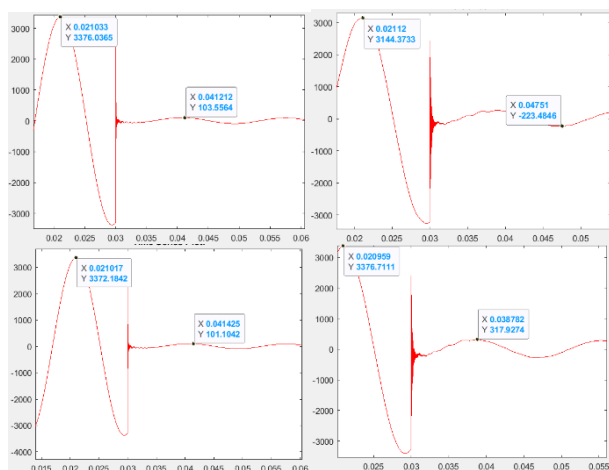
Rezultati simulacija urađeni su za sva četiri slučaja pri pojavi kratkog spoja na kod čvora 671. Rezultati su prikazani na slikama 5-7. Na svakoj slici prikazana su četiri razmatrana slučaja i to kada nisu priključene solarna i vjetroelektrana (gore lijevo), kada je priključena samo vjetroelektrana (gore desno), kada je priključena samo solarna elektrana (dole lijevo) i kada su priključene obje elektrane (dole desno).

Zbog velikog broja rezultata ovdje će biti prikazani samo slučajevi propada napona snimljeni na fazi A priključka vjetroelektrane.

5.1. Jednopolni kratki spoj između faze A i zemlje

Nakon poređenja rezultata koji su dobijeni prilikom simulacije kratkog spoja između faze A i zemlje primjetno je da je propad napona manji par desetina volti kada je priključena vjetroelektrana. Rezultati prikazani na slici 5 prikazuju propad napona mjereno na mjestu priključka vjetroelektrane na mrežu (kod čvora 633) pri zemljospoju faze A kod čvora 671. Dati su rezultati za sva četiri pomenuta slučaja.

Iz slike 5 može se primjetiti da napon na mjestu priključenja vjetroelektrane, tj. njegova amplituda, opadne sa nominalnih 3390 V na oko stotinjak volti u slučajevima kada nije priključena vjetroelektrana, a da kada je priključena vjetroelektrana pada na nekih 220-320 V, u zavisnosti da li je priključena još solarna elektrana ili nije.



Slika 5. Prikaz propada napona na mjestu priključenja vjetroelektrane pri jednopolnom kratkom spoju faze A

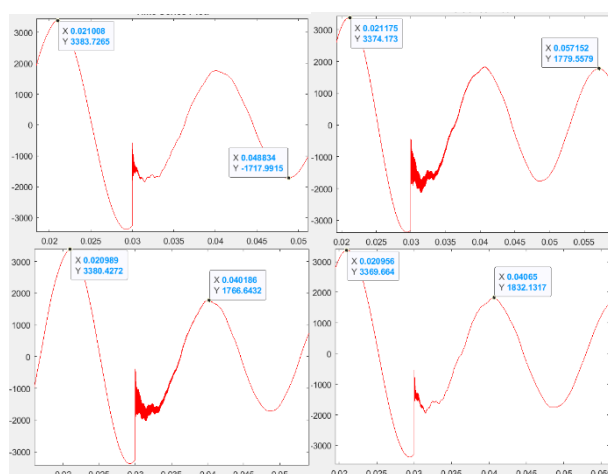
Prilikom propada mogu se registrovati i tranzijentni viši harmonici u naponu, tako da ukupna harmonijska distorzija (THD – Total Harmonic Distortion) napona skoči čak na 164,4%, ali se po nastajanju propada smanji na 0,9%. Ova pojava je ranije primećena i nazvana

harmonijski otisak (Harmonic Footprint) i karakteriše svaki propad napona [9].

5.2. Dvopolni kratki spoj između faza A i B

Rezultati propada napona na fazi A na mjestu priključka vjetroelektrane prilikom dvopolnog kratkog spoja između faza A i B prikazani su na slici 6. Opet su data sva četiri posmatrana slučaja. Sa slike se primjeti slična pojava kao i prilikom simuliranja jednopolnog kratkog spoja. Propad napona je nešto blaži, za par desetina volti, kada su priključene solarna ili vjetro elektrana, nego kada distributivni izvori nisu priključeni. Rezultati mjerenja na ostalim fazama takođe prikazuju napon, koji je stabilni kada su priključeni distributivni izvori električne energije.

U ovom slučaju dolazi do nešto manjeg skoka harmonijske distorzije tokom tranzicione periode propada napona. Vrednost THDU skoči na 49,9%, a na kraju ovog perioda padne na 0,41%.



Slika 6. Prikaz propada napona na fazi A priključka vjetroelektrane prilikom dvopolnog propada napona

5.3. Trojnog kratki spoj između faza A, B i C

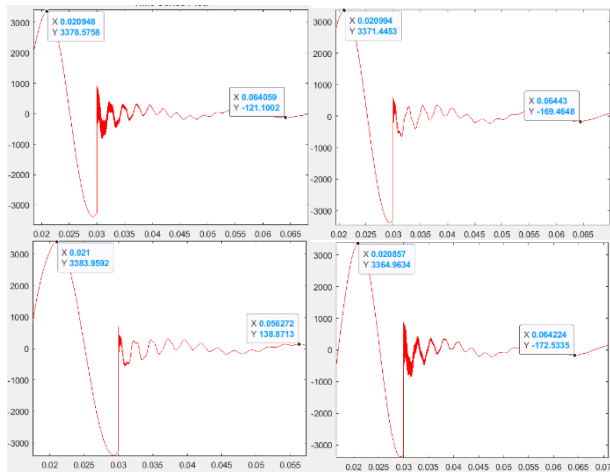
Prilikom trojnog kratkog spoja, takođe pratilo se ponašanje napona na sve tri faze na istim mjestima. Rezultati simulacija pokazali su takođe da se napon ponaša stabilnije kada su priključene solarna i vjetroelektrana. Razlika u propadu napona je opet u par desetina volti. Najveća razlika je kada su priključene obje elektrane u odnosu na slučaj kada nije priključena niti jedna elektrana i iznosi 50 V. Ova razlika u propadu napona predstavlja oko 1.2% od nominalnog napona, što je veoma značajno u može da utiče na pojavu da li će podnaponska zaštita da odreaguje ili neće. Rezultati simulacije na fazi A priključka vjetroelektrane u slučaju trojnog kratkog spoja prikazani su na slici 7.

U tranzijentnom delu propada napona opet se može uočiti visok nivo THDU (izobličenja napona), koje sada dostiže čak 145,5%, a po završetku propadanja napona ovo izobličenje sa ustali na 12,6%.

5.4. Ostali rezultati

Slični rezultati simulacija dobijaju se i na ostalim fazama (B i C) na mjestu priključenja vjetroelektrane sa malom razlikom u dobijenim vrednostima. Takođe, na priključku solarne elektrane, kada dođe do kratkog spoja na posmatranom mjestu (kod čvora 671), takođe se javljaju

slične pojave. Može se zaključiti, da rad vjetroelektrane i solarne elektrane u slučaju pojave kratkog spoja u distributivnoj mreži može pozitivno uticati na stabilnost napona, tj. da se može uočiti da su ispunjeni zahtevi LVRT.



Slika 7. Prikaz propada napona na fazi A priključka vjetroelektrane prilikom trolpnog propada napona

6. ZAKLJUČAK

U radu ispitivan je uticaj obnovljivih izvora energije na propade napona u distributivnoj mreži. Istraživanje je vođeno tako što su na IEEE 13 test mreži simulirane 3 vrste kratkog spoja (jednopolni, dvopolni i trolpni) pri tome mjereni propadi napona na 2 mjesta u mreži (na mjestu gdje se priključuje solarna i na mjestu gdje se priključuje vjetroelektrana) i to u 4 slučaja:

- kada nisu priključene niti solarna, niti vjetroelektrana,
- kada je priključena samo vjetroelektrana,
- kada je priključena samo solarna elektrana i
- kada su priključene obje elektrane.

Cilj istraživanja bio je da se pokaže mogućnost obnovljivih izvora da smanje propad napona na distributivnoj mreži. Ta pojava, poznatija kao LVRT, predstavlja mogućnost obnovljivih izvora energije (u ovom slučaju solarnih i vjetroelektrana) da prilikom pojave kvara u mreži, tj. propada napona, povećaju svoju proizvodnju reaktivne energije, a smanje proizvodnju aktivne. Time kompenzuju propad napona na mreži. U ovom radu potvrđena je ta sposobnost obnovljivih izvora, tj. uočeno je poređenjem rezultata da je propad napona prilikom kvara manji kada su priključene solarna i vjetroelektrana, ili jedna od njih, na mrežu. Razlika u propadu napona u nekim slučajevima doseže i do stotinjak volti, što je preko 2 posto od nominalne vrijednosti napona. To je značajan procenat za način rada podnaponske zaštite, kada se odlučuje da li će ona da odreaguje ili ne.

Ovim je pokazana još jedna prednost korišćenja obnovljivih izvora energije u distributivnim mrežama. Naravno rezultate ovog istraživanja treba uzimati sa određenom rezervom zbog više faktora kao što su vjerodostojnost simulacije rada električnih komponenti u Matlab/Simulink-u, generalizovanje distributivnih mreža (postoje mreže sa potrošačima veće snage, takođe distributivnim izvorima veće snage, sa većim brojem potrošača,...). Svi ovi faktori mogli su da utiču na

rezultate ovog istraživanja, ali ipak oni su u saglasnosti sa opštim zaključcima prikazanim u drugim naučnim i stručnim radovima.

7. LITERATURA

- [1] World Energy Council, „World Energy Resources – 2013 Survey: Summary“, London, 2013, <https://www.worldenergy.org>
- [2] British Petroleum, „Energy Outlook: 2020 edition“, BP Global, London, 2020, <https://www.bp.com>
- [3] European Commission, „Energy, Climate change, Environment - Overall targets and reporting“, www.ec.europa.eu
- [4] V. Katić, „Kvalitet električne energije – viši harmonici“, Edicija tehničke nauke - Monografije, br.6, Fakultet tehničkih nauka, Novi Sad, 2002.
- [5] J. Janković, V. Katić, „Uticaj propada napona na rad punionica za električna vozila napajanih iz sistema krovih FN elektrana“, Zbornik radova FTN, Novi Sad, God. 34 Br. 11, 2019, pp. 1993-1996.
- [6] D.D. Sabin, T.E. Grebe, A. Sundaram, „RMS voltage variation statistical analysis for a survey of distribution system power quality performance,“ IEEE Power Engineering Society Winter Meeting, New York, 1999.
- [7] A. Benali et al, „Power Quality Improvement and Low Voltage Ride Through Capability in Hybrid Wind-PV Farms Grid-Connected Using Dynamic Voltage Restorer“, IEEE Access, Vol.6, 2018, pp.68634-68648.
- [8] A.M. Stanisavljević, V.A. Katić, B.P. Dumnić, B.P. 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, February 2018, 115-129
- [9] V.A. Katić, A.M. Stanisavljević, „Smart Detection of Voltage Dips Using Voltage Harmonics Footprint“, IEEE Tran. on Industry Applications, Vol.54, No.5, Sep./Oct. 2018, pp.5331-5342.

Kratka biografija:



Vuk Vidačić rođen je 13.06.1996. god. u Trebinju. Gimnaziju „Jovan Dučić“ završio je 2015. god. Iste godine je uspješno položio prijemni ispit i upisao se na Fakultet tehničkih nauka, na studijski program Energetika, elektronika i telekomunikacije. Diplomirao je 2019. god. Kontakt: vuk13061996@gmail.com



Vladimir A. Katić, red. prof. rođen je 1954. godine u Novom Sadu. Diplomirao je na Fakultetu tehničkih nauka u Novom Sadu 1978. god., a magistrirao i doktorirao na Univerzitetu u Beogradu 1981. i 1991. godine, respektivno. Od 2002. god. je redovni profesor Univerziteta u Novom Sadu. Oblasti interesovanja su energetska elektronika, obnovljivi izvori električne energije, električna vozila i kvalitet električne energije.

EVALUACIJA UPOTREBLJIVOSTI NOVINSKIH APLIKACIJA ZA ANDROID UREĐAJE**USABILITY EVALUATION OF NEWSPAPER APPLICATIONS FOR ANDROID DEVICES**Stefan Milanović, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – Ovaj rad predstavlja analizu, detaljno ispitivanje i upoređivanje određenih novinskih aplikacija za Android uređaje. Analiza i ispitivanje upotrebljivosti dobrog korisničkog interfejsa podrazumeva evaluaciju po heuristikama i primenu Šnajdermanovih „Osam zlatnih pravila“. Proučavanjem i testiranjem ostvaruju se saznanja koja ukazuju na prednosti, slabosti i procenu upotrebljivosti određenih novinskih aplikacija za Android uređaje. Na osnovu rezultata ispitivanja priloženi su predlozi i smernice kako bi se ispravili uočeni nedostaci.

Ključne reči: novinske aplikacije, HCI heuristike.

Abstract – This thesis presents an analysis, detailed examination and comparison of certain newspaper applications for Android devices. The analysis and testing of the usability of a good user interface include evaluation by heuristics based on Shneiderman's "Eight Golden Rules". By studying and testing applications, knowledge is gained about the advantages, weaknesses and usability assessment of certain newspaper applications for Android devices. Based on the test results, suggestions and guidelines are provided in order to correct the observed disadvantages.

Keywords: newspaper applications, HCI heuristics.

1. UVOD

Zadatak rada predstavlja analizu, detaljno ispitivanje i upoređivanje određenih Android aplikacija koje se odnose na oblast informisanja. Analiza podrazumeva evaluaciju po heuristikama i primenu Šnajdermanovih „Osam zlatnih pravila“.

Testiranjem i proučavanjem ostvaruju se saznanja kojima se ukazuje na slabosti, pogodnost i procenu upotrebljivosti određenih novinskih aplikacija za Android uređaje. Na osnovu rezultata ispitivanja izvodi se poređenje i predlažu unapređenja za nedostatke ispitanih aplikacija. U grupu Android aplikacija predviđenih za analizu informativnog karaktera spada *The New York Times (NYT)*, *British Broadcasting Corporation (BBC) News* i *B92*.

Radi detaljnog ispitivanja neophodno je razumeti osnovne elemente i upoznati način rada i karakteristike Android operativnog sistema. Razumeti ideju i funkcionalnosti sistema, prepoznati njegove prednosti i mane. U ovom radu predviđeno je prvenstveno upoznavanje sa Android opera-

tivnim sistemom. Pod upoznavanjem podrazumeva se prolazak kroz istoriju verzija koje je omogućio Android.

Potom kratak osvrt na softversku i hardversku arhitekturu sistema. Nakon razjašnjenih osnova potrebno je uvideti načine instalacije aplikacija na uređaj i njihovu primenu. Zatim slede ključna uputstva prvog dela rada prikazana kroz dozvoljene mogućnosti implementacije aplikacija. Osvrt na pojedinačne elemente Android sistema. Detaljan opis vizuelnih efekata određenih elemenata i funkcionalnosti koje mogu da obezbede. Kraj prvog dela upućuje na ekspertske analize i objašnjena konkretna pravila kojima će biti sprovedena analiza i ispitivanje aplikacija za vesti. Drugi deo rada predstavljen kroz tri glavne celine gde svaka od navedenih aplikacija pojedinačno prolazi kroz ispitivanje zasnovano na evaluaciji po heuristikama i Šnajdermanovim pravilima.

Rezultat rada ogleda se u priloženim nedostacima i prednostima kod svake aplikacije. Zbog toga u nastavku rada predočeno je upoređivanje pronađenih prednosti i nestabilnosti kod navedenih aplikacija. Na kraju rada predviđeno je predlaganje novih i ažuriranje starih funkcionalnosti uz unapređenja korisničkog interfejsa radi poboljšanja aplikacija.

2. ANDROID OPERATIVNI SISTEM I UPOTREBLJIVOST ANDROID APLIKACIJA

Otvoren kod Android sistema baziran na *Linux OS* uz dostupnost, kako za izradu tako i za preuzimanje aplikacija, predstavlja prednost u odnosu na druge operativne sisteme [1].

2.1. Istorija razvoja Android operativnog sistema

Početak razvoja programerske platforme Android datira iz 2002. god. [1]. Razvoj Androida se povezuje sa kompanijom *Danger* čije su namere bile usmerene na razvoj prenosivih uređaja [1]. Uspešno realizovan proizvod, telefon pod nazivom *Sidekick*, privukao je pažnju osnivačima *Google-a* koji ga preuzima od 2005. godine [1].

2.2. Arhitektura i implementacija

Izvršavanje Android sistema moguće je osposobiti na hardveru na kom se izvršava *Linux* [1]. Prvobitna ideja Android OS je bila zasnovana na sistemu za mobilne telefone [1]. Softverska arhitektura Android operativnog sistema se može podeliti na *Linux* jezgro, radni sloj, programski okviri i aplikativni deo [2].

Pored navedenih radni sloj se može podeliti na biblioteke i radno okruženje (*ART*) [2]. Prednost koju pruža ovako organizovana arhitektura sistema jeste visok stepen modularnosti i enkapsulacije [2].

NAPOMENA:

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

2.3. Instalacija i primena

Aplikacije unutar Android OS mogu biti unapred ili naknadno instalirane upotrebom APK datoteke [3]. Preuzimanje APK fajla može se izvršiti preko pretraživača ili putem platformi koje služe za automatski instaliranje i pretragu digitalnog sadržaja i Android aplikacija [3]. Takva trenutno najzastupljenija platforma je *Google Play Store* koja spada u predefinisane aplikacije kod uređaja koji koriste *Google Mobile Services (GMS)* [3].

2.4. Dozvoljene mogućnosti prilikom dizajniranja aplikacije

Kada je reč o neophodnim delovima dizajna i kvaliteta aplikacije može se akcenat postaviti na analizu standarda Android vizuelnog dizajna i interaktivnih šablona. Prvenstveno podela ovih šablona može se sprovesti na osnovu polja na koja se odnose na navigaciju, notifikacije, korisnički interfejs i vizuelni kvalitet [4].

2.5. Ekspertske revizije upotrebljivosti

HCI je disciplina koja pokušava da objasni interakciju između ljudi i tehnologije, velik deo *HCI* je o tome kako treba praviti i raditi stvari, konkretno dizajnirati [5]. *HCI* heuristike imaju ulogu usmeravanja za projektovanje i izradu upotrebljivog i kvalitetnog dizajniranog korisničkog interfejsa kod kompjuterskih sistema [5]. Prilikom ispitivanja usmerenog heuristikama, proces zahteva mnogobrojna testiranja i rad sa određenim tehnologijama, iz tog razloga testiranju interfejsa neophodno je pristupiti na više načina. Ispitivanje interfejsa potrebno je izvesti upotrebom ekspertskih revizija kao što su evaluacija po heuristikama, revizija po smernicama, inspekcija konzistentnosti, *cognitive walkthrough* i formalna inspekcija utilitarnosti [6]. Analiza novinskih aplikacija podrazumeva evaluaciju po heuristikama i primenu sledećih Šnajdermanovih osam pravila:

- težiti konzistentnosti,
- omogućiti učestalim korisnicima upotrebu prečica,
- pružati informativne povratne informacije,
- projektovati dijaloge naglašene zatvorenosti,
- ponuditi prevenciju i rukovanje greškom,
- dozvoliti lako poništavanje efekata akcije, i
- redukovati opterećenje radne memorije [6].

3. ANALIZA ANDROID APLIKACIJA ZA VESTI

Analiza novinskih aplikacija realizovana je upotrebom evaluacije po heuristikama i Šnajdermanovih osam pravila na različitim verzijama Android uređaja. Najveći deo ispitivanja sproveden je na verzijama Android 6.0.1 i Android 8.1.0, ali po potrebi za određene segmente aplikacije korišćene su verzije Android 10 i 11.

3.1. The New York Times

The New York Times predstavlja dnevne novine osnovane 1851. god. u Sjedinjenim Američkim Državama, prvobitno publikovane u New York-u pod vlasništvom *The New York Times Company* [7].

Konzistentnost u kontekstu boja i fonta održana je kroz sve delove aplikacije. Jednostavnim i jasnim stilom bele podloge i crnog teksta asocira na prvobitna izdanja papirnih novina. Nedostatak konzistentnosti može se

primetiti kod prikazanih opcija za čuvanje i deljenje u različitim delovima aplikacije koji obavljaju istu ulogu prikaza liste artikala.

Zbog nepostojanja *widget*-a kod *The New York Times* aplikacije pravilo omogućene upotrebe prečica nije zadovoljeno. Međutim prečice koje se realizuju putem notifikacija implementirane sa više mogućnosti. Pored standardnog pristupa članku preko obaveštenja, aplikacija omogućava izvršavanje akcija čuvanja i deljenja. Na ovakav način širok spektar akcija može se izvršiti kraćim putem.

Povratne informacije upotrebom aplikacije ostavljaju pozitivan utisak i prisutne su u svakom delu aplikacije. Stoga ovo pravilo je ispunjeno i prilagođava se stepenu važnosti koji određena akcija nosi sa sobom.

Za projektovanje dijaloga u sklopu aplikacija za vesti nema mnogo mesta. Svodi se na niz aktivnosti i pojedinačne dijaloge najčešće u okviru podešavanja i prijavljivanja. Jednostavno određeno da korisnik bez dužih procesa izvrši željene akcije.

Kada je reč o proveru ponuđenih prevencija i rukovanja greškom pravilo je delimično zadovoljeno. Loš primer postoji kod validacije i formiranja komentara. Prostor za upis tekstualnog sadržaja na mestima lokacije i imena ili sadržaja komentara nije ograničen. Ovim propustom gde jedna komponenta prelazi preko druge narušava se dizajn korisničkog interfejsa. Primer koji ukazuje uspešnost kod ovog pravila može da se vidi kod sektora za komentare. Nakon ostavljanja praznih polja forme i potvrde slanja komentara, prikazuje se crveni tekst sa upozorenjem da polja ne mogu biti prazna.

Prilikom ispitivanja mogućnosti za poništavanje efekata akcije u obzir treba uzeti za funkcionalnosti za čuvanje tj. arhiviranje članka koje utiču na vizuelne promene korisničkog interfejsa. Poništavanje iste akcije se svodi na klik istog dugmeta, ponovljenim procesom prijavljena vest za kategoriju sačuvanih postaje objavljena. Sledeća situacija u kojoj je primenjen nešto drugčiji pristup poništavanja efekta odnosi se na segment komentara. Akcija prijave komentara ima jednostavan način poništavanja tako što se odabere dugme za preporuku. Sa druge strane ako je potreba korisnika da ukloni preporuku dovoljno je da ponovo stisne isto dugme.

Pored velike zatvorenosti aplikacije iz koje implicira nizak nivo kontrole korisnika, postoje neke stavke koje nisu obezbeđene u sklopu nje. Pravi primer je veliko odstupanje od uređivačkog režima u sklopu cele aplikacije. Ovakav režim podrazumevao bi skoro potpunu manipulaciju sadržajem koji korisnik želi da mu bude prikazan. Konkretno negodovanje određenih autora, tema i oblasti uz potpunu kontrolu korisnik bi mogao da napravi okruženje po svojoj želji.

Ispitivanje opterećenja radne memorije odnosi se na situacije u kojima je korisnik dužan da pamti podatke pri popunjavanju formi. U NYT aplikaciji postoje samo dva primera gde postoji potreba za čuvanjem već unetih podataka. Prilikom popunjavanja forme za komentara i prilikom slanja povratnih informacija uređivačima aplikacije potrebno je uneti ime koje u oba slučaja aplikacija sama zapamti nakon prvog unosa.

3.2. BBC News

BBC predstavlja javno finansiran sistem za emitovanje radio-televizije u Velikoj Britaniji, prvobitno nastao 1922. godine kao privatna kompanija [8]. BBC News predstavlja jedan poslovni deo *British Broadcasting Corporation* koji ima ulogu za prikupljanje i emitovanje vesti [8].

Detaljnou upotrebom i ispitivanjem svakog dela aplikacije jasno se može uočiti održavanje konzistentnosti boje i fonta u svakom segmentu. Kao narušavanje konzistentnosti uočava se propust kod vesti koje su integrisane sa istoimenog veb sajta. Iako u pokušaju da se što više prilagodi ostatku aplikacije dolazi do odstupanja u pogledu dizajna. Pored toga što je funkcionalnost uspešno implementirana i integracija određenih članaka sa veb sajta se ažurira iz minuta u minut ona predstavlja primer nekonzistentnosti unutar aplikacije.

Ispitivanje omogućenosti upotrebe prečica kod BBC News aplikacije zadovoljeno je u pogledu *widget*-a i delimično ispunjeno po pitanju notifikacija. Instalacijom aplikacije u paketu dolaze i dve različite vrste *widget*-a. Prva vrsta je lista vesti željene kategorije a druga pruža automatsko listanje pojedinačnih članaka iz selektovane kategorije. Drugi oblik prečice preko obaveštenja obezbeđen je samo za pristup članku. Dodatne opcije koje su mogle biti dostupne u sklopu pristiglog obaveštenja nisu implementirane.

Usluge informativnih povratnih informacija su kvalitetno obezbeđene u svakom segmentu aplikacije. Prilagođene su uticaju akcije. Ukoliko su akcije redovne postoji umerena naglašenost, dovoljna da korisnik bude svestan pokretanja akcije. Ukoliko je akcija od veće važnosti, postoji dodatna poruka koja obaveštava o izvršenoj radnji. Dijalozi su naglašene zatvorenosti i nisu čest primer rešavanja problema u BBC News aplikaciji.

Sloboda korisnika je u okviru ovog sistema u velikoj meri ograničena po pitanju rada, pisanja ili dodavanja stavke koje unapred nisu obrađene i predviđene od strane sistema. Iako je ponuđeno mnogo rukovanja i kontrole sa elementima unutar aplikacije, nije dozvoljen nikakav vid unosa. Jedini izuzetak se pronalazi prilikom pretrage, stoga aplikacija je zaštićena od većine unosa parametara koji bi na bilo koji način doveli do greške. Polje za pretragu predstavlja ujedno i jedini propust gde broj karaktera za unos nije ograničen.

Primeri koji se mogu navesti za izvršavanje i poništavanje određenih efekata najviše se svode na uređivanje sektora za korisnikove vesti ili na prilagođavanje podešavanja sa korisnikovim potrebama. Uređivanje spomenutog sektora obezbeđuje dodavanje ali i brisanje. Takvim pristupom svaka izvršena akcija može biti poništena.

Korisnik može da prilagodi tabove koje želi da gleda kada uđe u aplikaciju. Tabovi predstavljaju teme po kojima su članci grupisani. Prvih pet tabova je predefinisano i podrazumeva se da ih svaki korisnik vidi sa sadržajem koji pruža aplikacija. Naredni tabovi se dodaju po korisnikovoj volji. Takvim mogućnostima stiče se određena kontrola nad aplikacijom.

U BBC News aplikaciji broj situacija u kojima je potrebno pamćenje informacija, u koracima za izvršavanje određenih akcija, sveden je na nulu.

3.3. B92

B92 osnovana je prvobitno kao omladinska radio stanica 1989.god. sa središtem u Beogradu, vremenom prerasta u kompaniju koju 2015. godine preuzima *Antena grupa* [9].

Ispitivanjem konzistentnosti nailazi se na nekoliko propusta prilikom prilagođavanja boje sa različitim temama. Još jedan primer primećuje se prilikom izmene fonta putem omogućenih podešavanja. Promena nije prilagođena na nivou aplikacije već samo na njenim segmentima, konkretno primena postoji samo na detaljnom prikazu članka.

Prečice u sklopu aplikacije realizovane su putem *widget*-a, ali i putem notifikacija. Četiri vrste *widget*-a dozvoljavaju korisniku mnogo izbora prilikom pristupa aplikaciji i njenim konkretnim oblastima. Obaveštenja obavljaju osnovnu ulogu direktnog pristupa novom članku bez dodatnih akcija koje bi mogle biti implementirane u sklopu ove prečice.

B92 aplikacija ima kvalitetno implementiran deo povratnih informacija prilagođen stepenu važnosti akcije. Slična situacija kao i kod povratnih informacija, projektovanje dijaloga naglašene zatvorenosti je zadovoljeno kroz mali broj primera u okviru podešavanja gde se može na njih naići.

Prilikom testiranja aplikacije nailazi se na velik broj grešaka koje mogu biti kritične po rad aplikacije. Pristup zajedničkom video sadržaju aplikacije rezultuje prekidom rada. Istu situaciju imamo u određenim situacijama prilikom uređivanja kanala notifikacija. Postoje i greške koje rezultuju sa blažim posledicama gde je takođe glavni primer prilikom odabira kanala obaveštenja koji ne čuva korisnikov odabir.

Neuspešno rukovanje greškom vidi se prilikom ocenjivanja komentara koje može biti ponovljeno neograničen broj puta. Isti primer odnosi se na neispunjeno pravilo poništavanja efekata akcije. Nakon ocenjivanja ne postoji mogućnost da se korisnik predomisli.

Kontrola od strane korisnika nije obezbeđena u dovoljnoj meri, korisnik nema mogućnosti u sklopu aplikacije da prilagodi kategorije ili artikle koje želi da čita.

Ispitivanje opterećenja radne memorije pokazalo se kao uspešno jer prilikom pisanja većeg broja komentara, informacije o korisniku ostaju sačuvane.

3.4. Upoređivanje prednosti i nedostataka NYT, BBC News i B92 aplikacije

Upoređivanje aplikacija može biti izvedeno kroz nekoliko karakteristika i segmenata. Pristupačnost aplikacije je jednaka kod BBC i B92, međutim odstupanje postoji kod *NYTimes*-a gde određeni segmenti zahtevaju pretplatu. Korisnički interfejs varira u zavisnosti od potreba korisnika. Ne toliko intuitivni prikaz postoji kod NYT vesti zbog specifične organizacije *toolbar*-a i opcija u sklopu njega. Prednost koja se ističe samo u okvirima BBC-a odnosi se na bočni meni koji olakšava pristupačnost određenim delovima i utiče na povećavanje efikasnosti upotrebe aplikacije. Funkcionalnost notifikacija ističe se sa dodatnim opcijama za čuvanje i deljenje kod *NYTimes*-a. *Widget*-i vrhunac realizacije ostvaruju u sklopu B92 aplikacije sa više mogućnosti i dimenzija.

Kod BBC News aplikacije nedostatak komentara i njihovog unosa čini aplikaciju bezbednijom zbog izbegavanja različitih formi za unos, ali u isto vreme čini veliki

nedostatak funkcionalnosti. Najviše susreta sa greškama i lošim upravljanjem primećuje se u sklopu B92 vesti.

3.5. Predlog unapređenja korisničkog interfejsa i funkcionalnosti

NYT zadovoljava mnogobrojne funkcionalnosti iako kao glavni nedostatak nameće se dostupnost jednog ili više widget-a. Korisnički interfejs ove aplikacije moga bi se unaprediti drugčijim dizajnom kroz tabove ili upotrebom bočnog menija. Kao i kod BBC aplikacije, NYT korisnički interfejs postigao bi napredak omogućavanjem tamnog režima rada. BBC News korisnički interfejs ne zahteva dodatne izmene osim prilagođavanja integrisanih vesti u segmentu koji će biti u skladu sa ostatkom aplikacije. Sektor za komentare predstavlja glavnu funkcionalnost koja nedostaje u BBC aplikaciji. Pored toga dodatne akcije na notifikacijama bi doprinele efikasnijem radu sa aplikacijom. Unapređenje B92 korisničkog interfejsa moglo bi se postići uprošćavanjem navigacije kroz aplikaciju implementacijom bočnog menija. Po pitanju funkcionalnosti potrebna su mnogobrojna ispravljanja grešaka koja utiču na prekid aplikacije i na njeno neočekivano ponašanje. Funkcionalnost za upravljanje prikazanim kategorijama predstavlja jedan od većih nedostataka B92 aplikacije.

4. ZAKLJUČAK

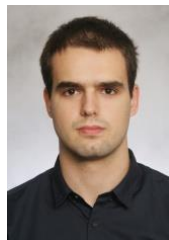
Detaljnim ispitivanjem i upoređivanjem uočava se nezadovoljavanje ili delimično zadovoljavanje mnogih heuristika od strane sve tri aplikacije. Propusti su pronađeni različitim vrstama testiranja i detaljno se opisuju u analizi rada. U radu je priloženo sve od zapažanja malih propusta poput boje i veličine slova do otkrivanja grešaka koje su kritične za rad aplikacije i nedostataka celih komponenti koje su neophodne za efikasniji rad. Završnom analizom i poređenjem aplikacija primećuje se najveća zatvorenost ali ujedno i bezbednost kod BBC News aplikacije. Zbog toga najmanji broj nepravilnosti je primećen u radu sa tom aplikacijom. Slična situacija sa umereno većom podložnošću grešaka pokazala se kod The New York Times aplikacije. Jednostavnog i tradicionalnog dizajna aplikaciju odlikuju mnogobrojne mogućnosti u odnosu na druge dve. Na kraju ispitivanje B92 aplikacije ukazalo je na mnogo propusta i dozvoljenih grešaka koje bi trebalo ispraviti. Pored nedostataka ili loše implementiranih segmenata, aplikacija se ističe sa pristupačnim i kvalitetnim dizajnom. Uz prilagođeni korisnički interfejs koji se pokazao bez značajnih kritika, dodatni režim rada i prečice sa početnog ekrana odlikuju ovu aplikaciju.

Navedeni nedostaci i sugestije redovnim ažuriranjem i prilagođavanjem sistema mogli bi se lako u relativno kratkom vremenskom roku otkloniti. Poslednje poglavlje ukazuje na neke slabosti koje bi prvenstveno trebalo popraviti ili implementirati. Na taj način obezbediti podršku aplikacijama i steći poverenje korisnika. Uklanjanje grešaka rezultovalo bi većom upotrebljivošću i efikasnijim radom sa aplikacijom.

5. LITERATURA

- [1] I. Pap, N. Lukić, *Projektovanje i arhitekture softverskih sistema: Sistemi zasnovani na Androidu*, 1. Izd, Novi Sad: Fakultet tehničkih nauka, 2015.
- [2] N. Lukić, *Predlog proširenja Android operativnog sistema servisima digitalne televizije*, Novi Sad: Fakultet tehničkih nauka, 2014.
- [3] Android Headlines, „What Is Google Play Store? Everything You Need To Know“, <https://www.androidheadlines.com/what-is-google-play-store.html> (pristupljeno u februaru 2021.)
- [4] Android Developers, „Core app quality“, <https://developer.android.com/docs/quality-guidelines/core-app-quality> (pristupljeno u februaru 2021.)
- [5] A. Dix, J. Finlay, G. D. Abowd, R. Beale, *Human-Computer Interaction*, 3rd Edition, Harlow: Pearson Education Limited, 2004.
- [6] Dragan Ivetić, „Interakcija čovak računar“, skripta sa predavanja, FTN, 2016.
- [7] Britannica, „The New York Times American newspaper“, <https://www.britannica.com/topic/The-New-York-Times> (pristupljeno u martu 2021.)
- [8] Britannica, „British Broadcasting Corporation“, <https://www.britannica.com/topic/British-Broadcasting-Corporation> (pristupljeno u martu 2021.)
- [9] B92, „Informacije o B92“, https://www.b92.net/o_nama/ (pristupljeno u martu 2021.)

Kratka biografija:



Stefan Milanović rođen 16.09.1995. godine u Sremskoj Mitrovici. Završio srednju školu Gimnaziju „Sava Šumanović“ u Šidu 2014. godine. Osnovne studije na odseku Računarstvo i automatika na Fakultetu tehničkih nauka u Novom Sadu završio je 2018. godine. Master rad na studijskom programu Primenjene računarske nauke i informatika – Elektronsko poslovanje odbranio je 2021.god.
kontakt:

milanovicstefann@gmail.com

KONFIGURISANJE I TESTIRANJE MIKROPROCESORSKOG RELEJA ZA ZAŠTITU ENERGETSKOG TRANSFORMATORA**CONFIGURATION AND TESTING OF MICROPROCESSOR RELAY FOR POWER TRANSFORMER PROTECTION**Ninoslav Subotić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu je predstavljen primjer zaštite energetskeg transformatora. Navedeni su osnovni principi rada relejne zaštite i opisane su osnovne zaštite koje se koriste u šticeњу energetskeg transformatora. U radu je korišćen jedan tip mikroprocesorskog releja i uređaj za sekundarno ispitivanje. Nakon detaljnog objašnjenja zaštita, praktičnim radom funkcije su ispitane odgovarajućim testnim alatom koji je korišten za sekundarno injektovanje struja. Na osnovu rezultata urađenih testova i funkcionisanja releja došlo se do određenih zaključaka.

Ključne reči: Relejna zaštita, Mikroprocesorski relej, Elektroenergetski transformator

Abstract – This document presents the basic principles of relay protection of power transformers. The basic protection functions used in the protection of power transformers are listed and described. One type of microprocessor relay and one secondary injection tool are used in test setup. Based on comparative analysis of the feedback in both the relay and the test tool obtained during practical tests, certain conclusions were presented.

Keywords: Relay protection, Microprocessor relay, Power transformer

1. UVOD

Osnovni cilj elektroenergetskog sistema jeste da se obezbjedi dovoljne količine kvalitetne električne energije potrošačima svih kategorija. Pored toga treba da se obezbjedi neprekidnost u napajanju i proizvodnji električne energije uz što je moguće manje troškove, elektrane, prenosna mreža i transformatorske stanice povezuju se u jedinstven elektroenergetski sistem, kako unutar manjih regija tako i unutar država pa i između više država.

Da bi elektroenergetski sistem mogao da funkcioniše potrebno je obezbjediti zaštitu svakog njegovog elementa (generator, transformator, vod, motor), a takođe i pojedinih cjelina u okviru elektroenergetskog sistema od kvarova. Zaštita cjelokupnog elektroenergetskog sistema od kvarova se ostvaruje relejnom zaštitom [1].

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je dr Duško Bekut, red. prof.

Oblast relejne zaštite predstavlja jednu od najvažnijih oblasti elektroenergetike koja treba da pruži adekvatnu zaštitu elemenata sistema, a posredno i zaštitu rukovaoca tim elementima. Iz tih razloga je neophodno da uređaji relejne zaštite budu ispravni i pouzdani, da bi u svakom trenutku mogli da odreaguju na unaprijed definisan način. U nastavku rada u drugom poglavlju dati su osnovni principi, svojstva i opšti zahtjevi koji su postavljeni pred relejnu zaštitu i koji je osnovni cilj relejne zaštite.

U trećem poglavlju govori se uopšteno o energetskom transformatoru kao uređaju i koja je svrha u njegovom korišćenju u elektroenergetskom sistemu, kao i dijelovi energetskog transformatora. Pored toga se govori i o kvarovima koji nastaju u energetskom transformatoru i vrste zaštita od tih istih kvarova i detaljno objašnjenje određenih zaštita koje su kasnije i testirane.

U četvrtom poglavlju urađeno je testiranje. Objašnjeni su uređaji koji su bili potrebni prilikom testiranja, a prikazani su i rezultati urađenih testova u određenom softveru uređaja za testiranje kao i u softveru releja.

Peto poglavlje ujedno i posljednje, a prikazuje kratak opis i zaključak u kom se obrazlaže ovakav vid testiranja.

2. RELEJNA ZAŠTITA

S obzirom na veliku vrijednost šticeđenih objekata, kao i na značaj normalnog snadbijevanja potrošača električnom energijom. Relejna zaštita ima vrlo važnu ulogu u pogonu elektroenergetskog sistema kao dio automatike elektroenergetskih postrojenja za proizvodnju, prenos i distribuciju električne energije.

Cilj primjene relejne zaštite je najbrže moguće isključenje elementa i/ili dijela elektroenergetskog sistema sa kvarom uz očuvanje funkcionalnosti ostalog dijela sistema [1].

Prema tome zadatak relejne zaštite elektroenergetskog postrojenja je da trajno nadzire karakteristične električne ili druge veličine (struju, napon, temperaturu) šticeđenog objekta i da u slučaju kvara ili opasnog pogonskog stanja automatski preduzme sve potrebne mjere da se kvar izbjegne ili da se svedu na minimum njegove posledice ako se već pojavio, kao i da o tome obavijesti pogonsko osoblje. Opšti zahtjevi koji se postavljaju pred relejnu zaštitu su [1]:

- Selektivnost.
- Brzina reagovanja.
- Osjetljivost zaštite.
- Pouzdanost.
- Jednostavnost.
- Ekonomičnost.

Najčešći kvarovi u elektroenergetskom sistemu su kratki spojevi (jednostruki, a potom i dvostruki), a zatim su to i prekidi provodnika, oštećenje elemenata sistema.

Kod kratkih spojeva su najčešći jednopolni kratki spojevi (70% do 95% svih jednostrukih kratkih spojeva), zatim dvopolni kratki spojevi (sa i bez spoja sa zemljom), dok su tropolni kratki spojevi veoma rijetki [1].

Digitalni releji predstavljaju poslednju generaciju zaštitnih releja. Digitalni relej kao hardverski uređaj u sebi objedinjuje zaštite, logičke, mjerne, kontrolne i monitoring funkcije [1].

3. ZAŠTITA ENERGETSKOG TRANSFORMATORA

Transformator je statički uređaj koji pomoću elektromagnetne indukcije pretvara jedan sistem naizmjeničnih struja u jedan ili više sistema naizmjeničnih struja iste učestanosti i obično različitih vrijednosti napona i struja. Pri svakoj transformaciji energije javljaju se gubici. Step en iskorišćenja energije kod energetskih transformatora (ET-a) je vrlo visok, što sa jedne strane omogućava tehnički opravdan prenos električne energije, a sa druge strane ih uvrštava u klasu najbojnijih električnih mašina u elektroenergetskom sistemu [2].

Kvarovi na transformatorima nastaju kao posledica slabljenja i oštećenja izolacije. Ta oštećenja mogu biti izazvana naprežanjem električne prirode (nastaju kao posledica atmosferskih i pogonskih prenapona u mreži), mehaničke prirode (izazvane dinamičkim silama u namotima) ili kao posledica prevelikih zagrevanja. Pomenuta naprežanja dovode do polakog i neminovnog starenja i slabljenja izolacije i do pojave kvarova.

Kada su u pitanju kvarovi kod transformatora postoji težnja za brзом eliminacijom kvara jer postoji opasnost od pucanja transformatorskog suda – kotla i paljenja ulja [1].

Kod transformatora se sreću slijedeći kvarovi:

- kratki spojevi između namota transformatora,
- kratki spojevi između navojaka iste faze,
- kratki spojevi sa zemljom,
- lokalna tinjanja izolacije.

Najčešće se događaju preskoci na izvodima transformatora i kratki spojevi između navoja iste faze. Kratki spojevi na izvodima transformatora nastaju kao posledica proboja ili zaprašnja provodnih izolatora. Kratki spojevi između navojaka iste faze nastaju zbog slabljenja izolacije izazvanog strmim prenaponskim talasima. Relativno su česti i proboji prema uzemljenim dijelovima, dok su kratki spojevi između namota rijetki [1].

Kao zaštite od kvarova koriste se sledeće zaštite:

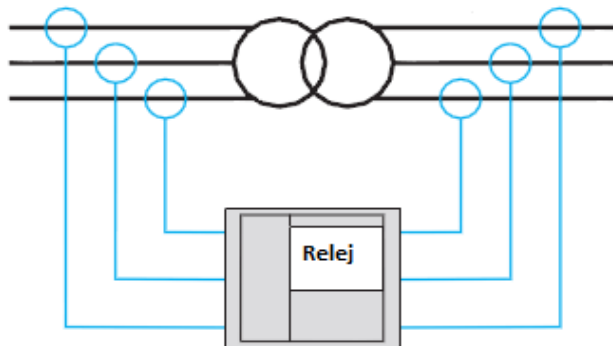
- diferencijalna zaštita transformatora,
- gasna zaštita (Buchholz relej),
- zemljospojna zaštita,
- trenutna prekostrujna,
- distantna,
- osigurači.

Kod transformatora, pored kvarova, postoje i opasna pogonska stanja. Opasna pogonska stanja su [1]:

- spoljašnji kratki spojevi,
- preopterećenje,
- isticanje ulja iz kotla.

3.1. Diferencijalna zaštita transformatora

Diferencijalna zaštita predstavlja osnovnu (temeljnu) zaštitu energetskog transformatora (ET-a). Danas ona predstavlja najčešći tip zaštite transformatora. Obavezna je za transformatore iznad 8MVA, za manje transformatore se rijeđe koristi zbog visoke cijene – slika 1.



Slika 1. Prikaz povezivanja releja na mrežu u izvedbi diferencijalne zaštite[3]

Diferencijalna zaštita mora djelovati isključivo u slučaju kvarova unutar svoje zaštitne zone (kvarovi između strujnih transformatora na koje je priključena). Riječ je o kvarovima u samom šticeu objektu (energetskom transformatoru) i kvarovima na primarnim spojnim vodovima na potezu od energetskog transformatora do strujnih transformatora s obje strane. U slučaju kvarova izvan šticeu zone, diferencijalna zaštita ne smije djelovati [4].

Diferencijalnom zaštitom se transformator štiti od:

- međufaznih kratkih spojeva u ET-u ili na izvodima iz ET-a,
- faznih kratkih spojeva,
- jednofaznih kratkih spojeva ako je zvjezdšte transformatora direktno uzemljeno.

Osnovni element diferencijalne zaštite energetskog transformatora jeste stabilizacioni diferencijalni relej. On djeluje na temelju poređenja struja koje ulaze i izlaze iz šticeu objekta (zaštitne zone). Sastoji se od:

- stabilizacione grane i
- diferencijalne grane.

Primjenjujući princip diferencijalne zaštite na transformatore, treba uzeti u obzir [4]:

- korekciju mogućeg faznog pomjeraja na faznim namotajima (usklađivanje faznog pomjeraja),
- efekat načina uzemljenja i sprege (filtriranjem nultih komponenti struje),
- korigovanje mogućeg neslaganja veličina struja iz strujnih transformatora (korekcija odnosa transformacije),
- posledica udarne struje tokom uključenja transformatora,
- moguću prepobudu.

3.2. Prekostrujna zaštita energetskog transformatora

Uz diferencijalnu zaštitu energetskog transformatora koja je ujedno i njegova osnovna zaštita, još se može koristiti i prekostrujna zaštita, kao rezervna (eng. „back-up“) zaštita. Ona mora u svakom slučaju djelovati s određenim vremenskim zatezanjem (vremenskim kašnjenjem) da

dopusti djelovanje diferencijalnoj zaštiti, koja djeluje trenutno (bez vremenskog kašnjenja, tj. s vlastitim vremenom djelovanja relejnog uređaja uvećanim za vrijeme djelovanja samog prekidača, što zajedno iznosi oko 100ms).

Ukoliko diferencijalna zaštita energetskog transformatora zakaže iz bilo kog razloga, tada prekostrujna zaštita treba isključiti energetski transformator [4].

Koriste se pritom najčešće sljedeće prekostrujne zaštite:

- niskopodesiva prekostrujna zaštita (IEC $I>$, ANSI 51);
- visokopodesiva (trenutna) prekostrujna zaštita (IEC $I>>$, ANSI 50).

Trenutna prekostrujna zaštita ($I>>$) će djelovati pri nastajanju međufaznih kvarova unutar osnovne zone štice, koju u ovom slučaju predstavlja sam energetski transformator, ovom zaštitom se eliminišu međufazni kratki spojevi na priključcima transformatora i u transformatoru [1].

3.3. Zaštita od kratkih spojeva sa zemljom

Diferencijalna zaštita nije dovoljno osjetljiva da detektuje unutrašnji jednofazni kratak spoj ukoliko se kvar nalazi u blizini neutralne tačke transformatora ili ako je struja zemljospoja ograničena [5].

U mreži u kojoj su zvezdišta transformatora uzemljena preko otpora za ograničenje struje jednopolnog kratkog spoja primjena diferencijalne zaštite nije efikasna.

Može se desiti da struje kratkih spojeva budu istog reda ili čak manje od veličine struje pri kojoj se djeluje diferencijalnom zaštitom, pa postoji opasnost da jedan dio kratkih spojeva ne bude detektovan.

To se posebno odnosi na kvarove koji nastaju bliže zvjezdistu transformatora [1].

3.4. Gasna zaštita

Kod transformatora kod kojih postoji dilatacioni sud sa konzervatorom može se primjenjivati gasna zaštita sa Buchholz relejom. Princip rada ove zaštite zasniva se na tome da pri pojavi kvarova (koji su uvijek praćeni lukom) unutar kotla dolazi do intenzivnog nastanka gasova i povećanja pritiska u kotlu.

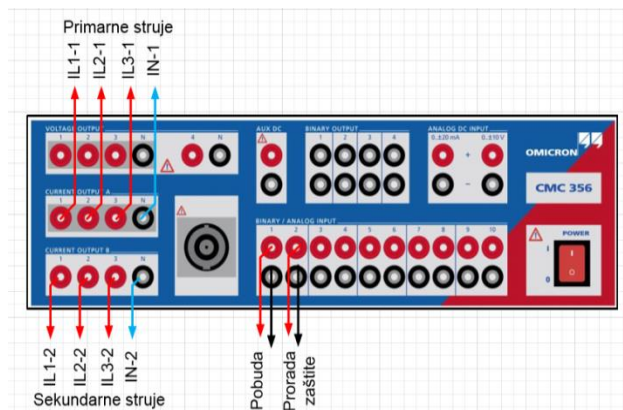
Interesantno je napomenuti da se sastav gasova razlikuje u zavisnosti od vrste kvara.

Ovaj tip zaštite predstavlja izuzetno jednostavnu i pouzdanu zaštitu i primjenjuje se kod gotovo svih uljin transformatora (ova zaštita se ne koristi kod transformatora snaga manjih od 160kVA). Pri pojavi kratkih spojeva vrijeme djelovanja ovim relejima je oko 0.1s [1].

4. TESTIRANJE ZAŠTITNIH FUNKCIJA

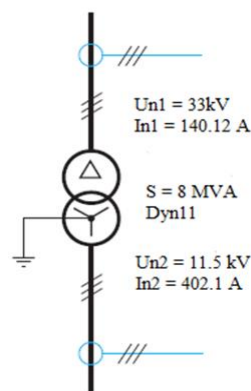
U ovom radu ispitane su zaštite: diferencijalna zaštita, prekostrujna, zaštita i zaštita od kratkih spojeva sa zemljom energetskog transformatora. Za testiranje je korišćen ispitni alat za sekundarno injektovanje struje i napona – slika 2.

Ispitni alat je korišćen za injekciju trofaznih struja, za praćenje izlaza releja koji signalizira pobudu i proradu zaštite.



Slika 2. Uređaj za injektovanje električnih veličina

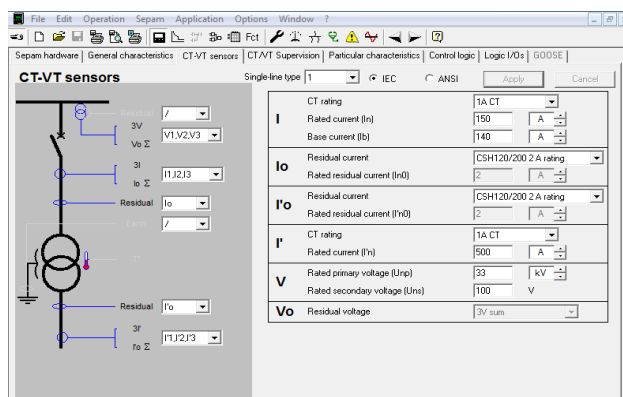
U ovom radu testira se zaštita energetskog transformatora kao štice objekta i za potrebe testiranja koristiće se energetski transformator sa karakteristikama prikazanim na slici 3.



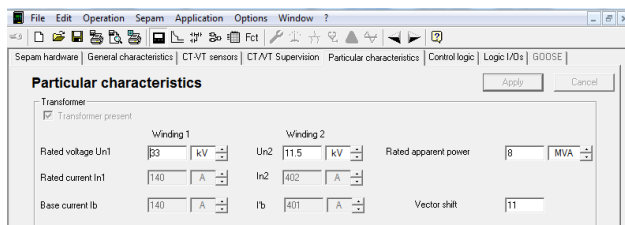
Slika 3. Osnovni parametri energetskog transformatora [3]

Nominalna snaga transformatora (S) je 8MVA, namotaj 1 sa visokonaponske strane ET-a je 33kV, a namotaj 2 sa niskonaponske strane ET-a je 11,5kV, a sprega transformatora je Dy11. Prenosni odnos ST1 na 33kV strani je 150/1, a prenosni odnos ST2 na 11,5kV je 500/1.

U numeričkim relejima u okviru podešavanja konfiguracije štice objekta, nalazi se podešavanje prenosnih odnosa strujnih transformatora čiji su sekundarni krajevi povezani na strujne kartice releja – slike 4 i 5.



Slika 4. Podešavanje osnovnih karakteristika strujnih i naponskih mjernih transformatora



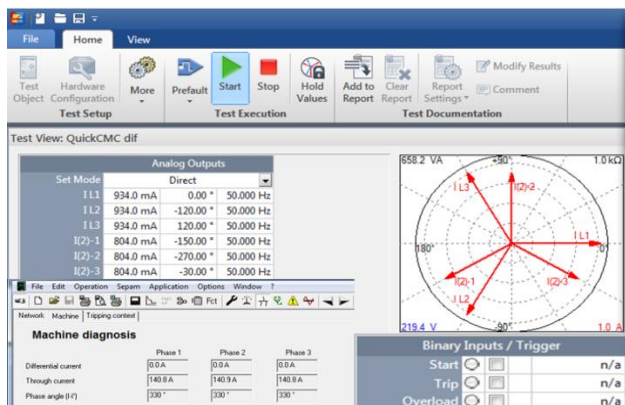
Slika 5. Podešavanje osnovnih karakteristika šticenog objekta

Prva i osnovna stvar koja je neophodna na samom početku procesa testiranja releja jeste provjera sekundarnih strujnih krugova i provjera tačnosti očitavanja mjerenja releja prilikom injekcije sekundarnih struja.

Prilikom testiranja određene zaštite neophodno je prvo podesiti odgovarajuće parametre zaštite koja se ispituje kao softveru releja tako i u softveru uređaja za ispitivanje.

Test stabilnosti diferencijalne zaštite se sprovodi da bi se utvrdilo da li je diferencijalna zaštita stabilna, diferencijalna zaštita ne bi trebalo da prodadi, kod normalno balansiranih opterećenja.

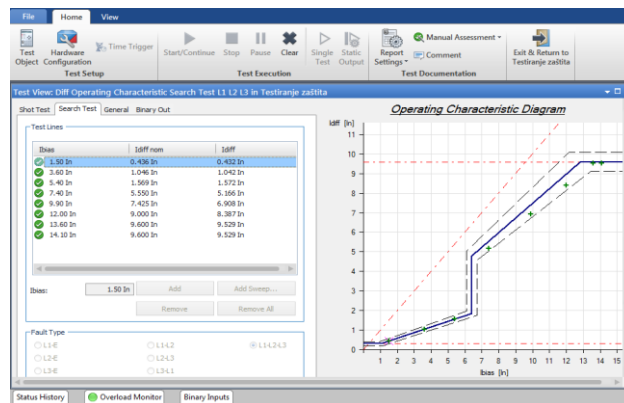
Na ovaj način se provjerava i ispravnost povezanih strujnih transformatora. Na osnovu vrijednost sekundarnih struja koje su injektovane i faznog pomjeraja tih struja, kao rezultat ovog testa očekuje se da diferencijalna struja bude 0, dok će postojati stabilizaciona struja koja bi trebalo da bude najveća od svih pojedinačnih struja. Sa slike 6. se vidi da zaštita nije proradila, da je stabilna.



Slika 6. Stabilnost diferencijalne zaštite

Test određivanja diferencijalno-stabilizacione karakteristike se vrši kako bi se pokazalo da li diferencijalna zaštita transformatora releja radi kako je podešeno odnosno da li podešenje zaštite u softveru releja odgovara stvarnom radu releja. Odnosno, da li će zaštita odreagovati kada je to potrebno i da li će ostati stabilna i neće odreagovati kada ne postoji kvar u šticenoj zoni. Ograničenje prorade zaštite je diferencijalno-stabilizaciona kriva, pa se ovim testom ispituje podešena kriva.

Kriva mora da se podesi i u softveru releja i u softveru uređaja za testiranje, kako bi se prepoznala kriva u softveru uređaja za testiranje, parametri krive moraju da se podudaraju u oba softvera, time se omogućava ispitivanje krive koju je korisnik podesio – slika 7.



Slika 7. Traženje diferencijalno-stabilizacione karakteristike

5. ZAKLJUČAK

Cilj ovog rada bio je da se uradi konfigurisanje i testiranje mikroprocesorskog releja za zaštitu energetskog transformatora. Da bi se to moglo realizovati, bilo je neophodno pružiti detaljan opis funkcionisanja određenih zaštita i funkcionisanja uređaja, odnosno njegovih zaštitnih funkcija. Nakon toga se izvršilo testiranje. Kako bi se uređaj što bolje testirao urađeni su testovi za sve vrste kvarova (jednopolne, dvopolne i trojpolne).

S obzirom na to da su sve dobijene vrijednosti u dobrom mjernom opsegu i da su svi testovi zadovoljili određene kriterijume, izvodi se zaključak da relej u potpunosti radi ispravno i da bi se u stvarnom pogonu prilikom pojava određenih kvarova ponašao baš onako kako se to od njega zahtijeva.

6. LITERATURA

- [1] Duško Bekut, „Relejna zaštita“, Fakultet Tehničkih nauka, Novi Sad, 2009.
- [2] Jovan Nahman, Vladica Mijailović, „Razvodna postrojenja“, Beograd, 2015.
- [3] Sepam, „Protection, metering and control functions“, User's manual, 2016.
- [4] Petar Sarajčević i Robert Kosor, „Zaštita u elektroenergetskom sustavu“, Sveučilište u Splitu, 2020.
- [5] Jelisaveta P. Krstivojević, „Digitalna zaštita energetskih transformatora od unutrašnjih kvarova“, Elektrotehnički fakultet, Beograd, 2015.

Kratka biografija:



Ninoslav Subotić rođen je u Vukovaru 1994. godine. Osnovne studije završio je na Fakultetu tehničkih nauka 2018. godine iz oblasti Elektrotehnika i računarstvo smjer Elektroenergetski sistemi. Master studije upisao je 2018. godine na istom fakultetu smjer Elektroenergetika – Električne mašine i energetska elektronika.

Kontakt: ninoslav.subotic1@gmail.com

**PLC ПРОГРАМ ЗА УПРАВЉАЊЕ ЕЛЕКТРОМОТОРНИМ ПОГОНИМА ВЕЛИКЕ
CHAGE****PLC PROGRAM FOR CONTROL OF HIGH POWER ELECTRIC MOTOR DRIVES**

Илија Гелић, Дејан Јеркан, *Факултет техничких наука, Нови Сад*

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У овом раду детаљно су обрађена три електромоторна погона топлане на природни гас. Описан је начин избора одговарајућег фреквентног регулатора, сензора и програмабилног логичког контролера за сваки електромоторни погон посебно. Такође, у практичном делу рада формиран је и образложен лествичасти дијаграм за управљање електромоторним погонима топлане на природни гас.

Кључне речи: Асинхрони мотор, Фреквентни регулатор, Сензори, PLC, Siemens S7.

Abstract – In this paper, three electric motor drives of a nature gas heating plant are discussed in detail. The method of selecting of appropriate frequency converter, sensors and PLC for each electric motor drive is described. Also, in the practical part of the paper, a ladder diagram for controlling the electric motor drives of a natural gas heating plant was formed and explained.

Keywords: Induction motor, Frequency converter, Sensors, PLC, Siemens S7.

1. УВОД

PLC контролери уводе се у индустрију због повећања поузданости и сигурности производног процеса, такође због смањења трошкова производње уз обезбеђивање високог квалитета производа и могућност брже преоријентације производње према захтевима потрошача и тржишта.

У овом раду обрађена су три практична примера савременог електромоторног погона топлане на природни гас. Електромоторни погони који се разматрају су: вентилатор свежег ваздуха котла, вентилатор рецикулације димних гасова и рецикулациона пумпа котла.

Поглавље три представља теоријске основне структуре и принцип рада фреквентног регулатора, од самог почетка па до садашњости. Такође, представиће се како се врши подешавање параметара фреквентног регулатора.

Затим, у поглављу четири описани су програмабилни логички контролери на које се ставља посебан фокус, јер они представљају мозак сваког индустријског процеса. Представиће се основни делови и принцип

рада PLC, као и предност њихове примене у савременим електромоторним погонима.

Као што је већ речено тема овог рада су три независна електромоторна погона, тако да се практичан део рада фактички састоји из три целине, задатка. Сваки задатак пред собом има следеће захтеве: избор адекватног фреквентног претварача и подешавање параметара, одабир и подешавање сензора за мерење величина асинхроног мотора и избор и програмирање PLC-а, односно писање управљачког лествичастог дијаграма за сваки погон посебно.

**2. ЕЛЕКТРОМОТОРНИ ПОГОНИ У ТОПЛАНИ
НА ПРИРОДНИ ГАС****2.1. Вентилатор свежег ваздуха котла**

Вентилатор свежег ваздуха котла покреће асинхрони мотор, а његову брзину обртања контролише и задаје фреквентни регулатор који је преко комуникације повезане са програмабилним логичким контролером. PLC надзире све мерене величине и параметре и на основу добијених резултата шаље сигнал фреквентном регулатору шта да уради са брзином обртања асинхроног мотора. Конкретно мисли се на следеће, ваздух за сагоревање природног земног гаса усисава се из вентилаторског простора, где је смештен вентилатор свежег ваздуха котла и то тако што се усисава кроз решетке на фасади просторије. Зависно од оптерећења котла регулише се рад вентилатора и количина ваздуха која се убацује у ложиште котла.

2.2. Вентилатор рецикулације димних гасова

Вентилатор рецикулације димних гасова, такође покреће асинхрони мотор чијом брзином обртања управља фреквентни регулатор, а фреквентни регулатор команду добија од PLC-а који управља целокупним погоном. Да би се у котлу омогућило континуално одвијање процеса сагоревања, потребно је осим поменутих процеса довођења горива и убацивања ваздуха у ложиште котла, одводити из котла димне гасове. Димни гасови настају као нежељено дејство приликом сагоревања природног гаса и уколико би дошло до нагомилавања поменутих гасова, односно ако не би било вентилатора рецикулације димних гасова који има задатак да их избаци из ложишта котла, приликом поновног паљења котла дошло би до експлозије у самом котлу.

2.3. Рецикулациона пумпа котла

Задатак рецикулационе пумпе котла јесте да „гура“ воду кроз котла, односно кроз сложен систем грејача

НАПОМЕНА:

Овај рад је настао на основу мастер рада чији је ментор био др Дејан Јеркан, доцент.

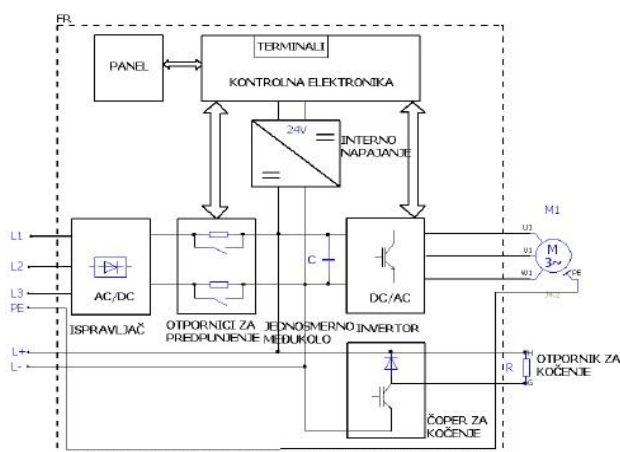
што за rezultata даје повећање температуре воде на жељену вредност која је задата преко PLC-a. Након достигнуте температуре воде која је задата преко PLC-a. Веома је битно да рецикулациона пумпа котла буде поуздана и да технолози који раде у топлани знају у сваком тренутку шта се дешава у самом котлу. Уколико би се десило да вреловодни котао приликом свог рада, односно када су укључени грејачи дужи временски период нема доток воде у систем, односно из неког разлога рецикулациона пумпа не ради исправно, врло лако може доћи до оштећења котла због превелике температуре или у најгорем случају до хаварије.

3. ФРЕКВЕНТНИ РЕГУЛАТОР

Развојем и падом цена компоненти енергетске електронике постало је могуће ефикасно и јефтино регулисати брзину обртања асинхроног мотора помоћу енергетских претварача фреквенције и напона. Фреквентни регулатори су електронски уређаји који омогућавају управљање брзином обртања и регулацију асинхроних трофазних и синхроних мотора. Брзина асинхроног мотора је пропорционална фреквенцији примењеног напона па је из тог разлога за промену брзине потребно мењати фреквенцију, што за последицу има промену напона, а ту могућност нам пружа фреквентни регулатор. Фреквентни претварачи претварају мрежни напон константне амплитуде и фреквенције у излазни напон жељене амплитуде и фреквенције. Такође, флуks у машини је пропорционалан односу амплитуде и фреквенције улазног напона тако да је за контролу флуksа и момента потребно уз фреквенцију мењати у амплитуду напона.

3.1. Главни елементи фреквентног регулатора

На наредној слици (1) приказана је интерна структура фреквентног регулатора, коју чине; исправљач, једносмерно међуколо, инвертор и електронско управљачко коло.



Слика 1. Интерна структура претварача

Исправљач је прва компонента енергетског претварача и директно је прикључен на мрежни напон константне амплитуде и фреквенције. Могу бити конструисани за једнофазни и трофазни напон, али и овом мастер раду акценат стављамо на трофазне. На основу ове чињенице може бити неуправљив (диода),

полууправљив (комбинација диода и тиристора) и управљив (тиристор).

Једносмерно међуколо можемо представити као складиште из кога мотор „извлачи“ енергију кроз инвертор. Служи за стабилизацију и прилагођавање исправљене величине на излазу исправљача ка инвертору. Међуколо може бити изведено на два начина; напонски и струјно.

Инвертор је последњи елемент енергетског претварача пре мотора и позиција где се одвија завршно прилагођавање излазног напона. Главни елементи инвертора су контролисани полупроводници, који су постављени у парове, а ти парови формирају три паралелне гране.

Управљачко коло извршава четири веома битна поступка која су неопходна за правилан рад фреквентног регулатора. Дакле, управљачко коло управља полупроводницима фреквентног регулатора, размењује податке између претварача и периферних уређаја, сакупља информације о стању погона и остварује заштитне функције за фреквентни регулатор и мотор који покреће.

4. ПРОГРАМАБИЛНИ ЛОГИЧКИ КОНТРОЛЕР

Програмабилни логички контролер је микроконтролерски систем у коме су хардвер и софтвер специјално адаптирани индустријском окружењу. PLC је дефинисан као дигитални електронски уређај који користи програмабилну меморију за памћење наредби којима се захтева извођење специфичних функција, као што су логичке функције, секвенцирање, пребројавање, мерење времена и израчунавање, у циљу управљања различитим машинама и процесима.

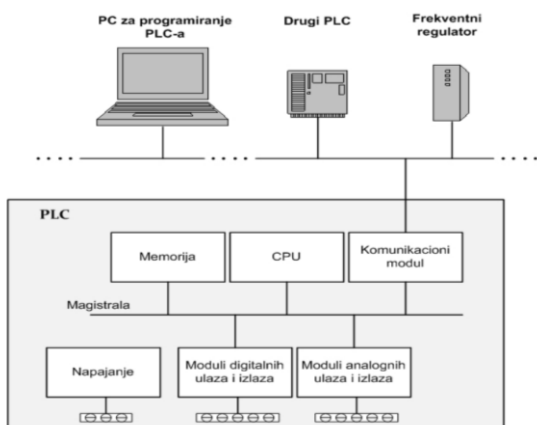
Програмабилни логички контролер као и сваки рачунар има оперативни систем, али је он знатно мањих могућности од оперативних система опште намене. Оперативни систем PLC-a подржава разне облике комуникационих протокола, а неки од њих су: RS485, PROFIBUS, MODBUS, TCP/IP, ... Због тога је могуће извести повезивање програмабилних логичких контролера и централног рачунара ради управљања индустријским процесима на даљину.

4.1. Основна конфигурација PLC контролера

На слици 2. налази се основна конфигурација програмабилног логичког контролера.

Главни елементи PLC контролера су: централна процесорска јединица (CPU), меморија за програм и податке, комуникациони део, напајање, улазни део (дигитални и аналогни), излазни део (дигитални и аналогни) и део за проширење.

Основа рада програмабилног логичког контролера заснива се на континуалном скенирању програма. Под скенирањем се подразумева пролаз кроз све услове у дефинисаном временском интервалу. Процес скенирања састоји се из пет независних фаза које заједно представљају скен циклус. Поменуте фазе скенирања су: провера статуса улаза (улазни скен), извршење програма (програmsки скен), провера и измена статуса излаза (излазни скен), опслуживање комуникационих портова и самотестирање.

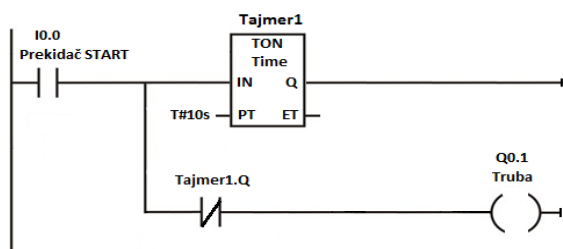


Слика 2. Основна конфигурација PLC-a

4.2. Лествичаста дијаграм PLC контролера

Лествичаста дијаграм састоји се од једне вертикалне линије која се налази на левој страни и низа линија које се гранају према десном делу. Линија са леве стране назива се bus bar, а линије које се гранају на десно називају се линије инструкција. Дуж линија инструкција смештени су услови који воде до инструкција позиционираних на десном крају дијаграма.

Када и на који начин се инструкција на десној страни извршава одређује логичка комбинација услова. Најважније групе елемената које се користе за формирање лествичастих дијаграма су: контакти, излази и блокови. Контакти могу бити нормално отворени или затворени, а излази репрезентују логички излаз, односно стање излазних уређаја док блокови репрезентују тајмере, бројаче, итд. На слици (3) која следи налази се пример једног лествичастог дијаграма.



Слика 3. Пример лествичастог дијаграма

5. РЕАЛИЗАЦИЈА ПРАКТИЧНОГ ДЕЛА РАДА

Пројектни задатак мастер рада састоји се из три целине, односно обрађена су три посебна електромоторна погона комплексног постројења топлане на природни гас. У овом тренутку важно је напоменути да ће у овом раду бити представљен само један од та три задатак, односно електромоторни погон вентилатора свежег ваздуха котла.

5.1. Електромоторни погон вентилатора свежег ваздуха котла

За асинхронни мотор вентилатора свежег ваздуха котла изабран је мотор произвођача „ABB“ чији су номинални подаци: снага 160 kW, напон 400 V, фреквенција 50 Hz, струја 282 A и брзина обртања 1488 o / min.

За напајање и регулацију брзине асинхронног мотора вентилатора свежег ваздуха вреловодног котла користи се фреквентни претварач произвођача „Danfoss“ из серије „VLT Aqua Drive“ конкретно модел FC 202 који је приказан на слици (4) испод.



Слика 4. Фреквентни регулатор Danfoss FC 202

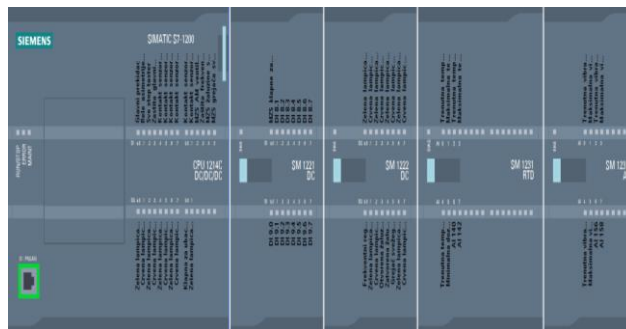
Подешавање параметара фреквентног регулатора за управљање асинхроним мотор вршимо помоћу Quick Menu опције у коју уносимо номиналне податке везане за мотор у следеће параметре:

- параметар 1-20 уписујемо 160 kW,
- параметар 1-22 уписујемо 400 V,
- параметар 1-23 уносимо 50 Hz,
- параметар 1-24 уносимо 282 A и
- параметар 1-25 уписујемо 1488 o/min.

Затим подешавамо параметар 5-12 који дефинише функцију дигиталног улаза 27. Након тога активирамо аутоматску адаптацију мотора (AMA) која се налази на параметру 1-29 која аутоматски подешава и оптимизује напредне параметре мотора.

5.2. Избор и програмирање PLC-a

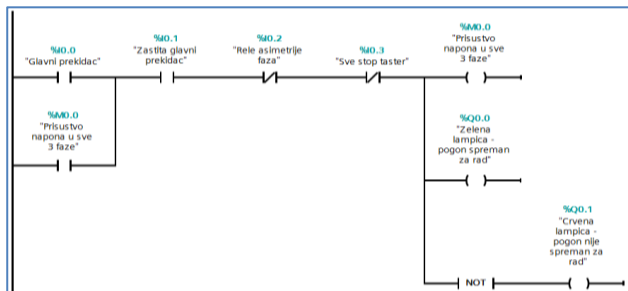
PLC који се користи у електромоторном погону вентилатора свежег ваздуха котла је компаније „Siemens“ SIMATIC S7 PLC, конкретно модел S7-1200. Централна процесорска јединица (CPU) има ознаку CPU 1214C DC/DC/DC (6ES7 241-1AG40-OXB0) и са својим дигиталним сигналним улазним и излазним модулима чини комплетну управљачку конфигурацију PLC-a за аутоматизацију електромоторног погона вентилатора свежег ваздуха котла. Конфигурација PLC-a за управљање поменутих погоних приказана је на слици (5) која следи у наставку поглавља.



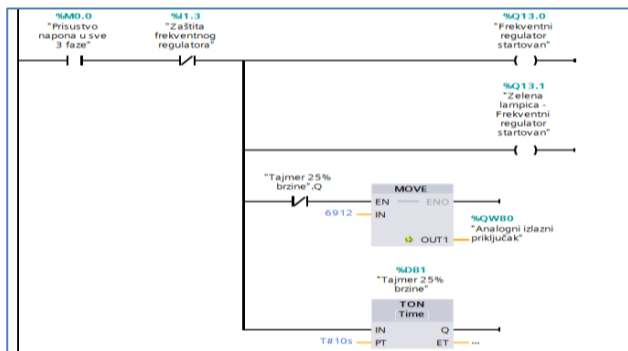
Слика 5. Конфигурација PLC-a за аутоматизацију електромоторног погона вентилатора свежег ваздуха котла

Након формирања PLC-а можемо прећи на формирање управљачког лествичастог дијаграма у програму TIA portal v13. Сада ћемо представити поједине лествичасте дијаграме управљања електро-моторним погоном вентилатора свежег ваздуха котла.

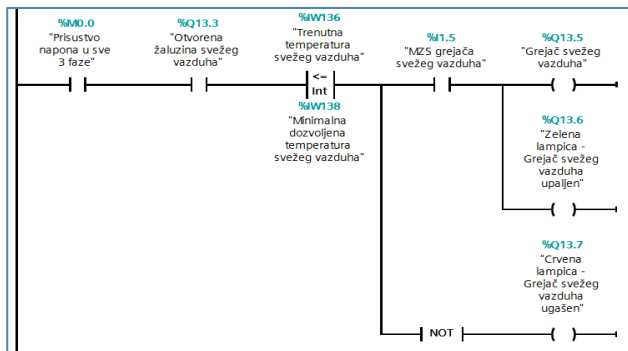
Network 1: Присуство напона у погону



Network 2: Стартовање фреквентног регулатора за управљање асинхроним мотором



Network 3: Загревање свежег ваздуха котла



5. ЗАКЉУЧАК

У раду су приказани поступци приликом комплетне обраде једног савременог електромоторног погона, конкретно електромоторног погона вентилатора свежег ваздуха котла.

Дакле, изабран је асинхронни мотор са одговарајућим перформансама, затим је одабран адекватан фреквентни регулатор и подешени су параметри регулатора који управља брзином обртања асинхронног мотора. На крају је изабран PLC контролер и формиран је лествичасти дијаграм за комплетну аутоматизацију погона.

Написани PLC програм испуњава све потребне услове који су неопходни за безбедан, ефикасан и економичан рад приликом процеса сагоревања природног гаса у вреловодном котлу. Програмабилни логички контролери налазе све већу примену у најразличитијим областима и делатностима, као што су управљање електроенергетским системима у разним индустријским процесима.

Циљ овог рад јесте да се представе PLC системи управљања и макар делимично добије увид у квалитет и предности система који се контролишу програмабилним логичким контролерима.

5. ЛИТЕРАТУРА

[1] Дарко Марчетић, Марко Гецић, Борис Марчетић, „Програмабилни логички контролери и комуникациони протоколи у електроенергетици“, Факултет техничких наука, Нови Сад, 2016.

[2] Влодо Поробић, „Програмабилни логички контролери и комуникациони протоколи у електроенергетици – примери са решењима“, Факултет техничких наука, Нови Сад, 2017.

[3] <https://gselektro.rs/>

[4] <https://files.danfoss.com/download/Drives/MG20MD02.pdf>

[5] Веран Васић, Ђура Орос, „Енергетска електроника у погону и индустрији“, Факултет техничких наука, Нови Сад, 2012.

[6] <https://www.siemens.com/rs/sr/home.html>

[7] Hans Berger, „Automating with SIMATIC S7-1200: Configuring, Programming and Testing with STEP 7 Basic; Visualization with WinCC Basic“, Publicis Publishing, Erlangen, 2013.

[8] Србијанка Турајлић, „Програмабилни логички контролери“, Електротехнички факултет, Београд, 2011.

Кратка биографија:



Илија Гелић рођен је у Новом Саду, 1993. године. Факултет техничких наука у Новом Саду уписао је 2015. године, а дипломирао је 2019. године. Мастер рад на истом факултету из области Електротехнике и рачунарства - Енергетска електроника и електричне машине одбранио је 2021. године.



Дејан Јеркан је доцент на Факултету техничких наука у Новом Саду, на Катедри за Енергетску електронику и претвараче. Област интересовања су му моделовање и дијагностика електричних машина, као и метода коначних елемената.

RAZVOJ 5G MREŽE, STAVOVI I OČEKIVANJA**5G NETWORK DEVELOPMENT, ATTITUDES AND EXPECTATIONS**Marinela Mikša, Željien Trpovski, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – Može se reći da su mobilni komunikacioni sistemi izvršili revoluciju u načinu komunikacije ljudi. 5G predstavlja naredni korak u evoluciji mobilne tehnologije. Za razliku od prethodnih generacija, ova generacija je orijentisana ka industriji i velikim korisnicima i nudi nove okvire i mogućnosti. Međutim, ona će unaprediti i svakodnevni život ljudi. Glavna prednost 5G mreže ogleda se u malom kašnjenju i većim brzinama prenosa.

Ključne reči: 5G mreža, IoT

Abstract – One could say that the mobile communications system have revolutionized the way people communicate. 5G is the next step in the evolution of mobile technologies. Unlike previous generations, this one is oriented towards industry and major users and provides new frameworks and possibilities. Although, it will also make everyday life more comfortable. The main advantages of 5G network are minimal delay times and high data rates.

Keywords: 5G mobile network, IoT

1. UVOD

Današnje vreme, vreme savremenog i modernog društva nemoguće je zamisliti bez korišćenja informaciono-komunikacionih tehnologija.

Tokom vremena pokazalo se da razne tehnologije uvode nove mogućnosti olakšavajući svakodnevne aktivnosti ljudi. Konstantno povećanje broja korisnika nametalo je potrebu za ubrzanim tehnološkim razvojem. 5G mreža predstavlja naredni korak u razvoju mobilne tehnologije.

Cilj ovog rada jeste da ukaže na prednosti i mogućnosti koje 5G mreža nudi i da pokaže kakvi su stavovi građana kada je u pitanju 5G mreža.

Nakon uvoda, u drugom poglavlju je prikazan razvoj bežičnih komunikacija. Prikazane su mreže po generacijama uz kratak opis svake generacije.

Treće poglavlje odnosi se na 5G mrežu, opisani su ciljevi i zahtevi 5G mreže, definisan je spektar i standardizacija 5G mreže.

U okviru četvrtog poglavlja opisan je Internet of Things (IoT). Definisana je oblast primene IoT, tehnologija rada IoT i područje primene IoT.

Peto poglavlje opisuje važnost optičke mreže za 5G.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Željien Trpovski, vanr.prof.

Šesto poglavlje odnosi se na razmatranje rezultata ankete koje je urađena za potrebe ovog rada. Tema istraživanja jeste Istraživanje stavova građana o 5G mobilnoj mreži.

Poslednje poglavlje odnosi se na zaključna razmatranja koja su donešena na osnovu proučene materije.

2. RAZVOJ BEŽIČNIH KOMUNIKACIJA

Evolucija tehnologija mobilnih mreža posmatra se kroz sledeće generacije [1]:

- 1G predstavlja prvu generaciju ćelijskih sistema koji su prenosili signale analognim putem, bez kodovanja, i služile su za prenos glasa.
- 2G označava drugu generaciju i ona donosi veliki broj inovacija, a kao najvažnije izdvajaju se: prenos informacija digitalnim putem, poboljšanje kvaliteta prenosa podataka, povećanje kapaciteta i pokrivenost. Prenos govora i dalje predstavlja dominantan servis, dok su novine koje ova generacija nosi: prenos kratkih poruka (SMS), slanje faksa i prenos podataka.
- 3G podrazumeva mrežu treće generacije. Treća generacija sa sobom nosi brži prenos podataka. Pojavljuju se IP adrese za mobilne telefone, pa dolazi do masovne proizvodnje "pametnih" telefona.
- 4G se odnosi na četvrtu generaciju. U ovoj generaciji akcenat je na protoku informacija, a ne kao do tada na mobilnim sistemima. Ono što je karakteristično za 4G: svaki mobilni telefon dobija svoju IP adresu, razmena multimedijalnih sadržaja bez ikakvih problema, mobilni uređaji preuzimaju funkcije prenosivih računara.
- 5G označava mrežu pete generacije i ono što se odnosi na njene aspekte je veća brzina prenosa, smanjena latencija koja podrazumeva brzinu prenosa sa jednog uređaja na drugi i sposobnost povezivanja mnogo više uređaja i opreme.

3. PETA GENERACIJA- 5G

Sa povećanjem potrebe za bežičnim povezivanjem sve većeg broja korisnika razvio se i sistem mobilne telefonije pete generacije (5G). Peta generacija predstavlja važan korak u mogućnostima koje nudi mobilna mreža.

Za razliku od dosadašnjih mogućnosti povezivanja pametnih telefona, tableta i prenosivih računara 5G će omogućiti nove usluge, uključujući i industrijski IoT [2]. 5G mreža se bazira na pet tehnologija [3]:

- milimetarskim talasima,
- malim baznim stanicama,
- masivnim MIMO sistemima,
- usmeravanju signalnih snopova i
- full duplex tehnologiji.

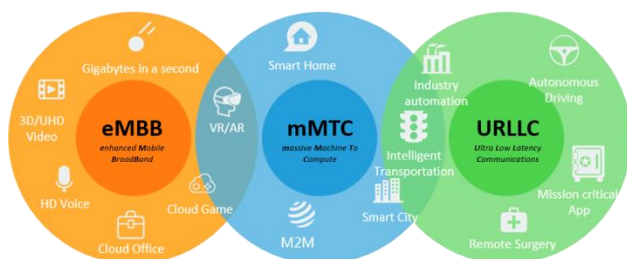
3.1. Ciljevi razvoja 5G mreže

Ciljevi ove generacije postavljeni su vrlo visoko i baziraju se na [2]:

- brzinama prenosa podataka do 20 Gb/s,
- povezivanje velikog broja uređaja,
- ultra niskom kašnjenju i
- velikom pouzdanošću.

U odnosu na prethodne generacije koje su se uglavnom zasnivale na povezivanju ljudi i omogućavale su pristup internetu sa bilo kog mesta, 5G se odnosi na povezivanje svega oko nas. Sam dizajn 5G mreže koji karakteriše pouzdana veza sa malim kašnjenjem može imati veliki uticaj na industriju, saobraćaj, medicinu, poslovanje preduzeća, društvo i mnoge druge oblasti. Druga važna karakteristika jeste velika količina podataka koja se prenosi veoma velikom brzinom.

Kada se govori o 5G mreži, razlikuju se tri slučaja upotrebe (Slika 1.) [3, 4]:



Slika 1. 5G servisi

- eMBB (Enhanced Mobile Broadband)- podrazumeva klasičan razvoj mobilne telefonije sa korisničkog aspekta (veći obim podataka, podržavanje još većeg broja korisnika, povećanje brzine prenosa podataka). Može se reći da je eMBB zapravo unapređenje 4G sa akcentom na mobilni internet.
- mMTC (massive Machine Type Communications)- odnosi se na povezivanje velikog broja uređaja na internetu putem 5G arhitekture. mMTC nalazi primenu kod pametnih kuća, zgrada, gradskih infrastruktura, električne mreže i drugo.
- URLLC (Ultra Reliable Low Latency Communications) – ovaj deo se fokusira na servise za koje je neophodno jako malo kašnjenje pri prenosu podataka i izuzetno visoku pouzdanost. Primena ovih servisa bila bi prvenstveno u industriji, saobraćaju, medicini (npr. bezbednost saobraćaja, autonomna vožnja, upravljanje uređajima na daljinu, hirurgija na daljinu i slično).

3.2. Zahtevi 5G mreže

Za svaki od ova tri slučaja postavljeni su i zahtevi koje mreža mora da ostvari:

- eMBB zahteva brzinu prenosa > 10 Gb/s,
- mMTC > 1M/km²,
- URLLC zahteva kašnjenje < 1ms.

Svetsko standardizaciono telo 3GPP pokrenulo je razvoj nove tehnologije radio pristupa koja se još naziva i 5G NR. Tehnički radovi započeli su u proleće 2016. godine, dok je prva verzija NR tehnologije bila dostupna krajem 2017. godine. Ovaj standard odnosio se samo na određene aspekte 5G.

Prethodni koraci ispunili su sve komercijalne zahteve koji su doveli do primene 5G mreže već u 2018. godini. 5G NR tehnologija i dalje koristi mnoge strukture i

karakteristike LTE-a, međutim ona ima drugačiji skup tehničkih rešenja [3].

Kada se govori o osnovnim zahtevima koje će 5G mreža morati da ispuni, govori se pre svega o podršci za višestruke usluge (multi- service) i višestruko korišćenje pokretne mreže (mobile network multi- tenancy) uz smanjenje kapitalnih i operativnih troškova i najbolje iskorišćenje resursa [6].

3.3. Spektar i standardizacija

Prema definiciji Međunarodne unije za telekomunikacije (ITU) upravljanje radio- frekvenčijskim spektrom predstavlja kombinaciju upravnih, naučnih i tehničkih postupaka potrebnih da se obezbedi delotvoran rad radio- komunikacijskih uređaja i službi bez prouzrokovanja štetnih smetnji.

Radio spektar je veoma važan resurs i predstavlja osnovu za bežičnu komunikaciju. Spektar je konačan resurs pa je dodela istog mora biti na najbolji i najfleksibilniji način.

Prva i druga generacija bile su na frekvencijama od oko 800- 900 MHz. Sa razvojem 3G i 4G mreže i potrebom da se prošire servisi novi opsezi su bili dodati i nižim i višim frekvencijama pa je tako novi opseg bio u rasponu od 450 MHz do 6 GHz.

Na različitim frekvencijama opsezi mogu imati različite karakteristike. Opsezi sa nižim frekvencijama dobri su za pokrivanje širokog područja poput prigradskih i ruralnih područja, razlog tome je propagacija.

Uvođenjem 5G mreže i zahteva koje ona nosi u smislu usluga i servisa koje zahtevaju velike brzine prenosa podataka, veliki kapacitet i malo kašnjenje javlja se i potreba za novim opsezima. Prva primena 5G koristiće opsege koje je koristila i prethodna generacija, dok se opsezi iznad 24 GHz razmatraju kao frekvenčijski opsezi za dopunu frekvenčijskim opsezima ispod 6 GHz. Nove opsege kontinualno definiše 3GPP, a razmatraju se frekvencije i iznad 60 GHz [5].

Prednost 5G mreže jeste ta što je dizajnirana fleksibilno, odnosno postoji mogućnost korišćenja svih raspoloživih opcija spektra, uključujući i licencirani i nelicencirani. Kako se može koristiti i u javnim i u privatnim mrežama otvaraju se novi potencijali primene u industriji [2,5].

4. INTERNET OF THINGS- IoT

Prva zvanična definicija IoT od strane ITU nastaje 2005. godine i tada su je definisali kao „globalnu infrastrukturu informacionog društva, koja međusobnim povezivanjem stvari (fizičkih i virtualnih) omogućava napredne usluge, a bazira se na postojećoj i rastućoj interoperabilnoj informaciono- komunikacionoj tehnologiji“.

Ova osnovna definicija se 2012. godine proširila sledećim dodatkom [5]:

- „na bazi identifikacije, prikupljanja, obrade i razmene podataka, IoT koristi stvari kako bi ponudio usluge za sve vrste aplikacija, osiguravajući zahteve za sigurnost i privatnost“ i
- „u širem smislu, IoT je vizija s tehnološkim i društvenim posledicama.“

Internet of Things (IoT) ili na srpskom jeziku Internet stvari je pojam koji se odnosi na umrežavanje, odnosno povezivanje uređaja ili objekata koji komuniciraju sa ljudima ili drugim fizičkim objektima i sistemima putem

internet, pri čemu se komunikacija može izvesti bez ljudske interakcije ili nadzora [6].

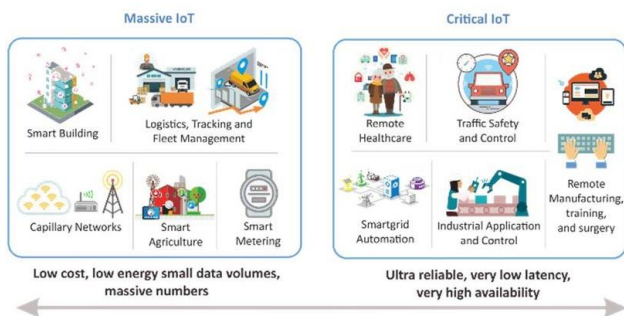
Primeri povezivanja objekata su veoma različiti, pa tako stvar povezivanja može biti bilo šta: zgrada opremljena senzorima za temperaturu koji omogućavaju praćenje i korigovanje potrošnje energije u realnom vremenu, kućni ljubimac sa čipom koji sadrži sve informacije o vlasniku, zdravstvenom stanju, vakcinama, ambalaža prehrambenih proizvoda koja sadrži bar kod sa informacijama o datumu proizvodnje koje frižider koji ima odgovarajuću opremu može da pročitati i obavesti korisnika o informacijama vezanim za istek roka i slično [5].

Prema tome, predmet povezivanja može se definisati kao fizički ili virtuelni objekat koji se zbog mogućnosti povezivanja, odnosno umrežavanja naziva i pametnim objektom (smart object).

4.1. Oblast primene

IoT je pokretačka snaga tehnologija na kojima se zasnivaju mMTC i URLLC, slučajevi primene ovih tehnologija mogu se podeliti u dve grupe a to su (Slika 2.) [6]:

- prva grupa (massive IoT) ima za cilj da poveže veliki broj uređaja. Ova grupa obuhvata aplikacije koje vrše razmenu malih i retkih paketa podataka od/do drugih samostalnih uređaja, iz tog razloga ovi uređaji moraju biti ekonomski pristupačni i sa malom potrošnjom energije, pri čemu nema strogih zahteva za kašnjenje. Glavni zahtevi ovih uređaja su niska cena, mala potrošnja energije, dobra pokivenost i skalabilnost.
- druga grupa (mission-critical IoT) sastoji se od osetljivih aplikacija, sa izuzetnim zahtevima za kašnjenje (reda veličine dela milisekunde). Zahtevi koji se odnose na ovu grupu su visoka dostupnost i pouzdanost i veoma nisku latentnost.



Slika 2. Oblast primene

4.2. Tehnologija rada IoT

Princip rada IoT zasniva se na ugrađenim senzorima koji prikupljaju informacije iz okoline i prosleđuju ih putem komunikacijske mreže.

Senzorski uređaji sa velikom efikasnošću i brzinom i niskom potrošnjom energije mogu se postavljati na različita mesta i mogu komunicirati bez posredstva čoveka po principu M2M (Machine to machine communication).

Okolina IoT je pametna, ona zahteva da se poveže veliki broj uređaja i da se generišu velike količine podataka. Kako bi se izbegli troškovi sopstvene infrastrukture razvijeno je računarstvo u oblaku (Cloud computing), koje se zasniva na deljenju servera, odnosno zajedničkom deljenju prostora. Koncept računarstva u oblaku

omogućava da se velike količine podataka (Big data) generišu i analiziraju bez velikih troškova.

Okolina IoT može se opisati na sledeći način: Svaki IoT uređaj (pametni uređaj) u sebi ima ugrađene senzore koji prikupljaju podatke i neke komunikacione module koji će prikupljene podatke proslediti kroz mrežu na računarski oblak gde se oni generišu i dalje obrađuju.

4.3. Područje primene IoT

Spektar usluga koje pruža IoT veoma je širok, neke od njih se mogu grupisati prema područjima primene [5,7]:

- Pametna kuća (Smart home)- koncept pametne kuće omogućava korisnicima da uz upotrebu različitih senzora koji su raspoređeni po kući upravljaju prostorom čineći im jednostavniji sam boravak u kući. Suština je da uređaji prikupljaju podatke putem senzor, komuniciraju razmenom prikupljenih podataka i usklađuju svoje funkcionisanje sa željama korisnika.
- Pametni grad (Smart city)- Ima za cilj da svojim uslugama poveća kvalitet života građana korišćenjem informaciono- komunikacionih tehnologija. Suština pametnog grada je u digitalizaciji, odnosno u upravljanju kompletnom gradskom infrastrukturom uz pomoć informaciono- komunikacionih tehnologija.
- Pametna okolina (Smart environment) ima za cilj informisanje korisnika o stanju okoline, klimatskim uslovima, zagađenju i prirodnim katastrofama. Ova usluga može biti zanimljiva građanima ali je isto tako veoma važna za stručnjake.
- Pametno zdravstvo (Smart healthcare) je jedna od najzanimljivijih i najvažnijih područja primene kako je zdravlje ljudi prioritet. Pametno zdravstvo nudi mnogo novih mogućnosti poput aplikacija za daljinski nadzor i praćenja stanja pacijenata, praćenje hroničnih bolesti, udaljeni monitoring pacijenata u realnom vremenu. IoT uz pomoć 5G tehnologije unaprediće zdravstvo i stvorice pametno upravljanje čitavim zdravstvenim sistemom.
- Pametna poljoprivreda (Smart agriculture) podrazumeva uvođenje pametnih sistema sa ciljem optimizacije poljoprivrednih procesa. Primenom IoT tehnologije putem brojnih senzora prikupljaju se potrebne informacije koje su od velikog značaja poljoprivrednicima dajući im trenutni uvid u stanje zemlje, poljoprivrednih kultura, mehanizacije i slično.

5. OPTIČKA MREŽA I 5G TEHNOLOGIJA

Optičko vlakno konstantno napreduje u pogledu kapaciteta prenosa, 1989. godine je kapaciteta prenosa sa jednim vlaknom iznosio 2.5 Gb/s dok se 2019. godine beleži povećanje na čak 32 Tb/s [8].

Implementacija četvrte generacije sistema za mobilne komunikacije, popularizacija pametnih telefona i tableta dovela je do pružanja usluga kao što su video pozivi i streaming medija. Daljim razvojem dolazi do pružanja niza usluga i servisa koje su zasnovane na oblaku (cloud tehnologija) a koje isto tako postavljaju svoje zahteve u pogledu brzine prenosa podataka.

Danas je pitanje zahteva koje postavlja 5G mreža a koje se zasniva na brzini komunikaciji sa malim kašnjenjem i povezivanjem širokog spektra uređaja .

Da bi se udovoljili zahtevi različitih 5G usluga, neophodno je koristiti izuzetno fleksibilne optičke mrežne uređaje i efikasno upravljati mrežom. Potrebne su

tehnologije za nadgledanje stanja mreže i za automatsko menjanje konfiguracija uređaja koji čine mrežu, u skladu sa njenim trenutnim stanjem [9].

6. ISTRAŽIVANJE STAVOVA GRAĐANA O 5G MOBILNOJ MREŽI

Za potrebe rada sprovedeno je istraživanje koje ima za cilj da utvrdi kakvi su zapravo stavovi ispitanika kada je u pitanju 5G mobilna mreža.

Za prikupljanje podataka korišćena je tehnika anketiranja, pri čemu je primenjen upitnik zatvorenog tipa. Istraživanje je sprovedeno u periodu februara meseca 2021. godine putem online upitnika. Anketom je obuhvaćeno 500 ispitanika.

Od ukupnog broja ispitanika 64% su činile osobe ženskog pola (322 ispitanika) dok su 36% činile osobe muškog pola.

U obzir su uzeti samo podaci ispitanika koji su čuli za 5G mrežu (493 ispitanika).

Zaključak istraživanja

Korisnici su svesni da je današnje društvo veoma vezano za savremene tehnologije, većina je bar delimično upoznata sa načinom funkcionisanja 5G mreže, međutim njihovi izvori informacija spadaju u manje pouzdane kanale informisanja, a kako se najveći broj ispitanika informiše putem društvenih mreža one imaju veliki uticaj na sam stav ispitanika. Izvori informacija su veoma važni ukoliko se govori o pouzdanim, tačnim i pravim informacijama.

Kako više od 50% ispitanika nije upoznato sa prednostima 5G mreže, potrebno je na neki pristupačan način dostaviti prave informacije korisnicima, mediji su jedan od načina da se to uradi na pravi način.

Kada se govori o stavovima ispitanika u pogledu uticaja na zdravlje čoveka i uticaja na životnu sredinu ispitanici su većinom odgovorili sa 3 (oko 40% ispitanika i u jednom i u drugom slučaju). Ovim odgovorom ispitanici potvrđuju da nisu baš sigurni na koji način funkcioniše ova mreža. Oko 25% ispitanika (kod oba pitanja) odgovorilo sa 5, tj. da ona veoma utiče.

Neophodno je da se stručna lica uključe u informisanje javnosti, kako bi se zabrinutost oko ovog pitanja smanjila.

7. ZAKLJUČAK

Informaciono društvo u kome danas živimo gotovo je nezamislivo bez prisustva računara. Ono što se sa sigurnošću može primetiti jeste da čovek ima sve manje slobodnog vremena, pa se korišćenje savremenih tehnologija nameće kao logično rešenje za zadovoljenje svakodnevnih potreba čoveka u cilju organizacije vremena.

Na osnovu izvršenog istraživanja zaključuje se da je stanovništvo voljno da prihvati nove tehnologije, međutim, informacije koje imaju dolaze sa neadekvatnih izvora informacija pa tako sklapaju neku pogrešnu sliku o ovoj tehnologiji.

Smatra se da će 5G mreža u narednih 10 godina dramatično promeniti način funkcionisanja skoro svega što nas okružuje.

IoT može značajno unaprediti razne segmente i nalazi primenu gotovo svuda, od domaćinstva pa do najvećih industrijskih postrojenja.

Potencijal 5G mreže veoma je velik, otvaraju se nova vrata, a mogućnosti za inovacijama su raznovrsne. 5G mreža će izvršiti transformaciju industrije i procesa poslovanja.

8. LITERATURA

- [1] <https://raf.edu.rs/citaliste/racunarske-mreze/2067-bezicne-mobilne-celijske-komunikacije-uz-osvrtnom-na-istorijat-mobilne-telefonije-2>
- [2] Harri Holma, Antti Toskala, Takehiro Nakamura, 5G Technology, JohnWiley & Sons Ltd. 2020
- [3] Erik Dahlman, Stefan Parkvall, Johan Skold, 5G NR: The Next Generation Wireless Access Technology, Elsevier Ltd, 2018.
- [4] <https://iconic.ftn.uns.ac.rs/category/blog/>
- [5] Tatjana Cvetković, Vladica Tintor, Regulativa elektronskih komunikacija, Visoka škola elektrotehnike i računarstva strukovnih studija, Beograd 2019.
- [6] Sassan Ahmadi, 5G NR, Elsevier Inc. 2019.
- [7] Ivan Livaja, Zvonimir Klarin, Utjecaj 5G mreže na Internet stvari, Stručni rad, Veleučilište u Šibeniku
- [8] <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6920305/>
- [9] Shoichiro Oda, Taku Saito, Kazuto Nishimura, Akinori Hayakawa, Optical Network Technologies for Enabling 5G Services, FUJITSU Sci. (2019)

Kratka biografija:



Marinela Mikša rođena je 17.12.1991. godine u Vršcu. Osnovnu školu završila je u Margiti, a zatim 2010. godine završava Gimnaziju „Borislav Petrov Braca“ u Vršcu. Završila je osnovne i master akademske studije na Departmanu za saobraćaj. 2019. godine upisuje master akademske studije na Departmanu za energetiku, elektroniku i telekomunikacije.



Željko Trpovski rođen je u Rijeci 1957. god. Doktorirao je na Fakultetu tehničkih nauka 1998. god. Oblast interesovanja su telekomunikacije i obrada signala.

Zahvalnica:

Izradu ovog rada pomogao je Fakultet tehničkih nauka u Novom Sadu, Departman za energetiku elektroniku i telekomunikacije, u okviru projekta pod nazivom: "Istraživanja u oblasti energetike, elektronike, telekomunikacija i primenjenih informacionih sistema u cilju modernizacije studijskih programa".

DETEKCIJA OBJEKATA U SAOBRAĆAJU UPOREBOM KONVOLUCIONIH NEURONSKIH MREŽA**OBJECT DETECTION IN TRAFFIC SCENES WITH CONVOLUTIONAL NEURAL NETWORKS**Sofija Pantović, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – Detekcija objekata je ključna tehnologija koja stoji iza naprednih sistema za asistenciju tokom vožnje. U ovom radu je prikazana primena modela konvolucionih neuronskih mreža u rešavanju problema detekcije objekata od interesa iz sekvence slika. Obeležene slike koje sadrže različite scenarije zabeležene tokom dnevne gradske vožnje preuzete su iz CrowdAI baze [1]. Slike su korišćene za učenje modela i ocenu njegovih performansi tokom faze testiranja. Analizirani su rezultati dobijeni korišćenjem detektora sa različitim brojem konvolucionih slojeva i različitim aktivacionim funkcijama neurona u cilju primene ovakvih modela za detekciju objekata učesnika u saobraćaju u realnom vremenu.

Ključne reči: Detekcija objekata, Klasifikacija, Konvoluciona neuronska mreža, Okvir

Abstract – Object detection is the key technology behind advanced driver assistance systems. This paper demonstrates an application of convolutional neural networks in the task of detecting objects of interest from the sequence of images. Labeled images with various driving scenarios recorded during daylight city drive are taken from CrowdAI database [1]. Images were used for model training and evaluation during testing phase. Results obtained using models with different number of convolutional layers and various activation functions are analyzed with purpose of using these models for real-time object detection.

Keywords: Bounding box, Classification, Convolutional neural network, Object detection

1. UVOD

Detekcija objekata iz sekvence slika je ključna tehnologija koja stoji iza naprednih sistema za asistenciju tokom vožnje (engl. *advanced driver assistance systems*). Sistemi za asistenciju tokom vožnje detektuju vozne trake, ivice puta, druga vozila ili pešake, sa ciljem poboljšanja sigurnosti putnika i drugih učesnika u saobraćaju.

Klasifikacija objekata je proces predviđanja klase objekta sa slike. Lokalizacija objekata odnosi se na identifikaciju lokacije jednog ili više objekata u slici i određivanje njihovog okvira (engl. *bounding box*).

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je dr Vladimir Bugarski, docent.

Kombinacijom ovih zadataka nastaje prepoznavanje tj. detekcija objekata iz slike [2]. Konvolucione neuronske mreže (engl. *convolutional neural networks*) su se pokazale kao uspešne kod prepoznavanja objekata u slici, pa se nameću kao logičan izbor za rešenje problema detekcije [3].

CrowdAI baza podataka od preko 65.000 obeležja iz 9.423 kadra visoke rezolucije korišćena je za obuku neuronske mreže. Skup podataka je obeležen kombinacijom ručnog unosa i tehnika mašinskog učenja. Objekti od interesa pripadaju sledećim klasama:

1. Automobil;
2. Kamion;
3. Pešak.

Pored oznake klase, obeležje sadrži koordinate okvira detektovanog objekta u formatu $[x_{min}, y_{min}, x_{max}, y_{max}]$.

2. KONVOLUCIONA NEURONSKA MREŽA

Konvoluciona neuronska mreža je algoritam dubokog učenja koji kao ulaz, između ostalog, prima sliku, dodeljuje značaj (naučene težine i pristrasnost (engl. *bias*)) različitim objektima u slici, kako bi bio u mogućnosti da ih razlikuje [4]. Konvolucione neuronske mreže se, kao i regularne višeslojne neuronske mreže, sastoje od jednog ulaznog, jednog izlaznog i barem jednog ili više skrivenih slojeva. Ono što razlikuje konvolucionu neuronsku mrežu od obične jesu konvolucionni slojevi i slojevi sažimanja.

2.1 Konvolucionni sloj

Konvolucija dve funkcije $f, g: \mathbb{R}^d \rightarrow \mathbb{R}$ definisana je kao:

$$(f * g)(x) = \int f(z)g(x - z)dz \quad (1)$$

U slučaju diskretnog skupa i dvodimenzionih tenzora, integral se pretvara u sumu:

$$(f * g)(i, j) = \sum_a \sum_b f(a, b)g(i - a, j - b) \quad (2)$$

gde a i b označavaju granice konvolucionog kernela u odnosu na centar.

Tokom faze propagiranja signala unapred, ulazna slika se konvoluiira sa kernelom tj. filterom, što jasno ukazuje da je zadatak obuke konvolucione mreže podešavanje težina u okviru različitih kernela. Proces konvolucije omogućava redukciju ulaznih slika u formu koja je lakša za obradu, bez gubitaka obeležja koja su esencijalna za postizanje dobre predikcije [4]. Uprošćena reprezentacija ulaznih slika se naziva aktivaciona mapa ili mapa obeležja.

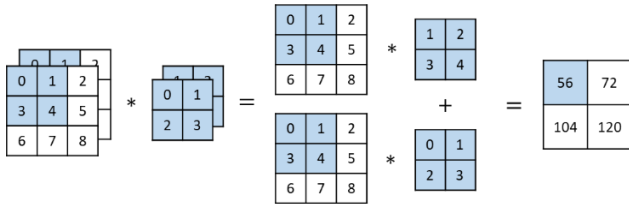
Za slučaj višestrukih kanala ulaza X i skrivene mape obeležja H važi relacija:

$$[H]_{i,j,d} = \sum_{a=-\Delta}^{\Delta} \sum_{b=-\Delta}^{\Delta} [V]_{a,b,c,d} [X]_{i+a,j+b,c} \quad (3)$$

gde je $[V]_{a,b,c,d}$ tenzor težinskih faktora u okviru kernela, $[X]_{i+a,j+b,c}$ deo ulazne slike obuhvaćene kernelom, dok d ukazuje na broj izlaznih kanala skrivenih mapa obeležja [4].

2.2 Primena konvolucije na slikama

Kada ulazne slike sadrže višestruke kanale potrebno je konstruisati konvolucionni kernel sa istim brojem ulaznih kanala. Skrivena mapa obeležja se formira sabiranjem rezultata konvolucije dvodimenzionih tenzora ulaza i tenzora kernela za svaki od kanala (slika 1).



Slika 1. Konvolucija višestrukih slojeva – primer sa dva ulazna kanala i jednim izlaznim

2.3 Sloj sažimanja

Tokom propagacije signala unapred kroz mrežu, prostorna rezolucija se postepeno smanjuje, informacije se združuju tako da što je signal dublje u mreži, veće je receptivno polje ulaza na koje je skriveni sloj osetljiv [5]. Operacija sažimanja se primenjuje identično kao i konvolucionni kernel, sa razlikom što su operatori sažimanja deterministički tj. ne sadrže parametre. Najčešće se implementiraju tako da računaju maksimalnu ili prosečnu vrednost elemenata zahvaćenih prozorom. Sloj sažimanja obrađuje ulazne kanale pojedinačno, te će za svaki ulazni kanal ovaj sloj na svom izlazu izbaciti njemu odgovarajuću izlaznu aktivacionu mapu.

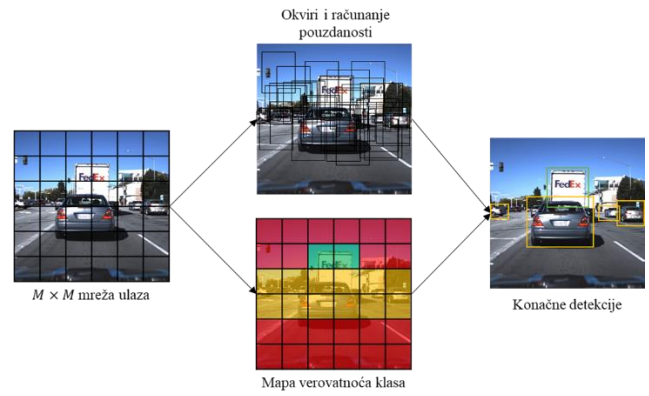
2.4 Arhitektura mreže

Tokom jednog prolaska kroz mrežu, model YOLO [4] arhitekture prikuplja značajna obeležja iz ulaznih slika i kasnije ih koristi za estimaciju okvira i klasa objekata u njima. YOLO model paralelno, za veći broj objekata u slici, estimira višestruke okvire i klasne verovatnoće za svakog od njih [6].

U poređenju sa sistemima slične namene, YOLO arhitektura, zbog jednostavnosti konfiguracije (ulazna slika kroz mrežu prolazi samo jednom), ima dobre performanse u realnom vremenu, čak i bez korišćenja naprednijih grafičkih komponenti [6].

Princip rada YOLO modela prikazan je na slici 2. Ulazna slika deli se na $M \times M$ jednakih ćelija. Unapred određen broj početnih okvira N koristi se za estimaciju objekata u svakoj od ćelija.

Pouzdanosti estimacije objekata u ćelijama formiraju mapu verovatnoća, koja govori da li se unutar okvira nalazi objekat ili ne, tj. koliko je model zapravo siguran u određenu predikciju. Na kraju se iz mapa verovatnoća klasa izdvajaju konačne detekcije objekata.



Slika 2. Princip rada YOLO modela.

2.5 Definicija okvira i računanje pouzdanosti

Jedan okvir definiše pet parametara: t_x, t_y, t_w, t_h i rezultat pouzdanosti. Koordinate piksela koji određuje gornji levi ugao okvira su date uređenim parom parametara (t_x, t_y) . Parametri t_w i t_h diktiraju visinu i širinu okvira, respektivno. Peti parametar, rezultat pouzdanosti, sadrži informacije o verovatnoćama za svaku od klasa u modelu. Zadatak konvolucione mreže za detekciju objekata jeste estimacija ovih pet parametara. YOLO estimira tenzor veličine $M * M * (N * 5 + C)$ za svaku sliku skupa za obuku, gde je $M * M$ broj ćelija, $N * 5$ broj apriornih okvira po ćeliji sa pet parametara za svaki okvir, dok je C broj klasa.

YOLO model vrši predikciju objekata u završnim slojevima mreže na osnovu klasifikacionih i lokalizacionih grešaka okvira, tj. grešaka između tačne i estimirane pozicije. Rezultat pouzdanosti p određene klase C jednak je umnošku verovatnoće postojanja objekta te klase unutar okvira i odnosa preseka i unije (engl. *intersection over union* - IOU) između apriornog i stvarnog okvira:

$$p(C) = p(\text{objekta}) \cdot IOU \quad (4)$$

Detektovani okvir koji se savršeno poklapa sa originalnim imaće izlaz 1, dok će svako odstupanje u preklapanju proizvesti manji IOU rezultat.

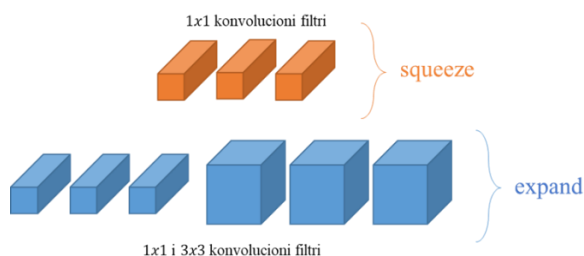
3. SQUEEZENET MODEL

Arhitektura neuronske mreže SqueezeNet koja čini unapređenu verziju YOLO modela (YOLOv3) [6], izabrana je za obuku nad CrowdAI bazom podataka. SqueezeNet mrežu karakteriše arhitektura sa minimalnim brojem parametara uz održavanje zadovoljavajućeg nivoa tačnosti.

U [6] i [7] su navedene strategije koje su korišćene kako bi se postiglo smanjenje parametara modela:

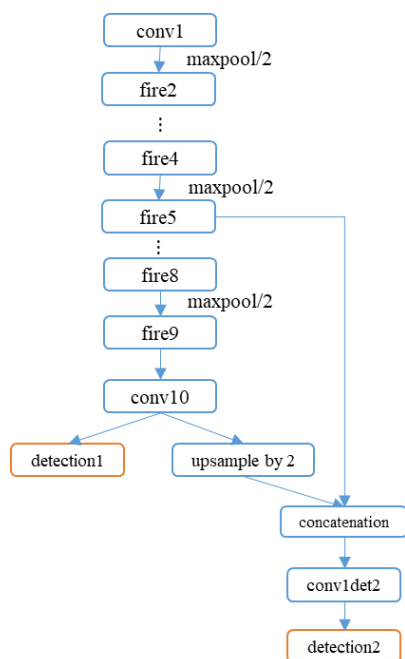
1. Umesto filtera veličine 3×3 , u konvolucionim slojevima su korišćeni filteri veličine 1×1 , čime se broj parametara smanjuje devet puta.
2. Upotrebom slojeva za sažimanje (engl. *squeeze*) je redukovano broj ulaznih kanala u filtere veličine 3×3 , što značajno smanjuje broj potrebnih parametara.
3. Smanjenje dimenzionalnosti u kasnijim nivoima mreže realizovano je postavljanjem koraka na vrednost veću od 1 u konvolucionim i slojevima sažimanja.

Predložene strategije smanjenja parametara su realizovane formiranjem okidačkog modula (engl. *fire modul*) (slika 3). Ovaj modul se sastoji od konvolucionog sloja sa 1x1 filtrima koji predstavlja sažimajući sloj, čiji se izlazi prosleđuju u sloj koji je kombinacija 1x1 i 3x3 konvolucionih filtara, takozvani proširujući sloj (engl. *expand*).



Slika 3. Organizacija konvolucionih filtara u okidačkom modulu.

Model SqueezeNet arhitekture prikazan je na slici 4.



Slika 4. SqueezeNet arhitektura sa 2 detekcione glave.

Počinje jednostavnim konvolucionim slojem nakon kog sledi 8 okidačkih modula, kod kojih se broj filtara postepeno povećava od početka ka kraju mreže. Sažimanje maksimumom realizovano je nakon početnog konvolucionog sloja, fire4 i fire8 modula.

Na završni konvolucionni sloj modela povezan je novi sloj proširen faktorom 2, koji se ujedno spaja i sa prethodnom mapom obeležja istih dimenzija. Na ovaj način formirane su dve tzv. detekcione glave, od kojih je jedna dvostruko manja i bolje detektuje sitnije objekte u slici. Broj i dimenzije detekcionih glava u korelaciji su sa raznolikošću i dimenzijama objekata koji se detektuju.

3.1. Generisanje okvira i priprema baze podataka

Dimenzije apriornih okvira određuju se algoritmom grupisanja k -najbližih suseda nad okvirima objekata iz skupa za obuku [5]. Analiziraju se svi okviri trening skupa, a kao rezultat generiše se k odabranih okvira koji najbolje odgovaraju njihovim dimenzijama.

Kako je cilj odrediti dimenzije početnih okvira koji vode do dobrog rezultata preklapanja, za računanje udaljenosti koristi se mera koja ne zavisi od veličine okvira (5).

$$d(okvir, centroid) = 1 - IOU(okvir, centroid) \quad (5)$$

Model detektora implementiran je korišćenjem programske platforme MATLAB. Ova platforma nudi različite modele konvolucionih neuronskih mreža sa već podešenim vrednostima težinskih faktora, naučenih tokom obuke nad Imagenet bazom podataka. Unapred naučene težine pomažu procesu obuke sopstvene mreže tako što ubrzavaju konvergenciju, kao i učenje. Nakon podele uzoraka na skupove, implementirana je funkcija za proveru validnosti uzoraka. Uzorci sa neodgovarajućim formatom slike, neočekivanim vrednostima koordinata okvira ili pogrešnim formatom labela isključeni su iz razmatranja. Proširivanje raznolikosti test uzoraka (engl. *data augmentation*) doprinosi povećanju tačnosti mreže. Implementirane su varijacije boja, horizontalne rotacije i skaliranje za 10%.

4. REZULTATI

CrowdAI baza sadrži 9.423 slike. Skup podataka podeljen je u odnosu 70:30 na trening i test skup, respektivno, te je u obuci učestvovalo 6.657 slika, dok je za procenu tačnosti modela korišćeno 2.766 uzoraka. Kako uzorci test skupa nisu učestvovali u obuci, nije postojala bojazan da su parametri mreže prilagođeni uzorcima koji se korišćeni za evaluaciju performansi.

Originalni uzorci test skupa su slike veoma visoke rezolucije (1920x1200x3). Početni pokušaji obuke konvolucionih mreža, koje prihvataju ulazne slike datih dimenzija, bili su neuspešni. Arhitektura, odabrana teorijskom analizom, pokazala se kao neadekvatna. Tokom procesa obuke, već u prvoj epohi, uočeno je da greška mreže ima nagli porast, čime je dalji trening mreže postao beskoristan. Uz pretpostavku da je arhitektura mreže previše jednostavna, pokušalo se sa povećanjem broja konvolucionih slojeva i povećanjem broja filtara u okviru njih. Međutim, pretpostavka da jednostavnost mreže dovodi do loših rezultata nije ni potvrđena, niti opovrgnuta. Naime, memorijski i procesorski kapaciteti računara korišćenog za istraživanje nisu bili dovoljni da se obuka završi. Stoga, ulazni frejmovi visoke rezolucije su bili komprimovani u format (227x227x3).

4.1 Evaluacione mere

Glavne evaluacione mere kod detekcije objekata su preciznost i osetljivost. Preciznost modela govori koliki je udeo tačno klasifikovanih objekata u skupu klasifikovanih, dok osetljivost govori koliki je udeo tačno klasifikovanih uzoraka u skupu svih pozitivnih uzoraka. Mera koja povezuje preciznost i osetljivost naziva se prosečna preciznost (engl. *average precision*).

Prosečna preciznost predstavlja srednju vrednost preciznosti dobijenih za N jednako razmaknutih pragova za pouzdanost. U izrazu (6), AP predstavlja prosečnu pouzdanost, N je broj pragova, a P obeležava preciznost.

$$AP = \frac{1}{N} \sum_{k=0}^{N-1} P \left[r = \frac{k}{N-1} \right] \quad (6)$$

Mera koja opisuje kvalitet nekog detektora je srednja prosečna preciznost mAP (engl. *mean average precision*). Srednja prosečna preciznost predstavlja srednju vrednost prosečnih preciznosti za sve klase koje model detektuje nad celokupnim test skupom (7).

$$mAP = \frac{1}{Q} \sum_{i=0}^{Q-1} AP_i \quad (7)$$

4.2 Poređenje rezultata mreža sa različitim brojem apriori okvira

U ovom eksperimentu korišćena je opisana arhitektura SqueezeNet konvolucione mreže sa dve detekcione glave. Mreža se sastoji od 13 konvolucionih slojeva, ne računajući aktivacione mape u detekcionim glavama. Unutar svih slojeva mreže korišćene su tri različite aktivacione funkcije: ReLU, leaky ReLU i hiperbolički tangens. U sloju sažimanja korišćena je metoda sažimanja maksimalnom vrednošću sa korakom 2.

Rezultati modela postignuti tokom 30 epoha za slučaj 11 i 5 apriornih okvira predstavljeni su u Tabeli 1 preko vrednosti prosečnih preciznosti po klasama i srednje prosečne preciznosti modela.

Tabela 1. Rezultati detektora sa različitim brojem apriori okvira.

Broj okvira	AP			mAP
	Automobil	Kamion	Pešak	
11	84,31%	57,35%	88,27%	76,64%
5	80,50%	41,09%	89,58%	70,39%

Naredna zamisao u pogledu poboljšanja tačnosti detektora je promena korišćenih aktivacionih funkcija. Ovde su pored ReLU aktivacione funkcije, performanse modela sa 11 apriori okvira evaluirane za hiperbolički tangens (HT) i leaky ReLU aktivacione funkcije.

Tabela 2. Rezultati detektora sa različitim aktivacionim funkcijama.

	AP			mAP
	Automobil	Kamion	Pešak	
ReLU	84,31%	57,35%	88,27%	76,64%
Leaky ReLU	85,07%	61,04%	95,99%	80,70%
HT	79,46%	19,21%	93,11%	63,93%

5. ZAKLJUČAK

Detektor objekata se u slučaju sa 11 apriori okvira, prema srednjoj prosečnoj preciznosti, pokazao kao uspešniji u odnosu na model sa 5 apriori okvira za oko 6%. Iz tog razloga je dalje istraživanje nastavljeno sa prvo pomenutim modelom. Leaky ReLU aktivaciona funkcija najbolje se pokazala u slučaju sve tri klase. Pokazuje dobre performanse čak i u slučaju klasa sa manjim brojem predstavnika u skupu podataka (Kamion, Pešak). Neki od pravaca daljeg istraživanja su testiranje modela na većim skupovima za obuku i sa kompleksnijim arhitekturama mreže.

6. LITERATURA

- [1] L. Hunjung, „Awesome vehicle datasets“ (CrowdAI baza podataka), <https://github.com/hunjung-lim/awesome-vehicle-datasets>
- [2] J. Brownlee, „A Gentle Introduction to Object Recognition With Deep Learning“, <https://machinelearningmastery.com/object-recognition-with-deep-learning>. (poslednji pristup u martu 2021. godine)
- [3] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, “Gradient-based learning applied to document recognition”, *Proceedings of the IEEE*, 86(11), pp. 2278–2324, 1998.
- [4] A. Zhang, Z.C. Lipton, M. Li, A.J. Sola, „Dive into Deep Learning“, <https://d2l.ai/> (poslednji pristup u martu 2021. godine)
- [5] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [6] F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, K. Keutzer, “SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and <0.5MB Model Size”, arXiv Prepr. arXiv1602.07360, 2016.
- [7] K. He, J. Sun, “Convolutional neural networks at constrained time cost“, *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Kratka biografija:



Sofija Pantović rođena je u Vrbasu 1995. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Automatika i upravljanje sistemima odbranila je 2021.god.

kontakt: sofija.pantovic33@gmail.com

U realizaciji Zbornika radova Fakulteta tehničkih nauka u toku 2020. godine učestvovali su sledeći recenzenti:

Aco Antić	Đorđe Lađinović	Milan Mirković	Slobodan Krnjetin
Aleksandar Erdeljan	Đorđe Obradović	Milan Rapajić	Slobodan Morača
Aleksandar Kovačević	Đorđe Vukelić	Milan Segedinac	Sonja Ristić
Aleksandar	Đula Fabian	Milan Simeunović	Srđan Kolaković
Kupusinac	Đura Oros	Milan Trifković	Srđan Popov
Aleksandar Ristić	Đurđica Stojanović	Milan Trivunić	Srđan Vukmirović
Bato Kamberović	Filip Kulić	Milan Vidaković	Staniša Dautović
Biljana Njegovan	Goran Sladić	Milena Krklješ	Stevan Gostojić
Bogdan Kuzmanović	Goran Švenda	Milica Kostreš	Stevan Milisavljević
Bojan Batinić	Gordana	Milica Miličić	Stevan Stankovski
Bojan Lalić	Milosavljević	Mijodrag Milošević	Strahil Gušavac
Bojan Tepavčević	Gordana Ostojić	Milovan Lazarević	Svetlana Bačkalić
Bojana Beronja	Igor Budak	Miodrag Hadžistević	Svetlana Nikoličić
Branislav Atlagić	Igor Dejanović	Miodrag Zuković	Tanja Kočetov
Branislav Nerandžić	Igor Karlović	Mirjana Damnjanović	Tatjana Lončar -
Branka Nakomčić	Igor Peško	Mirjana Malešev	Turukalo
Branko Milosavljević	Ivan Beker	Miroslava Radeka	Uroš Nedeljković
Branko Škorić	Igor Maraš	Mirko Borisov	Valentina Basarić
Damir Đaković	Ivan Mezei	Miro Govedarica	Velimir Čongradec
Danijela Ćirić	Ivan Todorović	Miroslav Hajduković	Veran Vasić
Danijela Gračanin	Ivana Katić	Miroslav Kljajić	Veselin Perović
Danijela Lalić	Ivana Kovačić	Miroslav Popović	Višnja Žugić
Darko Čapko	Ivana Maraš	Miroslav Zarić	Vladimir Katić
Darko Marčetić	Ivana Miškeljin	Mitar Jocanović	Vladimir Mučenski
Darko Reba	Jasmina Dražić	Mitar Đogo	Vladimir Strezoski
Dejan Ecet	Jelena Atanacković	Mladen Kovačević	Vlado Delić
Dejan Jerkan	Jeličić	Mladen Tomić	Vlastimir Radonjanin
Dejan Ubavin	Jelena Borocki	Mladen Radišić	Vojin Ilić
Dejana Nedučin	Jelena Demko Rihter	Nebojša Brkljač	Vuk Bogdanović
Dragan Ivanović	Jelena Radonić	Neda Milić Keresteš	Zdravko Tešić
Dragan Jovanović	Jelena Slivka	Nemanja	Zoran Anišić
Dragan Ivetić	Jelena Spajić	Stanisavljević	Zoran Brujić
Dragan Jovanović	Jovan Petrović	Nemanja Sremčev	Zoran Čepić
Dragan Kukolj	Jovanka Pantović	Nikola Đurić	Zoran Jeličić
Dragan Mrkšić	Laslo Nađ	Nikola Jorgovanović	Zoran Mitrović
Dragan Pejić	Lazar Kovačević	Nikola Radaković	Zoran Papić
Dragan Šešlija	Leposava Grubić	Ninoslav Zuber	Željen Trpovski
Dragana Bajić	Nešić	Ognjen Lužanin	Željko Jakšić
Dragana	Livija Cvetičanin	Pavel Kovač	
Konstantinović	Ljiljana Vukajlov	Peđa Atanasković	
Dragana Šarac	Ljiljana Cvetković	Petar Malešev	
Dragana Štrbac	Ljubica Duđak	Platon Sovilj	
Dragoljub Šević	Maja Turk Sekulić	Predrag Šidžanin	
Dubravka Bojanić	Marinko Maslarić	Radivoje Dinulović	
Dušan Dobromirov	Marko Marković	Radomir Kojić	
Dušan Gvozdenac	Marko Todorov	Radovan Štulić	
Dušan Kovačević	Marko Vekić	Relja Strezoski	
Dušan Uzelac	Maša Bukurov	Slavica Mitrović	
Duško Bekut	Matija Stipić	Slavko Đurić	
Đorđe Ćosić	Milan Čeliković	Slobodan Dudić	

