

WEB APLIKACIJA ZA PODRŠKU RADA OFF-GRID SOLARNIH ELEKTRANA I BATERIJSKIH SKLADIŠTA**WEB APPLICATION FOR SUPPORTING OFF-GRID SOLAR POWER PLANTS AND BATTERY STORAGE**

Lana Slović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu prikazan je sistem za upravljanje off-grid solarnim elektranama i baterijskim skladištima. Sistem omogućava odabir lokacija za postavljanje solarnih panela i baterija, izračunavanje količine električne energije koju proizvode solarni paneli, prikaz dijagrama potrošnje, proizvodnje energije i nivoa napunjenosti baterija, kao i predikciju vremenskih uslova pomoću metoda mašinskog učenja, na osnovu kojih se procenjuje buduća proizvodnja energije u panelima. Cilj ovog rada je doprinos razvoju rešenja za upravljanje obnovljivim izvorima energije, koja omogućavaju efikasnije korišćenje resursa, smanjenje zavisnosti od fosilnih goriva i povećanje energetske održivosti.

Ključne reči: *Obnovljivi izvori energije, solarni paneli i baterijska skladišta, veb, mašinsko učenje, predikcije*

Abstract – This paper presents a system for managing off-grid solar power plants and battery storage facilities. The system enables the selection of locations for installing solar panels and batteries, calculation of the amount of electricity generated by the solar panels, display of consumption diagrams, energy production, and battery charge levels, as well as weather forecasting using machine learning methods to estimate future energy production by the panels. The aim of this paper is to contribute to the development of solutions for managing renewable energy sources, enabling more efficient resource utilization, reducing dependence on fossil fuels, and increasing energy sustainability.

Keywords: *Renewable energy sources, solar panels and battery storage, web, machine learning, predictions*

1. UVOD

Sve veća potražnja za održivim izvorima energije u poslednjim decenijama podstakla je razvoj tehnologija obnovljivih izvora, poput solarnih elektrana. U Srbiji je uočljiv napredak u ovoj oblasti, što potvrđuje rast broja domaćinstava koja koriste solarne panele i doprinose energetske stabilnosti i nezavisnosti kroz primenu obnovljivih izvora energije.

Ovaj rad istražuje razvoj web aplikacije za upravljanje off-grid solarnim elektranama i baterijskim skladištima.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Bošković, docent

Cilj aplikacije je omogućiti korisnicima da u realnom vremenu prate proizvodnju, potrošnju i skladištenje energije, uz analizu i vizualizaciju podataka. Aplikacija koristi savremene tehnologije poput mašinskog učenja, pružajući alate za predikciju i optimizaciju sistema. Krajnji cilj je razvoj softverskog rešenja koje doprinosi efikasnijem korišćenju resursa i širenju primene obnovljivih izvora energije.

2. OSNOVNI POJMOVI**2.1. Obnovljivi izvori energije i solarna energija**

U 21. veku, sve veća svest o ograničenjima i ekološkim posledicama korišćenja fosilnih goriva dovela je do razvoja obnovljivih izvora energije (OIE) poput solarnih, vetroelektrana, hidroelektrana i geotermalnih izvora, koji obećavaju čistu energiju i energetske nezavisnost. Ipak, OIE imaju ograničenja, poput niske gustine energetskih tokova (solarno zračenje u proseku 150–250 W/m², a vetar i voda oko 500 W/m²) i njihove promenljivosti, što zahteva značajne troškove za opremu za prikupljanje i skladištenje energije. Solarni paneli zauzimaju vodeću poziciju među OIE, omogućavajući efikasnu konverziju solarne energije u električnu energiju, toplotu i hlađenje. Iako njihova instalacija ima visoke troškove, tehnologije poput fotonaponskih (PV) pretvarača sa efikasnošću od 15–17% nalaze sve širu primenu. PV sistemi u srednjim geografskim širinama proizvode 900–1500 kWh/kWp godišnje, odnosno 120–200 kWh/m² godišnje, čineći ih ključnim za prelazak na održiviju energetske budućnost [1].

2.2. Prikupljanje i obrada podataka

Kako bi se dobili verodostojni rezultati, neophodno je prikupiti precizne podatke o vremenskim uslovima na osnovu kojih će se računati proizvodnja energije i vršiti predikcije. Podaci koji se prikupljaju uključuju trenutnu temperaturu, procenat oblačnosti, vreme izlaska i zalaska Sunca, a prikupljaju se periodično, na svakih sat vremena, za željenu lokaciju na osnovu njenih koordinata. Pored prikupljenih podataka, potrebni su i određeni konstantni parametri koji igraju ključnu ulogu u izračunavanju proizvodnje električne energije.

S obzirom na to da se radi o solarnim panelima, tokom noći (između momenta zalaska i momenta izlaska Sunca) količina proizvedene električne energije biće jednaka nuli, dok će se u suprotnom koristiti formule (1), (2) i (3) za izračunavanje trenutne količine električne energije proizvedene u solarnim panelima.

$$T_{celija} = T_{spoljna} + k \cdot (1 - C) \quad (1)$$

$$f(T_{celija}) = 1 - \beta \cdot |T_{celija} - T_{referentna}| \quad (2)$$

$$E_{trenutno} = N_{panela} \cdot P_{instalirano} \cdot \eta \cdot (1 - C) \cdot f(T_{celija}) \cdot t \quad (3)$$

Parametri:

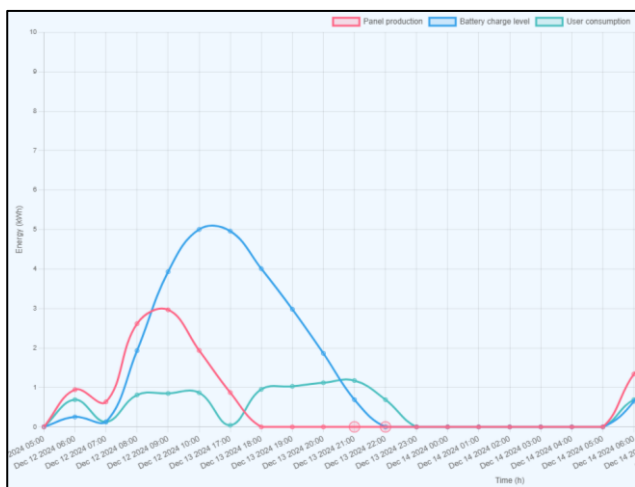
- k – koeficijent uticaja oblačnosti na temperaturu solarnih ćelija, upotrebljena vrednost je 5°C
- β – odražava uticaj temperature na efikasnost, upotrebljena vrednost je $0,005^{\circ}\text{C}^{-1}$
- $T_{referentna}$ – temperatura pri kojoj se efikasnost solarnih panela ne menja, upotrebljena vrednost je 25°C
- η – efikasnost solarnih panela, upotrebljena vrednost je 80%

Uslovi zajedničkog funkcionisanja solarnih panela i baterijskih skladišta u odnosu na potrošnju su sledeći:

- Ukoliko solarni paneli proizvedu više energije nego što se potroši, tada se viškom energije puni baterijsko skladište, ukoliko ima kapaciteta.
- Ukoliko solarni paneli proizvedu manje energije nego što se troši, tada se neophodna energija crpi iz baterijskih skladišta, ukoliko skladišta poseduju tu količinu energije.
- Ukoliko je baterijsko skladište prazno, a proizvodnja energije je manja od predviđene potrošnje, tada je potrošnja energije jednaka proizvodnji energije.
- Ukoliko je baterijsko skladište puno, a proizvodnja energije je veća od predviđene potrošnje, tada višak proizvedene energije curi.

Uz navedene formule i uslove izračunavanja, dobija se prikaz proizvodnje električne energije, nivoa napunjenosti baterije i potrošnje električne energije za jedan sistem.

Vrednosti izračunate na osnovu navedenih formula i uslova se čuvaju u bazi podataka. Ponašanje sistema se najbolje može ispratiti formiranjem dijagrama na kom se prikazuju sledeće tri vrednosti: trenutna potrošnja energije, trenutna proizvodnja energije i nivo napunjenosti baterije u određenom trenutku. Primer takvog dijagrama je prikazan na slici 1.



Slika 1. Dijagram proizvodnje, potrošnje energije i nivo napunjenosti baterije

2.3. Mašinsko učenje i predikcije

Mašinsko učenje se kontinuirano razvija u svetu informacionih tehnologija i dobija na značaju u različitim poslovnim sektorima. Popularno je među svim tehnologijama. To je oblast proučavanja koja omogućava računarima da automatski uče i poboljšavaju svoje performanse na osnovu iskustva. Definiše se kao tehnologija koja se koristi za obučavanje mašina da izvršavaju različite radnje, kao što su predviđanja, preporuke, procene itd., na osnovu istorijskih podataka ili prethodnog iskustva. Mašinsko učenje se stoga fokusira na jačanje kompjuterskih programa uz pomoć prikupljanja podataka iz različitih opažanja. Omogućava računarima da se ponašaju kao ljudska bića, obučavajući ih uz pomoć prethodnog iskustva i prediktivnih podataka [2].

Random Forest je jedan od najpoželjnijih algoritama mašinskog učenja. Slično KNN-u (K Nearest Neighbour) i stablu odlučivanja, on takođe omogućava rešavanje klasifikacionih kao i regresionih problema, ali se preferira kada postoji potreba za rešavanjem složenijih problema i poboljšanjem performansi modela. Algoritam random forest se bazira na konceptu ansambla učenja, što je proces kombinovanja više klasifikatora.

Klasifikator random forest se sastoji od kombinacije više stabala odlučivanja. Ova kombinacija uzima u obzir prosečnu predikciju svih stabala i poboljšava tačnost modela. Veći broj stabala u šumi dovodi do veće tačnosti i sprečava problem preterivanja. Dodatno, zahteva manje vremena za obuku u poređenju sa drugim algoritmima [2].

3. ARHITEKTURA SISTEMA

Višeslojna arhitektura ove aplikacije ima jasno razdvojene slojeve za frontend, backend i bazu podataka. Za dodavanje podataka o geografskim lokacijama i vremenskim svojstvima, aplikacija komunicira sa postojećim serverima kao što su OpenWeatherMap i GoogleMapsAPI.

3.1. Backend sloj

Servisni deo aplikacije koristi *Node.js*, *Express.js* za definisanje ruta i *Python*. Sadrži *mongoose* modele. Sastoji se od mikroservisa koji imaju komunikaciju sa svojim bazama podataka i međusobno komuniciraju preko *REST API*-ja. Takođe, servis prikuplja podatke sa *OpenWeatherMap*.

Mikroservisna arhitektura:

- **UserAuthService:** Za prijavu i registraciju korisnika.
- **UserService:** Upravljanje operacijama nad korisnicima (blokiranje, pretraga).
- **PanelsService:** Obrada operacija nad panelima (dodavanje, brisanje, proizvodnja energije).
- **BatteriesService:** Upravljanje nad baterijama (dodavanje, brisanje, ažuriranje).
- **ConstantParametersService:** Dodavanje i upravljanje konstantnim parametrima za proizvodnju energije.
- **EnergyConsumptionService:** Upravljanje podacima o potrošnji energije.

- **HistoryDataService:** Čuvanje istorijskih podataka o proizvodnji i vremenu.
- **ForecastPredictionService:** Python mikroservis. Predikcija proizvodnje energije koristeći algoritam mašinskog učenja RandomForest.

3.2. Frontend sloj

Klijentski deo aplikacije razvijen je u *React.js* i sadrži komponente i servise koji komuniciraju sa mikroslužbama na serverskoj strani. Koristi *React-Leaflet* biblioteku za izbor lokacije na mapi i poziva *OpenWeatherMap* za prikaz trenutnih vremenskih parametara.

3.3. Baza podataka

Za skladištenje podataka koristi se *MongoDB*. Svaki mikroservis ima svoju bazu podataka u kojoj se čuvaju informacije o korisnicima, panelima, baterijama, parametrima i istorijskim podacima o proizvodnji, potrošnji i nivou napunjenosti baterije.

S obzirom na to da mikroservisi međusobno komuniciraju preko *REST API*-ja, tim putem se propagiraju promene između njih kako bi se postigla konzistentnost sistema. Primer: *UserService* ima funkcionalnost za blokiranje/odblokiranje korisnika, gde menja vrednost polja u bazi koje određuje da li je korisnik blokiran ili ne. Isto to polje je relevantno za *UserAuthService*, jer se na osnovu njega dozvoljava korisniku prijava na sistem. Iz tog razloga, *UserService* nakon promene vrednosti navedenog polja, šalje *UsersAuthService* servisu informaciju o novoj vrednosti polja i *UsersAuthService* ažurira podatke u svojoj *MongoDB* bazi.

4. PREDIKCIJE VREMENSKIH USLOVA I IZRAČUNAVANJE BUDUĆE KOLIČINE PROIZVODNJE ELEKTRIČNE ENERGIJE NA OSNOVU NJIH

Mikroservis za predikciju vremenskih uslova predstavlja posebnu Python aplikaciju, koja komunicira sa drugim servisima koristeći Flask API. Koristi napredne tehnike mašinskog učenja, i putem Random Forest algoritma vrši obradu istorijskih podataka o temperaturi i oblačnosti. Model koristi podatke o trenutnim i prethodnim temperaturama, kao i o oblačnosti, da bi učio obrasce koji se javljaju tokom vremena. Da bi se modelu omogućilo da koristi istorijske informacije, podaci o temperaturi i oblačnosti se "laguju". Ovo znači da se dodaju prethodne vrednosti temperature i oblačnosti iz poslednjih 48 časova kao nove karakteristike u skupu podataka. Time se omogućava modelu da prepozna obrasce u podacima koji su se dogodili u prošlosti i da ih koristi za predikciju budućnosti. Na osnovu pripremljenih podataka, model se obučava korišćenjem algoritma mašinskog učenja, poput Random Forest regresora. Ovaj algoritam analizira obrasce u podacima i uči kako da predviđa buduće temperature na osnovu ulaznih karakteristika (lagovanih temperatura i oblačnosti). Obučavanje uključuje deljenje podataka na obučeni i test skup, pri čemu se model optimizuje da bi postigao što tačnije predikcije. Nakon obučavanja, model može da predviđa temperature za narednih 24 časa. Na kraju, rezultati se dobijaju kao niz predviđenih temperatura i oblačnosti za naredni period.

5. IMPLEMENTACIJA PREDIKCIJA VREMENSKIH SERIJA

Implementacija predikcija budućih vremenskih serija obuhvata tri funkcije: za pripremu podataka, treniranje modela i predikciju budućih podataka. Biblioteke neophodne za implementaciju su *numpy* i *scikit-learn*, koje su neizostavan deo rada sa podacima i modelima mašinskog učenja. Podaci se učitavaju iz CSV fajla koji sadrži istorijske informacije, a zatim se pripremaju u funkciji `prepare_data` prikazane u listingu 1.

```
def prepare_data(data, target_column):
    #Dodavanje lagovanih vrednosti za temperaturu i oblačnost
    for i in range(1, 49):
        data[f'temperature_lag_{i}']=data['currentOutsideTemperature'].shift(i)
        data[f'cloudiness_lag_{i}']=data['currentCloudinessPercent'].shift(i)

    data = data.dropna()
    X = data[[f'temperature_lag_{i}' for i in range(1,49)] + [f'cloudiness_lag_{i}' for i in range(1, 49)]]
    y = data[target_column]

    return X, y
```

Listing 1. Funkcija za pripremu podataka za predikcije

Nakon pripreme podataka sledi treniranje modela a zatim i predikcija budućih vrednosti na osnovu primrepljenih podataka u funkciji `prepare_data`. Funkcije `train_model` i `predict_future_data` su prikazane u listingu 2.

```
def train_model(X, y):
    X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.1,random_state=42)

    model=RandomForestRegressor(random_state=42)
    model.fit(X_train, y_train)
    try:
        check_is_fitted(model)
        print("Model je uspešno treniran.")
    except:
        print("Model nije treniran.")

    return model, X_test, y_test

def predict_future_data(model, data):

    last_24_data=data.iloc[1][[f'temperature_lag_{i}' for i in range(1, 49)] + [f'cloudiness_lag_{i}' for i in range(1, 49)]] .values.reshape(1, -1)

    future_data = []

    for _ in range(48):
        predicted_data=model.predict(last_24_data)
        future_data.append(predicted_data[0])
```

```

last_24_data = np.roll(last_24_data, shift=-
1)
last_24_data[0, -1] = predicted_data[0]

return future_data

```

Listing 2. Funkcije za treniranje modela i predikcije budućih vrednosti

Model kreiran i treniran na ovaj način, tokom evaluacije je pokazao da su vrednosti Root of the Mean of the Square of Errors (RMSE) i Mean of Absolute value of Errors (MAE) manje od 1, što dokazuje tačnost modela. Primer RMSE i MAE za neki od lokacija za koje su se vršile prognoze:

- RMSE: 0.6755029072419337
- MAE: 0.5621307692307664

Nakon treniranja modela, evaluacije modela, i predikcija vremenskih uslova, sledi izračunavanje buduće proizvodnje električne energije za solarne panele na datoj lokaciji za koju se vršila predikcija. Izračunavanje se vrši na osnovu formula i parametara navedenih u drugom poglavlju, a prikaz rezultata predikcija se vrši putem dijagrama (buduće temperature, stepeni oblačnosti i količina proizvedene električne energije). Na slici 2 je prikazano poređenje vremenske prognoze preuzete sa *OpenWeatherMap API*-ja, izračunate proizvodnje energije na osnovu nje, vremenske prognoze putem mašinskog učenja i Random Forest algoritma, i njoj odgovarajuće izračunate proizvodnje energije.



Slika 2. Vremenska prognoza preuzeta sa *OpenWeatherMap API*-ja (gore) i preikcije uz mašinsko učenje (dole)

6. ZAKLJUČAK

Ovaj rad obrađuje razvoj sistema za upravljanje off-grid solarnim elektranama, kao i predikcije vremenskih uslova na osnovu kojih se dobija buduća količina električne energije koja će biti proizvedena. Uz pomoć prikupljanja podataka, korisničkog interfejsa i tehnologija mašinskog učenja omogućeno je praćenje proizvodnje i potrošnje električne energije, kao i nivoa napunjenosti baterije u sistemu solarnih panela i baterijskih skladišta, uz pružanje pomenutih predikcija. Samim tim, olakšano je praćenje sistema i dalje planiranje proizvodnje električne energije i korišćenja resursa u solarnim elektranama.

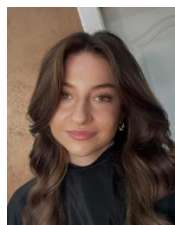
Rad doprinosi time što omogućava predviđanje budućih vremenskih uslova uz određenu tačnost predikcija, kao i buduće proizvodnje energije na osnovu tih uslova. Mikroservisna arhitektura sistema omogućava laku modifikaciju i nadogradnju sistema, kao i integraciju sa drugim sistemima, što ga čini fleksibilnim i prilagodljivim. Raznovrsnost tehnologija implementiranih u sistemu pokazala se kao efikasan način za postizanje preciznih rezultata. Rezultati dobijeni korišćenjem sistema ukazuju na visoku upotrebljivost algoritama mašinskog učenja u solarnim elektranama.

S obzirom na potencijal obnovljivih izvora energije, razvoj sistema za upravljanje ovakvim resursima i predikcije buduće proizvodnje električne energije doprinosi efikasnom korišćenju istih, smanjenju troškova i zavisnosti od neobnovljivih izvora energije. Osim što doprinosi energetske održivosti, ovakav sistem može poslužiti kao model za slične aplikacije u drugim sektorima baziranim na obnovljivim resursima.

4. LITERATURA

- [1] Rakhmatov A., Primov O., Mamadaliyev M., Tòrayev S., Xudoyazarov U., Ulugmurodov E., Xaydarov S., Razzoqov I., (2024), E3S Web of Conferences, Advancements in renewable energy sources (solar and geothermal): A brief review, 1-4, doi: 10.1051/e3sconf/202449701009
- [2] *Javatpoint*, (2024), pristupljeno u decembru 2024. godine sa <https://www.javatpoint.com/basic-concepts-in-machine-learning>

Kratka biografija:



Lana Slović rođena je u Prijepolju 2001. god. Po završetku Prijepoljske gimnazije, 2019. godine upisuje Fakultet tehničkih nauka u Novom Sadu, studijski program Primenjeno softversko inženjerstvo. Nakon završenih osnovnih akademskih studija, 2023. godine upisuje master akademske studije iz iste oblasti.

Kontakt: slovic17@gmail.com